Cluster analysis

Sneha Das

DTU

02582 Computational Data Analysis, 2025

Recap

Subspace models (PCA, Sparse PCA, PLS, CCA)

- What for?
- How?

Agenda

- What is cluster analysis
- Similarity and dissimilarity measures
- K-means clustering
- Hierarchical clustering
- Gaussian mixture
- Validation and model selection

Cluster Analysis

Unsupervised classification

- Separating or clustering observations
- Intuitive but vague definition

Given an underlying set of points, partition them into a collection of **clusters** so that points in the same cluster are close together, while points in different clusters are far apart.

Purpose

- Seeing structure in data
  - Gaining understanding
- Dimensionality reduction
- Outlier detection

Similarity and dissimilarity measure

In what sense are points close in one cluster and far from points in another cluster?

Similarity takes a large value when points are close.

Dissimilarity takes a large value when points are far apart. This reflects the **distance** between observations.

Any monotone-decreasing function can convert similarities to dissimilarities.

Both similarity and dissimilarity measures can be subjective. For example comparing the taste of three ice creams.

Clustering as an optimization problem

Defining the *variability* of points within a cluster:

$$Variability(c) = \sum_{i \in c} dist(mean(c), x_i)^2$$

Defining the dissimilarity of our clustering:

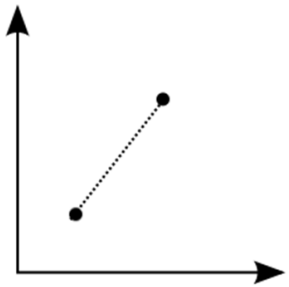
$$Dissimilarity(C) = \sum_{c \in C} variability(c)$$

Clustering as an optimization problem II

- Why variability and not variance?
- What happens if we minimize dissimilarity?

Euclidean distance

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{p} (x_{ik} - x_{jk})^2}$$

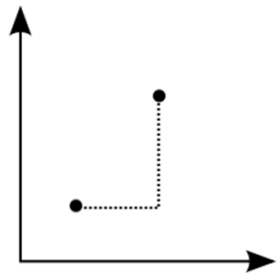


Useful for **quantitative** variables

Ordinal variables can be transformed to a quantitative scale.

Manhattan distance

$$d(x_i, x_j) = \sum_{k=1}^{p} \left| x_{ik} - x_{jk} \right|$$



Quantitative variables

Manhattan distance also called **city block distance** og **Hamming** distance.

Mahalanobis distance

$$d(x_i, x_j) = \sqrt{(x_i - x_j)^T \Sigma^{-1} (x_i - x_j)}$$

The distance is based on data itself and the two points are assumed to be of the same distribution with equal dispersion Σ.

Quantitative variables

Tanimoto distance (comparing sets, binary vectors)

$$d(x_i, x_j) = \frac{x_i^T x_j}{x_i^T x_i + x_j^T x_j - x_i^T x_j}$$

Let the sample x have $x_k = 1$ if it possesses the i^{th} attribute, and $x_k = 0$ otherwise.

The ratio of the number of shared attributes to the number possessed by x_i or x_j.

Often used in information retrieval and biological taxonomy - works well for **categorical** variables.

Weighted distances

$$d(x_i, x_j) = \sum_{k=1}^{p} w_k d_k(x_{ik}, x_{jk}),$$

$$\sum_{k=1}^{p} w_k = 1$$

Give different weight to the p attributes (variables).

w_k indicates that attribute k uses an individual dissimilarity and all p dissimilarities are then combined into one total dissimilarity (d) for all attributes.

Note that setting $w_k = 1/p$ does *not* necessarily give equal influence to the attributes.

We would have to normalize with the average distance for the k^{th} attribute.

Unsupervised clustering

We have data but no information about class belonging

- Group data in clusters
- Observations that are "near" each other should belong to the same class/cluster
- This can help us unveil an unknown structure in data
- It is like classification but without an answer, ie unsupervised

