Machine Learning
Clustering

1

Clustering methods

Objective ⇒ homogeneous groups of points (CLUSTERS):

- points within a cluster should be similar
- points from different clusters should be dissimilar

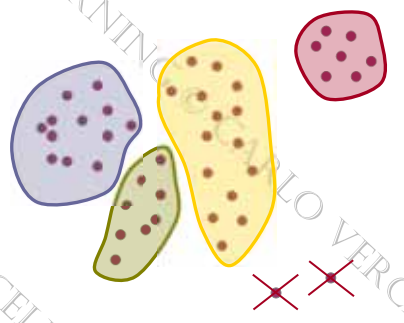
Can be used for:

providing meaningful interpretations

preliminary grouping of points

detecting outliers

selecting landmarks



2

Clustering methods

Several application domains…

- **Marketing**
  partition consumers into market segments (better understand the relationships between different groups of consumers/potential customers)
- **Social network analysis**
  recognize communities (hubs) within large groups of people
- **Genomics**
  build groups of genes with related expression patterns (often such groups contain functionally related proteins)
- **Social science**
  identify areas (hot spots) where there are greater incidences of similar types of crime to manage law enforcement resources

3

Clustering methods

Based on the logic used for building clusters we can have:

PARTITION METHODS
the data set is divided into a pre-fixed number of clusters

HIERARCHICAL METHODS
perform several partitions based on a tree structure

DENSITY-BASED METHODS
look at the number of points lying within the neighborhood of each point

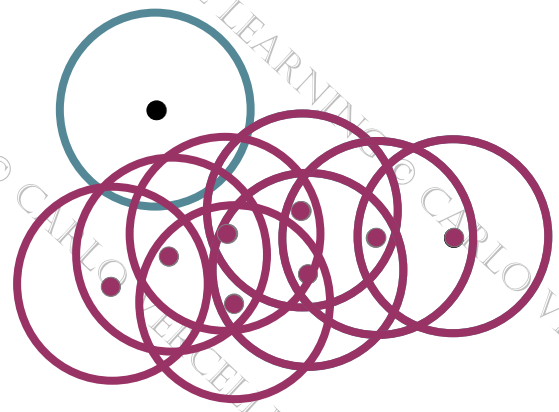
GRID METHODS
perform a preliminary partition based on a grid structure

4

Density based methods

Core points, Reachable points, Noise

It groups together points with many nearby neighbours.
Points that lie alone in low-density regions are marked as outliers.
The final model depends on two parameters: the radius of the neighborhood and the minimum number of points falling in it.
DBSCAN algorithm is deterministic (i.e. generates the same clusters when the same data in the same order are given).



5

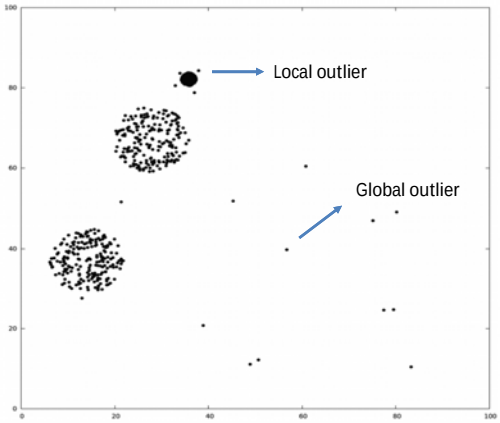
Density based methods

HDBSCAN algorithm (self-adjusting)
Outlier detection resorting to density-based clustering.

It identifies "local outliers", i.e. points that might be different from other parts in their local neighborhood but which are not necessarily global outliers.

Uses a range of distances to separate clusters of varying densities from sparser noise.

Example of a 2D dataset with global and local outliers



6

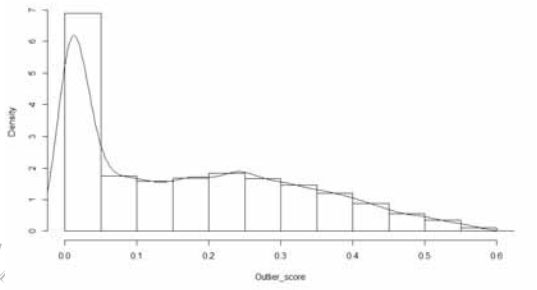
Density based methods

HDBSCAN algorithm
For each point a neighborhood density (ND) measure is computed.

The outlier score measures at what extent the ND of a point is far from the NDs of its neighbors.

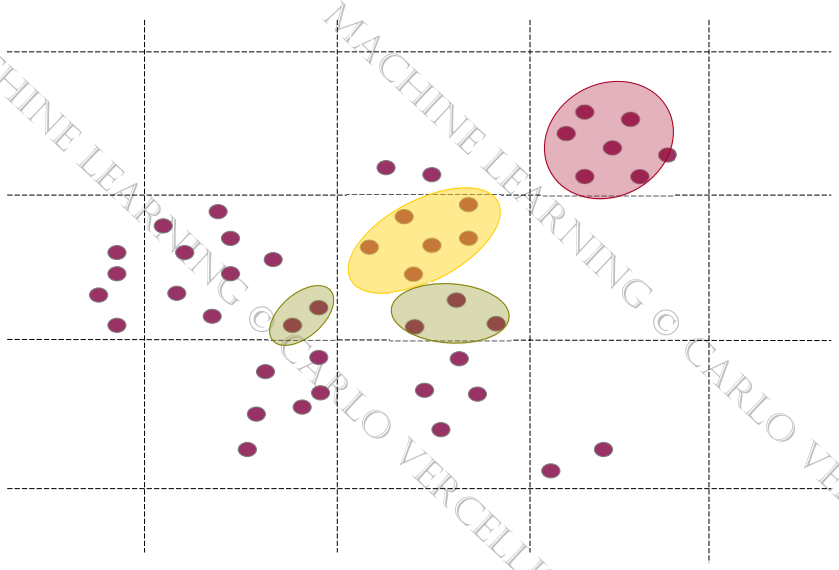
The higher the outlier score, the more likely the point is an outlier.

Distribution of the outlier scores



7

Grid methods



8

Clustering methods

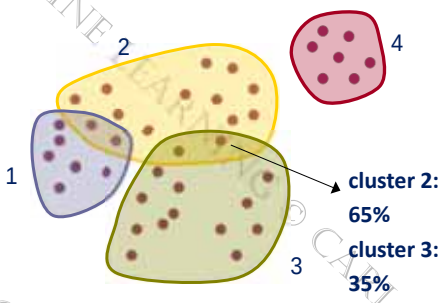
Based on the logic used for assigning the points we can have:

EXCLUSIVE METHODS

FUZZY METHODS

COMPLETE METHODS

PARTIAL METHODS



9

Clustering methods

General requirements:

FLEXIBILITY ⇒ numeric and categorical attributes

ROBUSTNESS ⇒ stability of the clusters (noise)

EFFICIENCY ⇒ small computing time

A huge number of possible partitions…

… clustering is *NP*- hard for $K \geq 3$

$$\frac{1}{K!}\sum_{h=1}^{K}\binom{K}{h}h^m$$ possible combinations

⇓

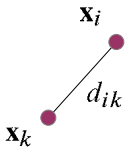
Most methods are heuristic in nature!

10

Affinity measures

DISTANCE MATRIX

$$D = [d_{ik}] = \begin{bmatrix} 0 & d_{12} & \cdots & d_{1,m-1} & d_{1m} \\ & 0 & \cdots & d_{2,m-1} & d_{2m} \\ & & \cdots & \vdots & \vdots \\ & & & 0 & d_{m-1,m} \\ & & & & 0 \end{bmatrix}$$



$$d_{ik} = \text{dist}(\mathbf{x}_i, \mathbf{x}_k) = \text{dist}(\mathbf{x}_k, \mathbf{x}_i), \quad i, k \in \mathcal{M}$$

SIMILARITY MEASURE

$$s_{ik} = \frac{1}{1 + d_{ik}} \qquad s_{ik} = \frac{d_{max} - d_{ik}}{d_{max}}$$

11

Affinity measures: numeric attributes

EUCLIDEAN DISTANCE:

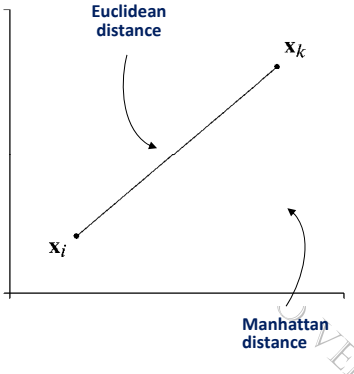
$$\text{dist}(\mathbf{x}_i, \mathbf{x}_k) = \sqrt{\sum_{j=1}^{n} (x_{ij} - x_{kj})^2}$$

MANHATTAN DISTANCE:

$$\text{dist}(\mathbf{x}_i, \mathbf{x}_k) = \sum_{j=1}^{n} |x_{ij} - x_{kj}|$$

MINKOWSKI DISTANCE:

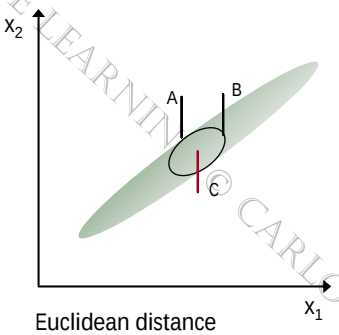
$$\text{dist}(\mathbf{x}_i, \mathbf{x}_k) = \sqrt[q]{\sum_{j=1}^{n} |x_{ij} - x_{kj}|^q}$$



12

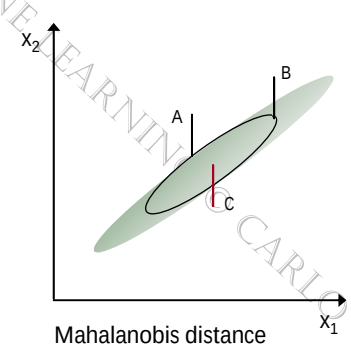
Affinity measures: numeric attributes

MAHALANOBIS DISTANCE: $\text{dist}(\mathbf{x}_i, \mathbf{x}_k) = \sqrt{(\mathbf{x}_i - \mathbf{x}_k)\mathbf{V}^{-1}(\mathbf{x}_i - \mathbf{x}_k)'}$



Euclidean distance

→ the set of points equidistant from a given location is a sphere



Mahalanobis distance

→ stretches the sphere to account for correlation among variables

13

Affinity measures: binary attributes

Contingency table

		point $\mathbf{x}_k$		
		0	1	totale
point $\mathbf{x}_i$	0	p	q	$p+q$
	1	u	v	$u+v$
	totale	$p+u$	$q+v$	n