K-means clustering

- Super simple algorithm for clustering

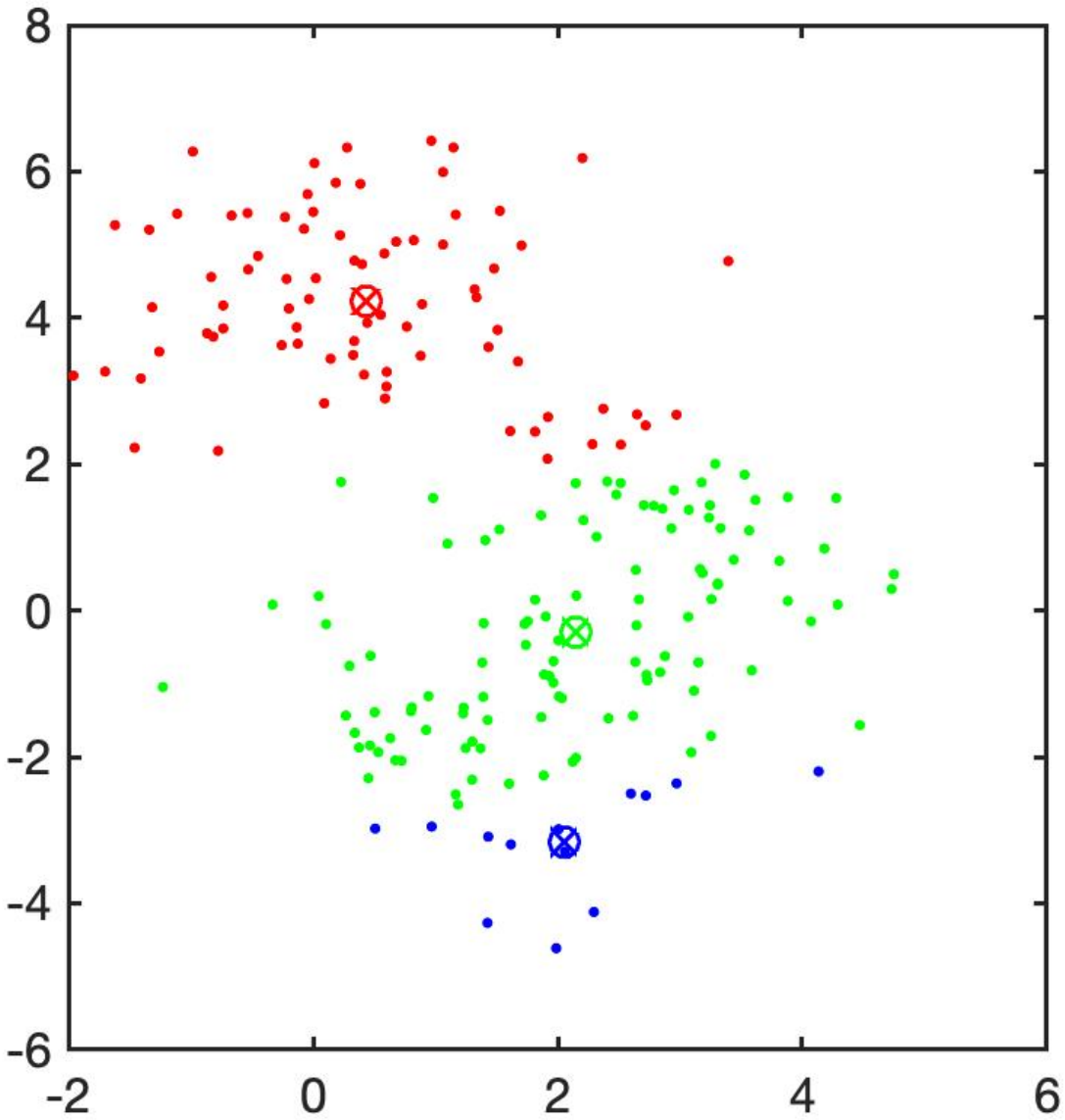
K-means clustering

Decide how many clusters, K, there should be, and randomly initialize K cluster centers (centroids).

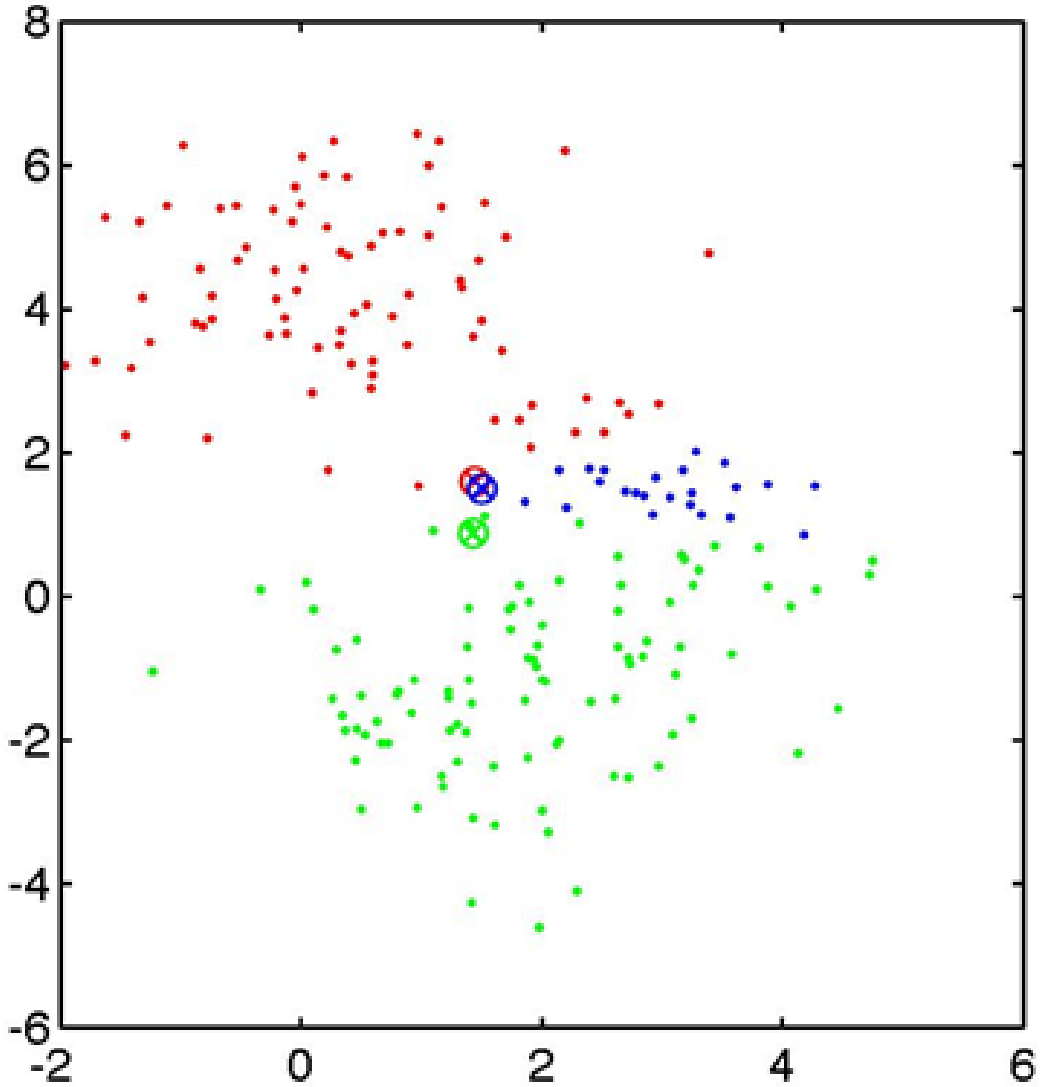
Alternating between two steps (until assignments do not change)

- Assign each point to the closest cluster center (centroid)
- Compute new cluster centers (centroids) according to the assignments

Example - initialize 3 centroids in dense areas



Example - initialize 3 centroids close to common center of gravity



K-medoids

- K-medoids use one of the observations as cluster center
  - Computationally much heavier than K-means
- Makes K-medoids more robust to outliers than K-means
- Also refered to as PAM, Partitioning Around Medoids