COEFFICIENT OF SIMILARITY (symmetric attributes)

$$\text{dist}(\mathbf{x}_i, \mathbf{x}_k) = \frac{q+u}{p+q+u+v}$$

JACCARD DISTANCE (asymmetric attributes)

$$\text{dist}(\mathbf{x}_i, \mathbf{x}_k) = \frac{q+u}{q+u+v}$$

15

Affinity measures: categorical attributes

- Nominal attributes
  - Jaccard coefficient
  - extension of the coeff. of similarity $\text{dist}(\mathbf{x}_i, \mathbf{x}_k) = \frac{n - f}{n}$

- Ordinal attributes
  1) standardization $x'_{ij} = \frac{x_{ij} - 1}{H_j - 1}$
  2) distance measures for numeric attributes

Affinity measures: general case

In case of both numeric and categorical variables:

- define a binary index δ_{ikj} taking the value 0 when
  - at least x_{ij} or x_{kj} is missing
  - the attribute is binary and asymmetric $x_{ij} = x_{kj} = 0$

- define the coefficient Δ_{ikj} which indicates how much the attribute contributes to the similarity of points $\mathbf{x}_i$ and $\mathbf{x}_k$

  - if the attribute is binary or nominal, we set $\Delta_{ikj} = 0$ if $x_{ij} = x_{kj}$, otherwise $\Delta_{ikj} = 1$

  - if the attribute is numeric, we set $\Delta_{ikj} = \frac{|x_{ij} - x_{kj}|}{\max_l x_{lj}}$

- compute the similarity coefficients between $\mathbf{x}_i$ and $\mathbf{x}_k$

$$\text{dist}(\mathbf{x}_i, \mathbf{x}_k) = \frac{\sum_{j=1}^{n} \delta_{ikj} \Delta_{ikj}}{\sum_{j=1}^{n} \delta_{ikj}}$$

Partition methods

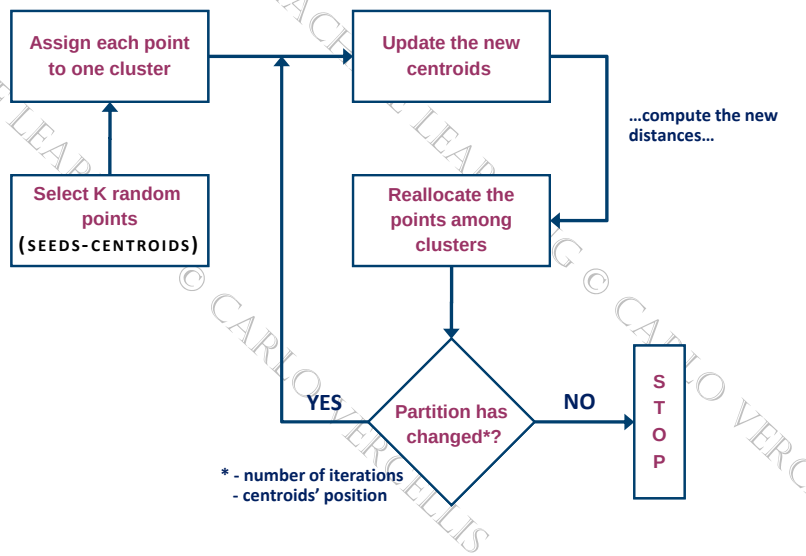
PARTITION METHODS: general framework

- initialization ⇒ points are divided into K non-empty groups (usually exhaustive and mutually exclusive)
- iteration ⇒ points are reassigned with the aim of improving the quality of the partition
- stop ⇒ no points are further reassigned (other stopping criteria)

Partition methods are greedy!