Hierarchical Clustering

- Generates a tree of observations (dendrogram)
- Each level of the tree reflects one number of clusters
- From all data in one cluster down to one observation in each cluster

Hierarchical clustering

- Do not require input of number of clusters
- Uses dissimilarity between clusters
- Two approaches
  - Bottom-up: Agglomerative (commonly used)
  - Top-down: Divisive
- $n - 1$ levels in the hierarchy
- At each level perform split or merge which gives largest between-group dissimilarity

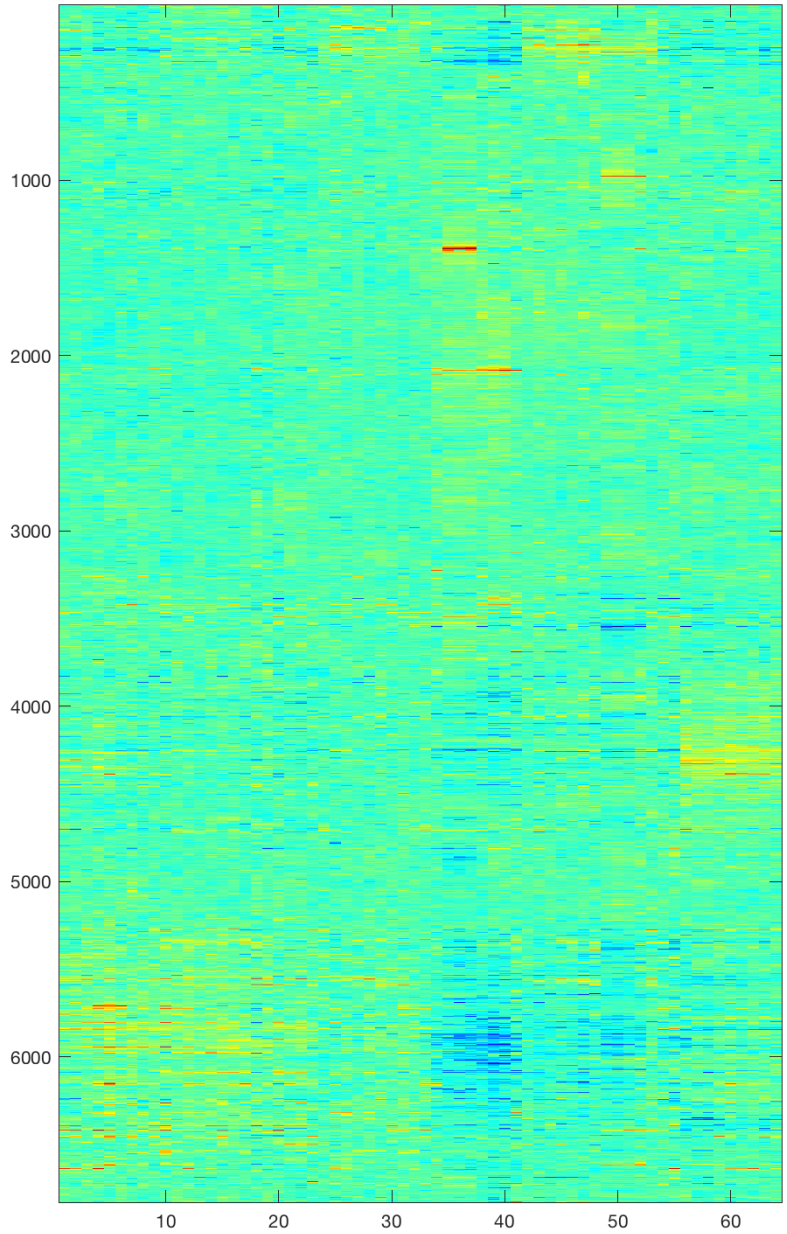
Microarray example

- The mRNA microarray data from ESL, example 4
- Samples from cancer tumors
- 6830 genes
- 64 samples

Unsupervised clustering:

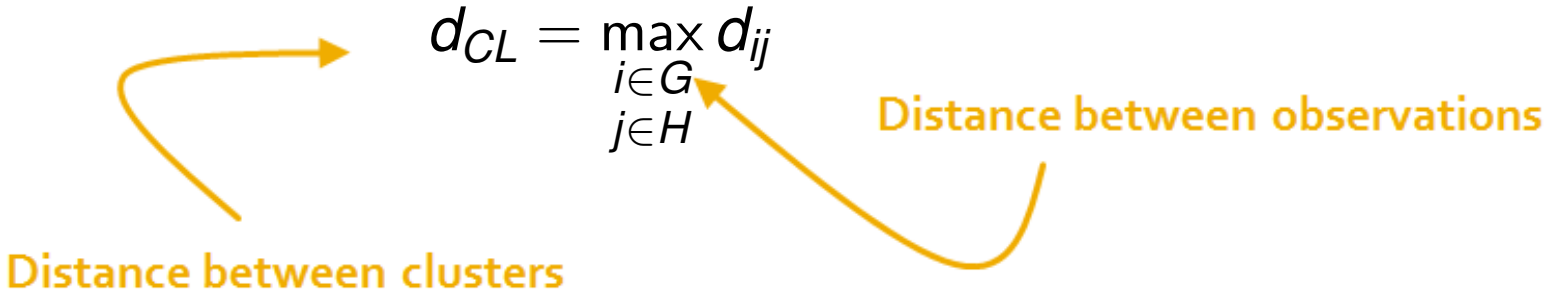
Which samples are most similar to each other, in terms of their expression profiles across genes.

Do similar samples share the same form of cancer?



Complete-linkage hierarchical clustering

Cluster-cluster distance measured as the distance from the furthest pair of points (i, j) from the clusters (G, H) respectively.



Also called the **furthest-neighbor** technique

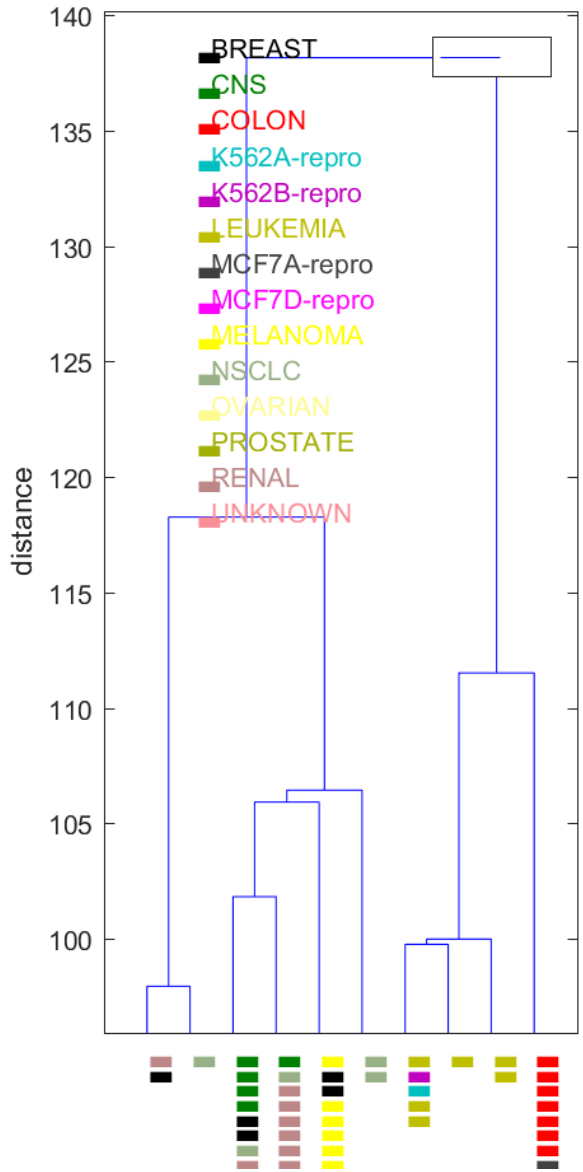
Microarray example - complete-linkage clustering

- Sample-sample: Euclidean
- Cluster-cluster: Complete
- Dendrogram cut at 10 nodes

Tends to give **balanced trees** independent of data

A **dendrogram** consists of many U-shaped lines connecting objects in a hierarchical tree.

The **height** of each U represents the distance between the two objects being connected.



Single-linkage hierarchical clustering

Cluster-cluster distance measured as the distance of the **closest** pair of points (i, j) from the clusters (G, H) respectively.

$$d_{SL} = \min_{\substack{i \in G \\ j \in H}} d_{ij}$$

Also called the **nearest-neighbor** technique

Chaining - a problem

Bottom-up example

	Copenhagen	Nuuq	Edinburgh	Reykavik
Copenhagen	0 km	3535 km	983 km	2107 km
Nuuq	3535 km	0 km	2765 km	1430 km
Edinburgh	983 km	2765 km	0 km	1374 km
Reykavik	2107 km	1403 km	1374 km	0 km

{Copenhagen} {Nuuq} {Edinburgh} {Reykavik}
{Copenhagen, Edinburgh} {Nuuq} {Reykavik}
{Copenhagen, Edinburg, **Reykavik** } {Nuuq} *single linkage*

or

{Copenhagen, Edinburg} {Nuuq, **Rekavik** } *complete linkage*

Ward-linkage hierarchical clustering

Cluster-cluster distance measured as the increment in within-cluster sum of squares, when merging clusters G and H

$$d_{Ward} = \sqrt{n_G n_H \frac{||\bar{x}_G - \bar{x}_H||_2^2}{(n_G + n_H)}}$$

Ward's distance measures how much the sum of squares will increase when we merge the two clusters,

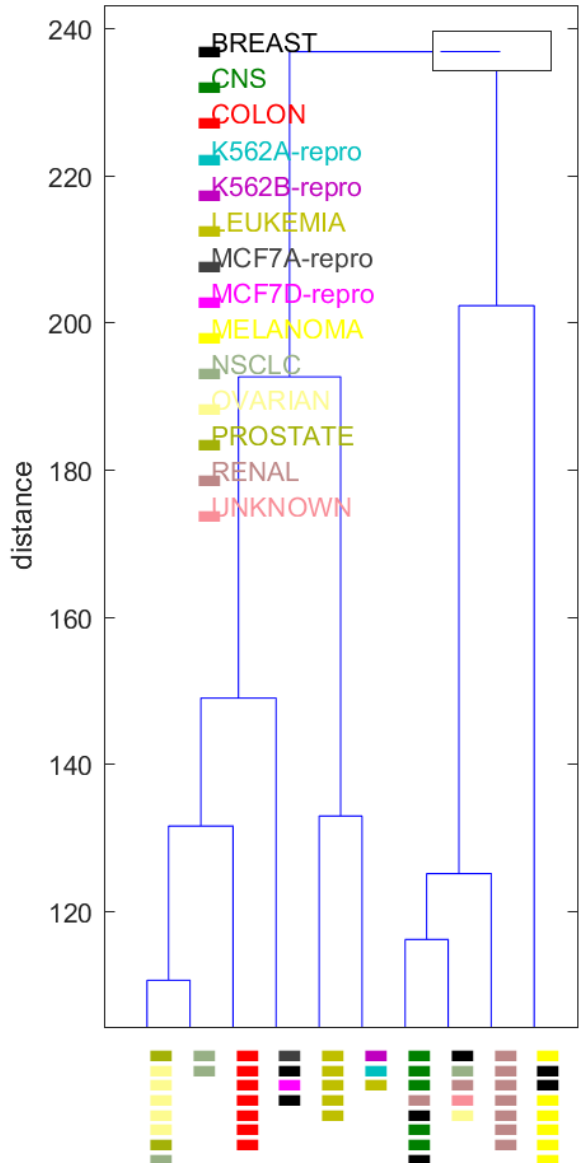
$$d_{Ward} = \sum_{i \in G \cup H} ||x_i - \bar{x}_{G \cup H}||_2^2 - \sum_{i \in G} ||x_i - \bar{x}_G||_2^2 - \sum_{i \in H} ||x_i - \bar{x}_H||_2^2$$