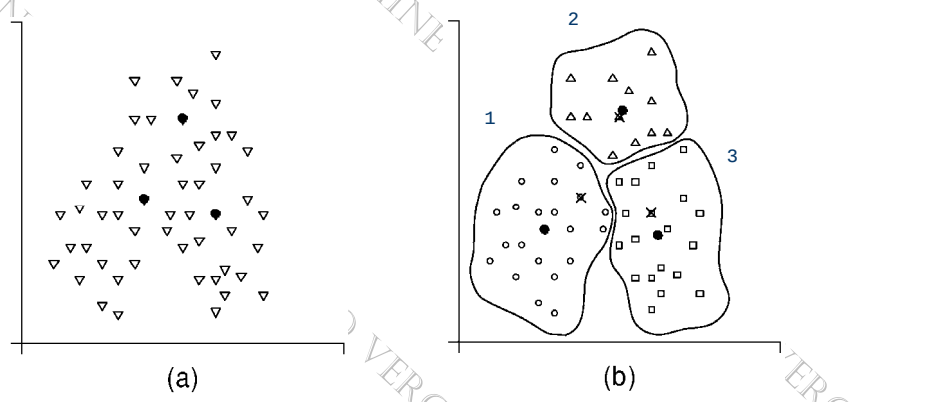
18

K-means algorithm



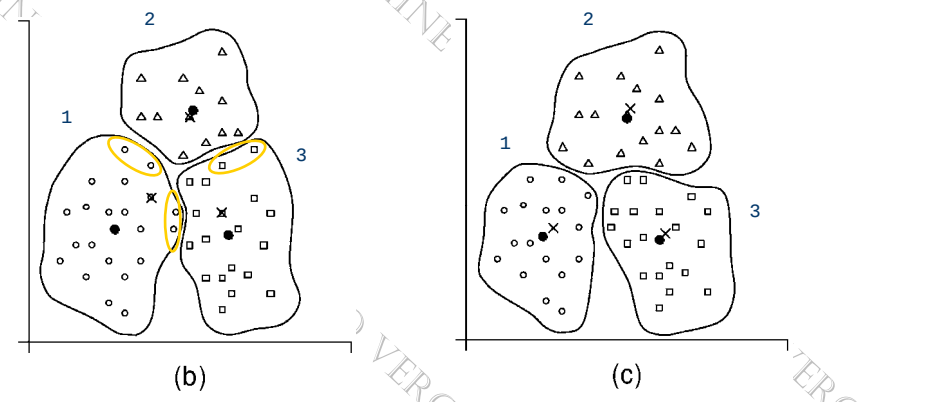
19

K-means algorithm



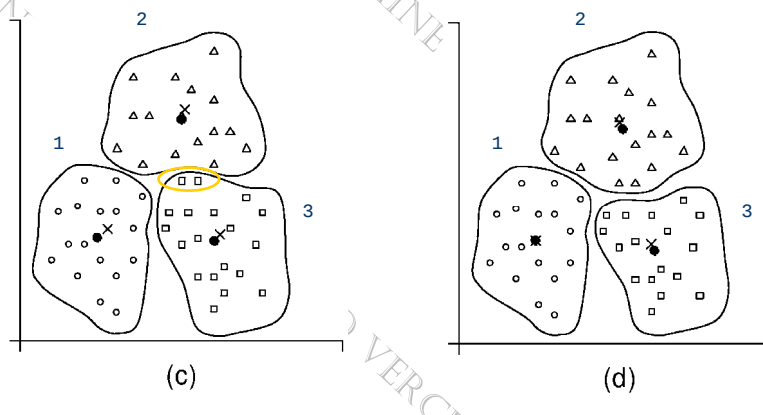
20

K-means algorithm



21

K-means algorithm



22

Comments on the K-means algorithm

STRENGTH AND WEAKNESS

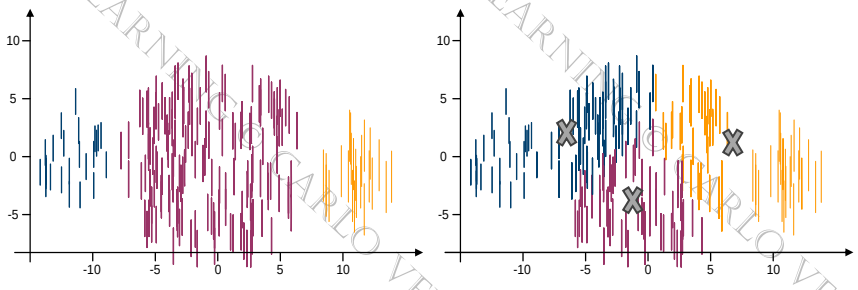
- relatively efficient: O(m*K*t*d) [m points, K clusters, t iterations, d variables]
- it converges for common similarity measures mentioned above (most of the convergence happens in the first few iterations)
- need to specify the number of clusters in advance
- often terminates at a local optimum (greedy algorithm)
- results may vary based on random seed selection, i.e. clusters may be different from one run to another (try out multiple starting points, FFT algorithm)
- applicable only when mean is defined…what about categorical data?
- unable to handle noisy data and outliers

24

Comments on the K-means algorithm

STRENGTH AND WEAKNESS

- it may have problems when (natural) clusters:
  - are of different sizes



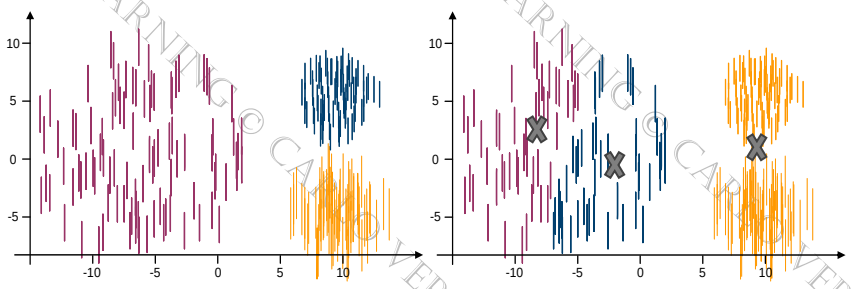
original data set K-means (3 clusters)

25

Comments on the K-means algorithm

STRENGTH AND WEAKNESS

- it may have problems when (natural) clusters:
  - are of different sizes
  - have different densities



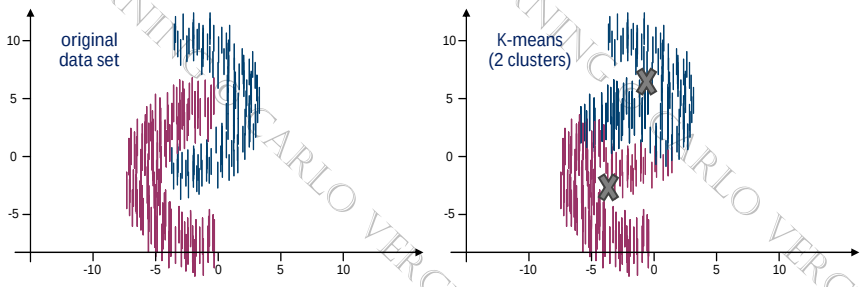
original data set K-means (3 clusters)

26

Comments on the K-means algorithm

STRENGTH AND WEAKNESS

- it may have problems when (natural) clusters:
  - are of different sizes
  - have different densities
  - have non-convex shapes



One solution is to use many clusters ⇒ find parts of clusters ⇒ need to put them together