Microarray example - Ward-linkage clustering

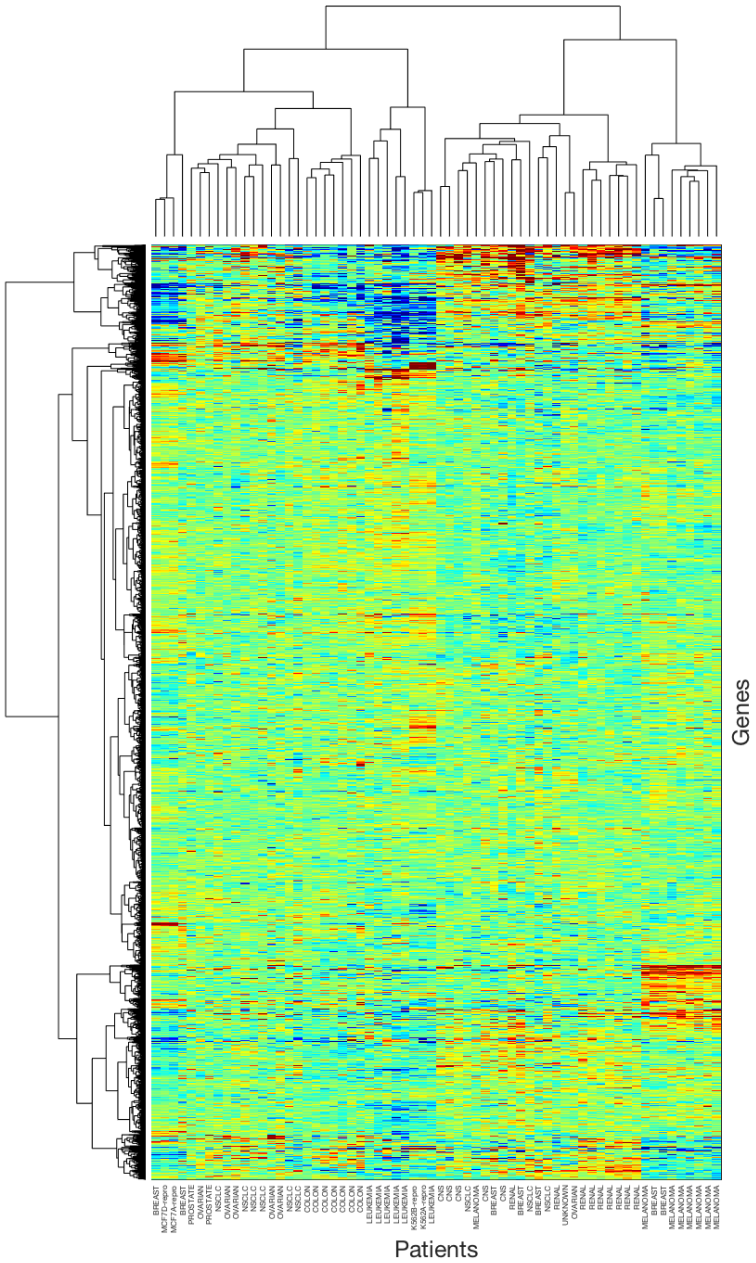
- Sample-sample: Euclidean
- Cluster-cluster: Ward

Tends to give a good **compromise** between balanced/unbalanced clusters

Can only be used with Euclidean distance



Two way clustering (biclustering)



- Cluster both genes and patients
- Presented as a heatmap

In Matlab: `clustergram(X)`
In R: `cim()` function from the `mixOmics` package
In Python: `SpectralBiclustering` function in `scikitlearn`

A warning...

Hierarchical clustering will **always** generate a dendrogram

- Even when data are completely random
- Be careful with the interpretation
- It is the application that tells if the structure is relevant (use your **domain knowledge**)

The problem of validation

- How to select the number of clusters?

Selecting the number of clusters

- Gap-statistics
- Silhouette method (heuristic)
- Goodness-of-fit measures (when a distribution is assumed)
  - Chi-squared statistics
  - Kolmogorov-Smirnov statistics
  - AIC and BIC
- Biological or physical interpretation

Why not cross validation?

Silhouette method - heuristic

Let $a(i)$ denote the average distance between observation i and all other observations assigned to the same cluster.

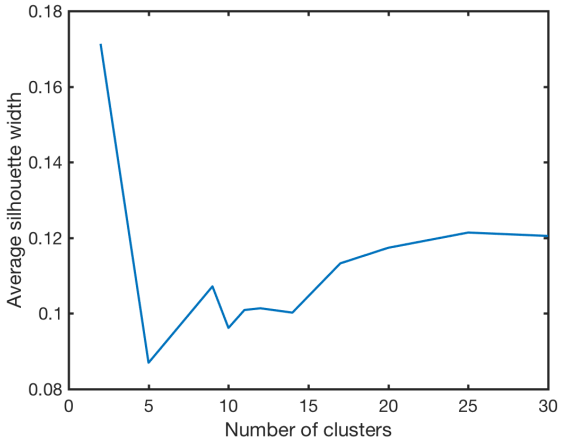
Let $b(i)$ denote the average distance between observation i and all observations assigned to the neighboring cluster (cluster where i is not a member and where average distance is largest).

Silhouette is defined as

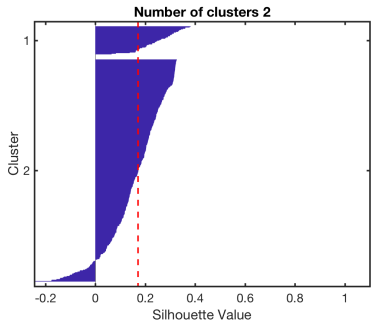
$$s(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}}$$

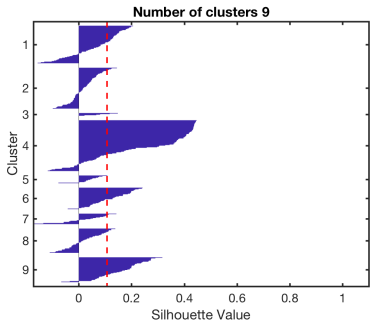
The average $s(i)$ over all observations is an estimate of how appropriately the points are clustered. Select number of clusters when all clusters have observations above average silhouette width, or choose K^* with the maximum average silhouette.