27

Pre and post-processing practical suggestions

PRE-PROCESSING

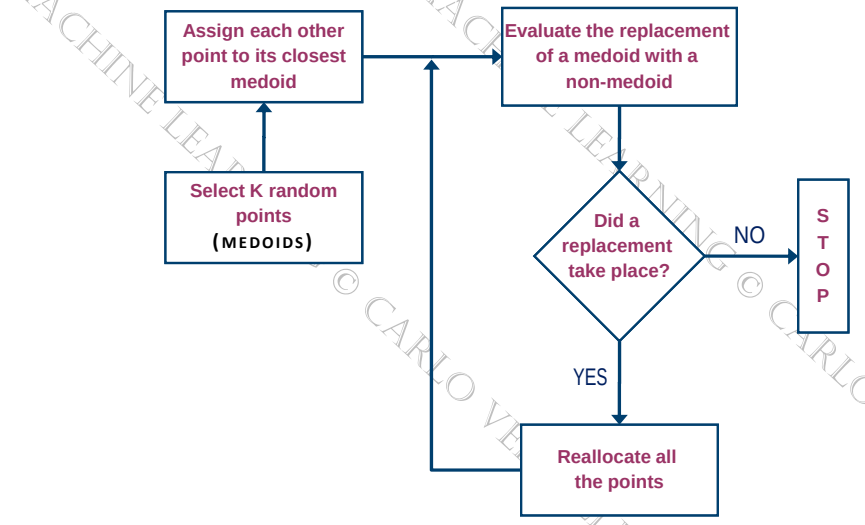
- normalize (or standardize) the data
- eliminate outliers

POST-PROCESSING

- eliminate small clusters that may represent outliers
- split "loose" clusters, i.e. clusters with relatively high SSE
- merge clusters that are "close" and have relatively low SSE

28

K-medoids algorithm



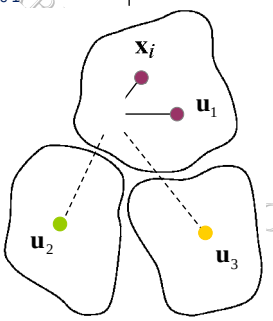
29

K-medoids algorithm

Each medoid's replacement is evaluated as follows:

- consider all the pair of points $(\mathbf{x}_i, \mathbf{u}_h)$
- evaluate how a third point $\mathbf{x}_k$ contribute to the exchange between $\mathbf{x}_i$ and $\mathbf{u}_h$

case 1



$\mathbf{x}_k$ would be assigned to the new medoid $\mathbf{x}_i$

The contribution to the exchange is

$$R_{ihk} = \text{dist}(\mathbf{x}_i, \mathbf{x}_k) - \text{dist}(\mathbf{u}_h, \mathbf{x}_k)$$

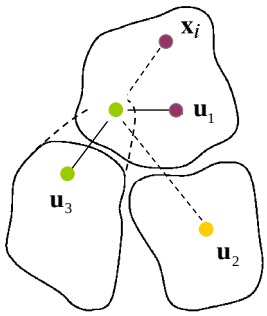
↓

it can be positive, negative or zero

30

K-medoids algorithm

case 2



$\mathbf{x}_k$ would be assigned to a different cluster

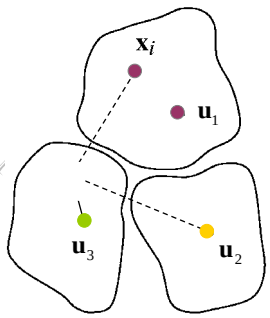
The contribution to the exchange is

$$R_{ihk} = \min_{\mathbf{u}_e \in U, e \neq h} \text{dist}(\mathbf{u}_e, \mathbf{x}_k) - \text{dist}(\mathbf{u}_h, \mathbf{x}_k)$$

↓

it is nonnegative

case 3



$\mathbf{x}_k$ remains in its cluster

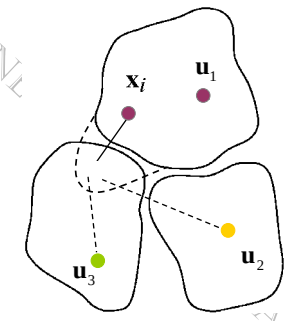
The contribution to the exchange is

$$R_{ihk} = 0$$

31

K-medoids algorithm

case 4



$\mathbf{x}_k$ would be assigned to the new medoid $\mathbf{x}_i$

The contribution to the exchange is

$$R_{ihk} = \text{dist}(\mathbf{x}_i, \mathbf{x}_k) - \min_{\mathbf{u}_e \in U, e \neq h} \text{dist}(\mathbf{u}_e, \mathbf{x}_k)$$

↓

it is nonpositive

For each pair $(\mathbf{x}_i, \mathbf{u}_h)$ the overall contribution is computed:

$$T_{ih} = \sum_{\mathbf{x}_k \notin U} R_{ihk}$$

The pair giving raise to the minimum of this contribution is exchanged (provided it is negative)

32

Hierarchical methods

HIERARCHICAL METHODS

- are based on a tree structure (dendrogram)
- use the distances among points to derive clusters merging or splitting
- do not require the number K of clusters as an input

AGGLOMERATIVE ALGORITHMS (bottom-up techniques)

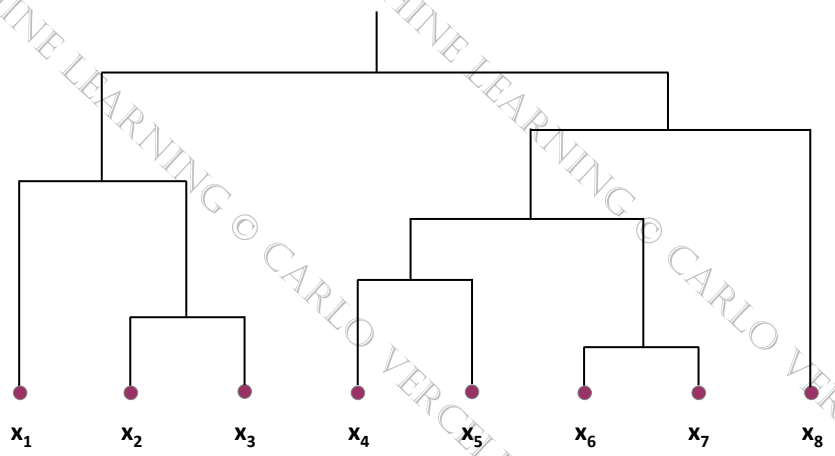
- initially each point represents a single cluster
- iteratively the two clusters with the minimum distance are merged together and the proximity matrix (distances among all clusters) is updated
- all points are comprised in a single cluster ⇒ stop

DIVISIVE ALGORITHMS (top-down techniques)

- initially all points are comprised in a single cluster
- iteratively choose a cluster and split it so to obtain two clusters with the maximum distance
- each point represents a single cluster ⇒ stop

Agglomerative hierarchical methods

Iteration 0: 8 clusters (init)



Agglomerative hierarchical methods

At each iteration the two clusters with the minimum distance are merged

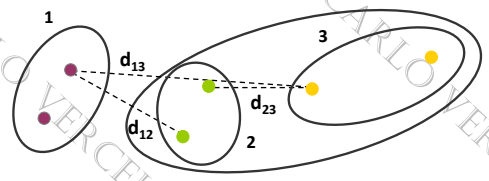
Several metrics can be used to evaluate the proximity (distance) between a pair of clusters

MINIMUM DISTANCE (SINGLE LINKAGE)

The distance of two clusters is based on the two most similar (closest) points in the different clusters

$$\text{dist}(C_h, C_f) = \min_{\substack{\mathbf{x}_i \in C_h \\ \mathbf{x}_k \in C_f}} \text{dist}(\mathbf{x}_i, \mathbf{x}_k)$$

- sensitive to noise and outliers
- biased towards elliptical clusters



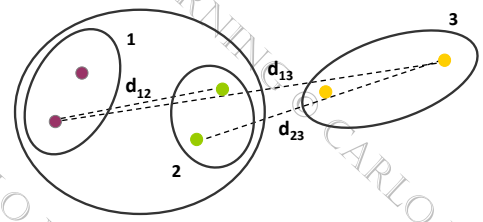
Agglomerative hierarchical methods

MAXIMUM DISTANCE (COMPLETE LINKAGE)

The distance of two clusters is based on the two least similar (most distant) points in the different clusters

$$\text{dist}(C_h, C_f) = \max_{\substack{\mathbf{x}_i \in C_h \\ \mathbf{x}_k \in C_f}} \text{dist}(\mathbf{x}_i, \mathbf{x}_k)$$

- less sensitive to noise and outliers
- tends to break large clusters
- biased towards globular clusters



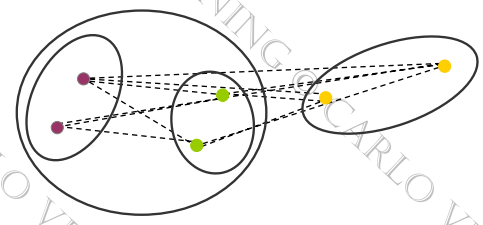
Agglomerative hierarchical methods

AVERAGE DISTANCE

The distance of two clusters is the average of the sum of the pairwise distances of the points in the two clusters

$$\text{dist}(C_h, C_f) = \frac{\sum_{\mathbf{x}_i \in C_h} \sum_{\mathbf{x}_k \in C_f} \text{dist}(\mathbf{x}_i, \mathbf{x}_k)}{\text{card}\{C_h\}\,\text{card}\{C_f\}}$$

- less sensitive to noise and outliers
- biased towards globular clusters



Agglomerative hierarchical methods

WARD'S MINIMUM VARIANCE METHOD

It minimizes the total within-cluster variance

- at the initial step all clusters are singletons
- compute the squared Euclidean distances among all the points
- compute the increase in total within-cluster variance for all possible merging

  Let d_{ik} d_{ij} and d_{jk} be the pairwise distances between the clusters C_i, C_j and C_k

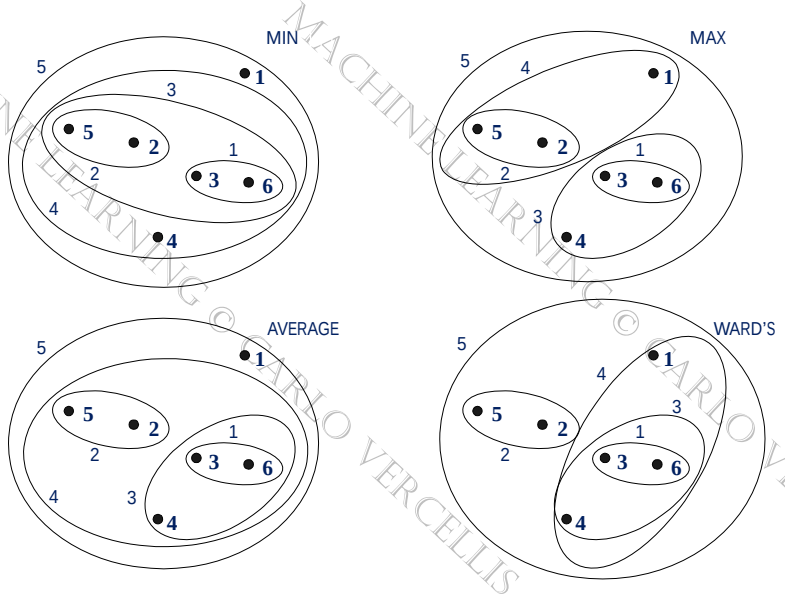
  Let $d_{(ij)k}$ the distance (within-cluster variance) between the new cluster $C_i \cup C_j$ and C_k, where $d_{(ij)k} = \alpha_i \cdot d_{ik} + \alpha_j \cdot d_{jk} + \beta \cdot d_{ij} + \gamma\,|d_{ik} - d_{jk}|$