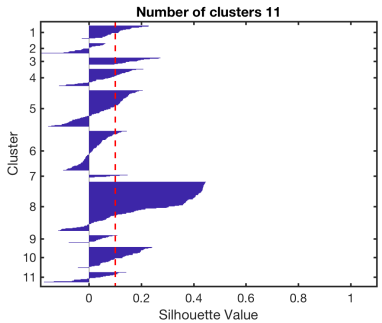
Example - Zip data

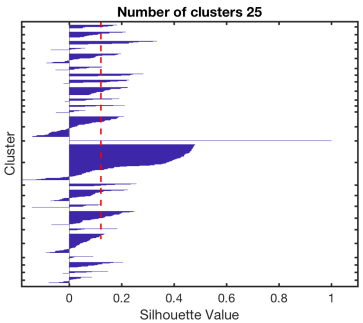


Using Average silhouette: Optimal $K^* = 2$ or 25?









Using silhouette plots: Optimal $K^* = 2$ or 9?

Within cluster dissimilarity

Distance between all points in one cluster (Euclidean distance)

$$D_\ell = \sum_{\substack{C_k(i)=\ell \\ C_k(j)=\ell}} ||x_i - x_j||^2$$

$$= \ldots$$

$$= N_\ell \sum_{C_k(i)=\ell} ||x_i - \bar{x}_\ell||^2$$

Within cluster dissimilarity

$$W_k = \sum_\ell \frac{1}{2N_\ell} D_\ell$$

($C_k(i) =$ cluster for obs. i when we have k clusters in the model.)

Gap-statistic

Compares the log criterion value with K clusters to the expected log criterion value for **uniformly distributed** data (20 simulations)

$$G(K) = log(U_k) - log(W_k)$$

- U_k within cluster dissimilarity, simulated data - mean over 20 samples
- W_k within cluster dissimilarity, actual data

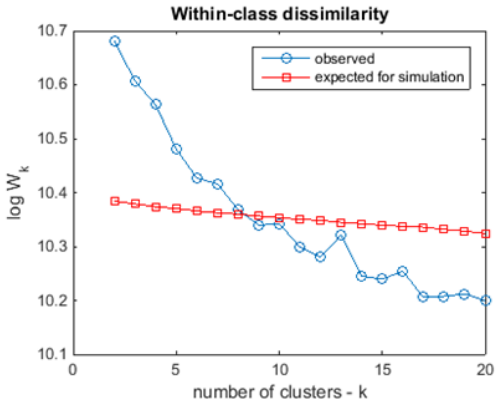
Choose

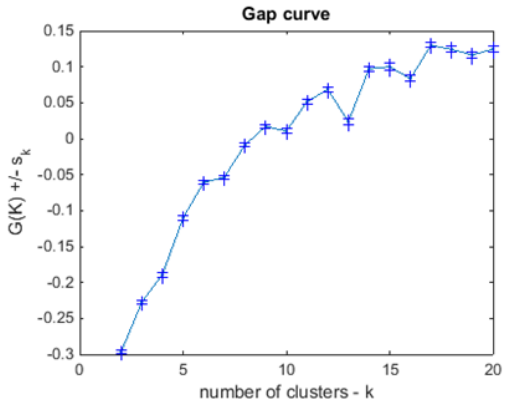
$$K^* = \arg\min_k \{K | G(K) \geq G(K+1) - s'_{K+1}\}$$

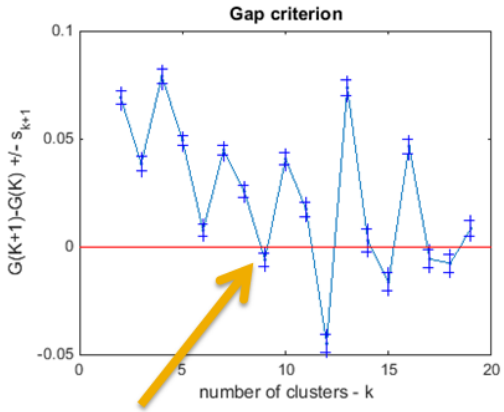
where

$$s'_{K+1} = \text{std}(\log(U_K))\sqrt{1 + 1/20}$$

Example - zip data







Optimal $K^* = 9$

Based on simulation - might differ from one simulation to another

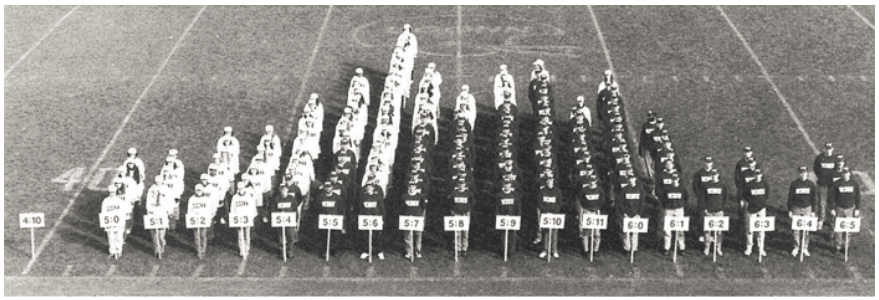
Gap-statistic

- Can be used both for K-means, K-medoids and hierarchical clustering
- Works with different measures of the within-cluster dissimilarity

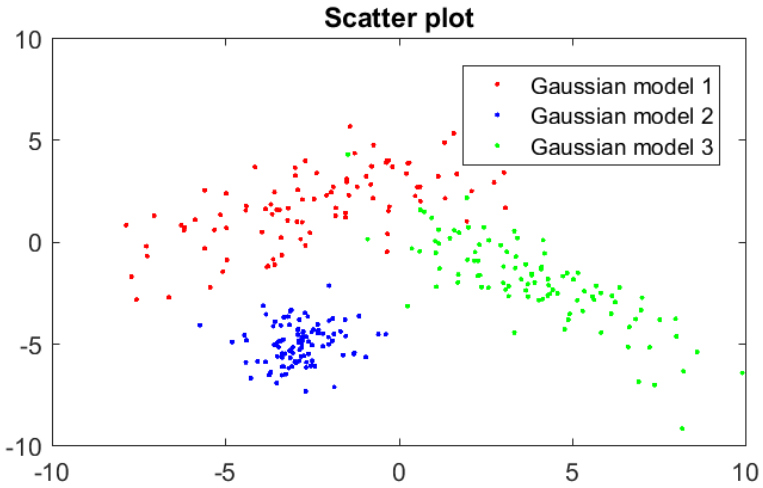
Gaussian Mixture Modeling and Expectation Maximization

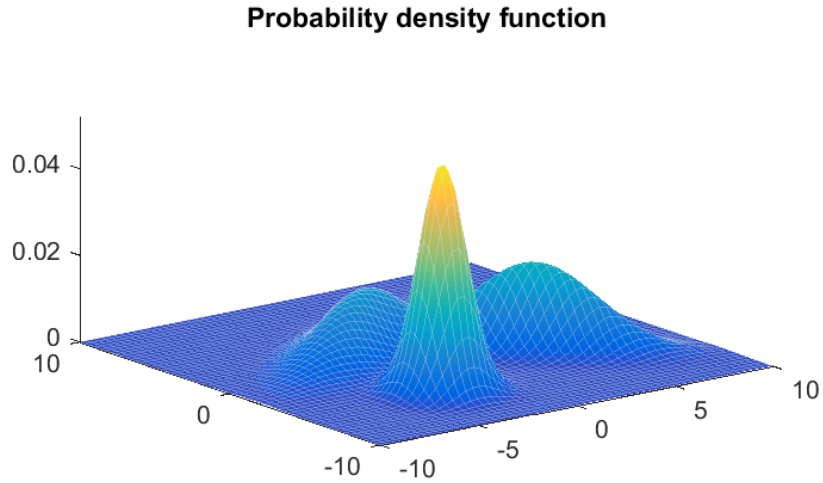
Gaussian Mixture Modeling:

- Data belong to one of several Gaussian distributions
- An un-observed (latent) random variable selects which distribution the observation comes from.
- This gives a (complicated) likelihood function
- Easily solved using the EM algorithm



Gaussian models





Observations from three Gaussian distributions

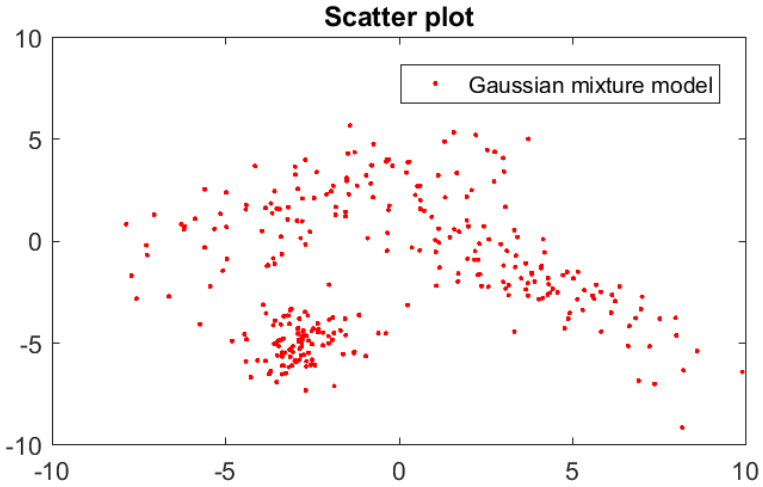
$$\begin{cases} X_i \in N(\mu_1, \Sigma_1) & \text{if } Z_i = 1 \\ X_i \in N(\mu_2, \Sigma_2) & \text{if } Z_i = 2 \\ X_i \in N(\mu_3, \Sigma_3) & \text{if } Z_i = 3 \end{cases}$$

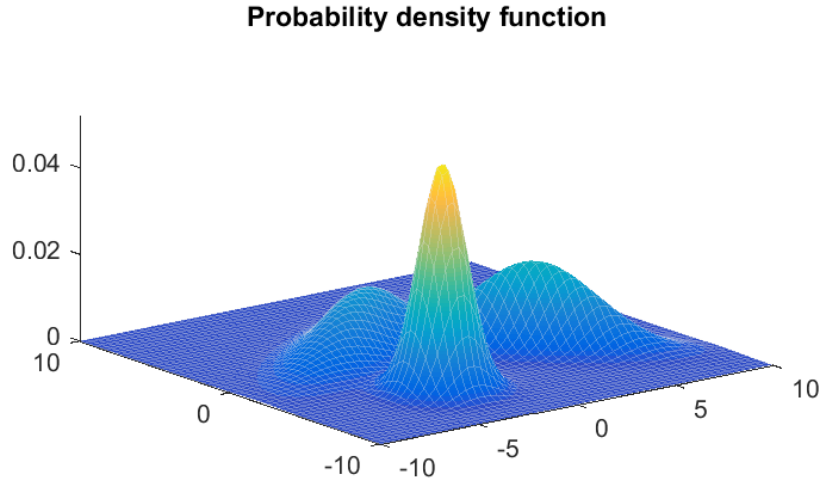
Known variables, Z_i, indicate cluster/distribution

$$\begin{cases} Z_i = 1 & i = 1, ..., 100 \\ Z_i = 2 & i = 101, ..., 200 \\ Z_i = 3 & i = 201, ..., 300 \end{cases}$$

What are the unknown model parameters?