  Parameters depend on clusters size.

- merge the two clusters leading to the minimum increase of within-cluster variance

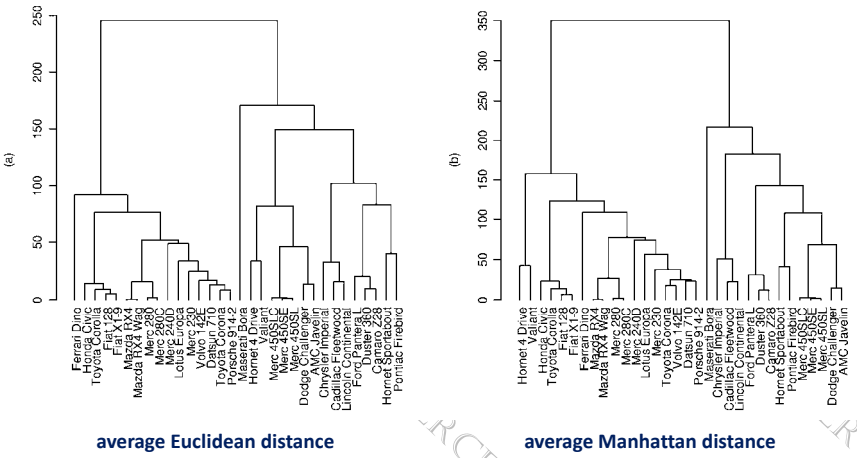
- less sensitive to noise and outliers
- biased towards globular clusters

Hierarchical clustering comparison



41

Agglomerative hierarchical methods

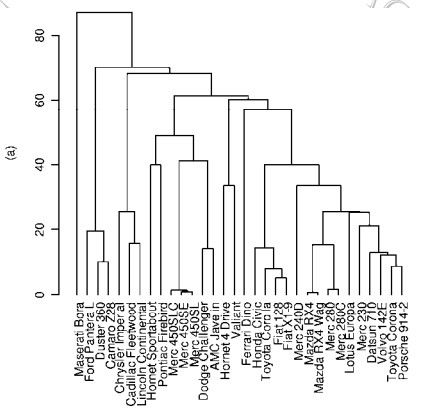


average Euclidean distance

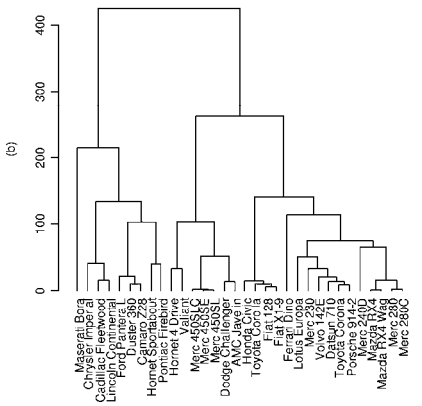
average Manhattan distance

42

Agglomerative hierarchical methods



minimum Euclidean distance

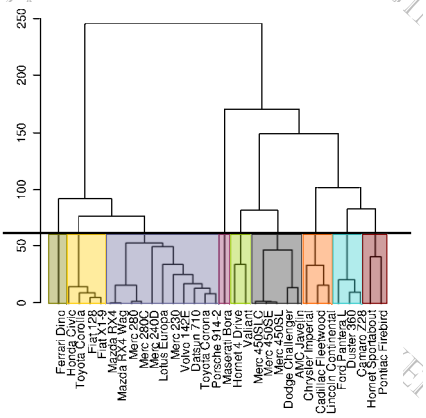


maximum Euclidean distance

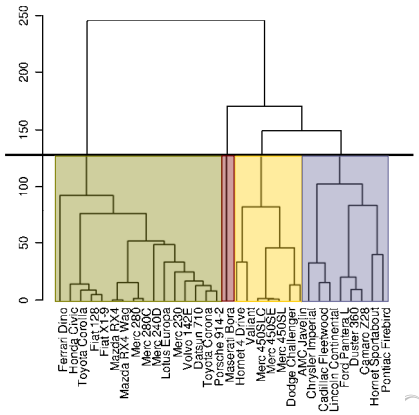
43

Agglomerative hierarchical methods

Any desired number of clusters can be obtained by "cutting" the dendogram at the proper level



average Euclidean distance
9 clusters cut



average Euclidean distance
4 clusters cut

44

How many clusters?