Gaussian mixture models





Observations from three Gaussian distributions

$$\begin{cases} X_i \in N(\mu_1, \Sigma_1) & \text{if } Z_i = 1 \\ X_i \in N(\mu_2, \Sigma_2) & \text{if } Z_i = 2 \\ X_i \in N(\mu_3, \Sigma_3) & \text{if } Z_i = 3 \end{cases}$$

Un-observed variables, Z_i, indicate cluster/distribution

$$\begin{cases} P(Z_i = 1) = \pi_1 \\ P(Z_i = 2) = \pi_2 \\ P(Z_i = 3) = \pi_3 \end{cases}$$

The probabilities sum to one,

$$\pi_1 + \pi_2 + \pi_3 = 1$$

What are the unknown model parameters?

Likelihood for the Gaussian mixture example

Parameters in the Gaussian mixture model are,

$$\theta = (\pi_1, \pi_2, \pi_3, \mu_1, \mu_2, \mu_3, \Sigma_1, \Sigma_2, \Sigma_3)$$

and

$$\mathbf{Z} = (Z_1, ..., Z_n) \qquad (\ n \text{ might be very large})$$

The likelihood is

$$L(\theta; \mathbf{x}, \mathbf{Z}) = \prod_{i=1}^{n} \sum_{j=1}^{3} \mathbb{I}_{\{Z_i=j\}} \pi_j f(x_i; \mu_j, \Sigma_j)$$

with ML estimate

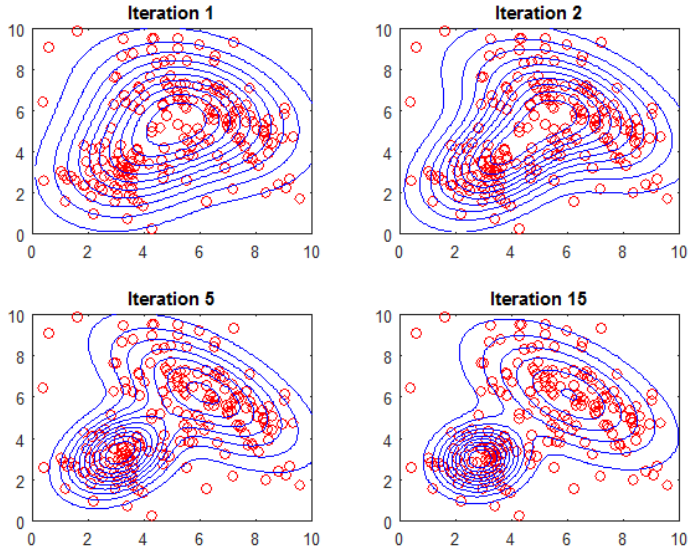
$$\theta_{ML} = \arg\max_{\theta, \mathbf{Z}} \log L(\theta; \mathbf{x}, \mathbf{Z})$$

Finding a solution $(\theta, \mathbf{Z})$ is much simplified by the **EM-algorithm**.

The Gaussian mixture EM-algorithm

1. Initialize means μ, covariances, Σ and mixing coefficients π.
2. **Expectation step**
   Calculate the responsibilities (conditional probabilities) γ_{ij} for Z_i belonging to cluster j using Bayes formula.
3. **Maximization step**
   Calculate weighted (using γ_{ij}) mean and covariance estimates μ Σ. Calculate mixing probabilities π_j based on means of the responsibilities over i.
4. Iterate until convergence

Gaussian mixture simulation



Simulation of a Gaussian Mixture model

$$P(Z_i = 1) = 0.3$$

$$P(Z_i = 2) = 0.7$$

$$X_i \in N\left(\begin{bmatrix} 3 \\ 3 \end{bmatrix}, \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}\right) \quad \text{if } Z_i = 1$$

$$X_i \in N\left(\begin{bmatrix} 6 \\ 6 \end{bmatrix}, \begin{bmatrix} 4 & -2 \\ -2 & 4 \end{bmatrix}\right) \quad \text{if } Z_i = 2$$

and $i = 1...200$

Some tricks

- High dimension (large p) and few data (small n)
  - Postulate same covariance structure for all clusters, $\Sigma = \Sigma_1 = ... = \Sigma_K$
  - Postulate diagonal covariance structure, $\Sigma = \text{diag}(\sigma_1, ..., \sigma_K)$
  - Regularize, $\Sigma = \Sigma + \lambda I$
  - Use first principal components instead of all dimensions in data
- Selecting number of clusters K
  - Clustering based on Gaussian model
  - Use information criteria (AIC or BIC) to select number of clusters
- The structure in data is described by μ, Σ and π

Exercise - K-means clustering

1. Run K-means clustering using kmeans_demo (Kmeans_demo.m, kmeans_demo.R, or kmean_demo.py). The demo fits K-means clustering to tree classes each sampled from a 2D mixture of Gaussian distributions, you can change the values of K in the code.
   - Try different numbers of clusters. Which value(s) of K seem reasonable? Why?

Exercise - Hierarchical clustering

2. Perform hierarchical clustering on the zip data (example 3 in ESL). The data consist of 400 samples of handwritten digits 0-9 in 16×16 grayscale images (= 256 features).
   - Try different dissimilarity measures. Which work best?
   - Where would you cut the dendrogram (ie how many clusters should we choose)?
   - You may use hierarchicalEx.m, HierarchicalEx.py, or hierarchicalEx.R.

Exercise - Gap statistic

3. Use the gap-statistic to select K for K-means clustering on the zip data.Use the gap-statistic to select K for K-means clustering on the zip data.
   - You will need to write the calculations of the within-class dissimilarities and the gap statistic yourself.
   - Use KmeansEx.m, KmeansEx.R, or kMeans_ex.py
   - You may use hierarchicalEx.m, HierarchicalEx.py, or hierarchicalEx.R.
   - Try first kmeans and then kmedoids (Optional).

Exercise - Gaussian mixture

4. We have data with four different measures from flowers of three different species (*Fisheriris.csv*). There are 50 observations of each species. See if you can identify three clusters in data using gaussian mixture modelling. (Two of the species are very similar)

Iris setosa

Iris versiocolor

Iris verginica

- Plot data using a scatterplot matrix
- Loop over different numbers of clusters
- Plot BIC/AIC for different model orders
- Notice the different extra parameters in the provided Gaussian Mixture function - they might be necessary.