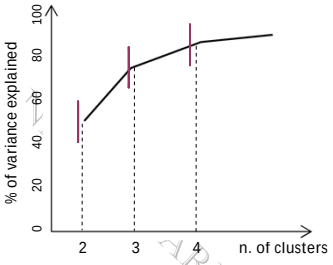
Alternative methods are available…

RULE OF THUMB $K \approx \left(\frac{m}{2}\right)^{1/2}$

ELBOW METHOD

The percentage of variance explained* is set as a function of the number of clusters

*Is the ratio of the within-cluster variance to the total variance



SILHOUETTE-BASED METHOD

Choose the number of clusters giving rise to the largest average silhouette

46

Clustering validity: internal measures

Internal measures: used to measure the goodness of a clustering structure independently of external information

COHESION of each cluster measures how closely related are objects in a cluster

$$\text{coes}(C_h) = \sum_{\substack{\mathbf{x}_i \in C_h \\ \mathbf{x}_k \in C_h}} \text{dist}(\mathbf{x}_i, \mathbf{x}_k)$$

SEPARATION of a pair of clusters measure how distinct or well-separated a cluster is from other cluster

$$\text{sep}(C_h, C_f) = \sum_{\substack{\mathbf{x}_i \in C_h \\ \mathbf{x}_k \in C_f}} \text{dist}(\mathbf{x}_i, \mathbf{x}_k)$$

OVERALL COHESION

$$\text{coes}(\mathcal{C}) = \sum_{C_h \in \mathcal{C}} \text{coes}(C_h)$$

OVERALL SEPARATION

$$\text{sep}(\mathcal{C}) = \sum_{\substack{C_h \in \mathcal{C} \\ C_f \in \mathcal{C}}} \text{sep}(C_h, C_f)$$

The lower the cohesion and the higher the separation are, the better the clustering is.

48

Clustering evaluation

SILHOUETTE COEFFICIENT:

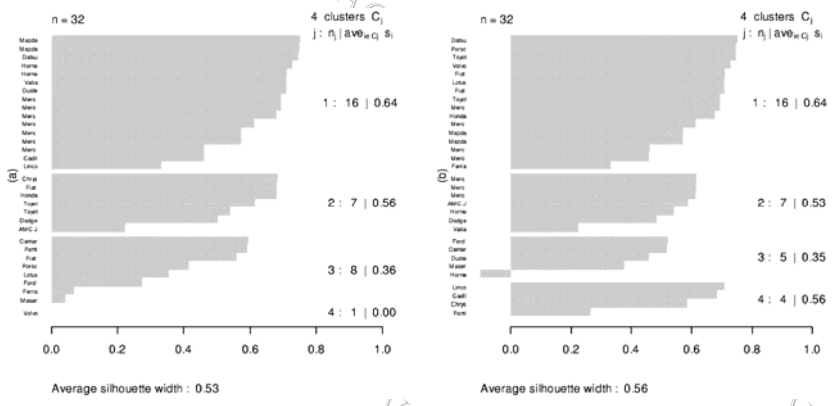
- compute the average distance u_i of $\mathbf{x}_i$ from all the other points in the same cluster
- compute the average distance of $\mathbf{x}_i$ from all the other points comprised in a different cluster
  (let v_i be the minimum of these distances)
- the silhouette coefficient is given by

$$\text{silh}(\mathbf{x}_i) = \frac{v_i - u_i}{\max(u_i, v_i)}$$

- within [-1,1]
- the closer to 1 the better
- average silhouette

49

Clustering evaluation



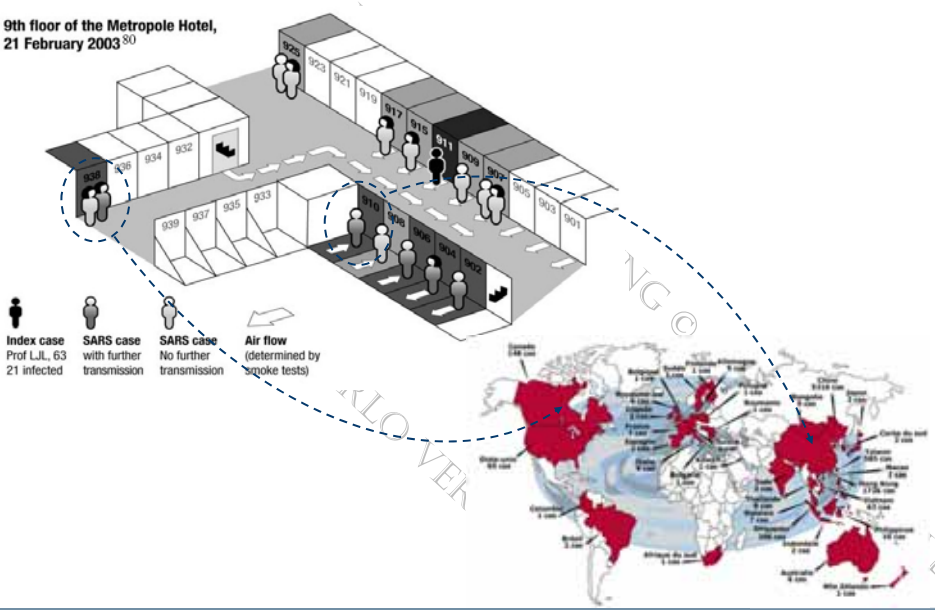
50

SARS 1: origin and evolution

- **November 16, 2002**
  A 45-year-old man in Guangdong Province (100 km from Hong Kong) becomes ill with an unusual respiratory illness. First patient of SARS (Severe Acute Respiratory Syndrome).
- **November 27, 2002**
  There are some alerts, but for various reasons they do not register as clearly as they should have.
  The WHO (World Health Organization) receives a Chinese-language news report of a flu outbreak in China. The report has an English heading…but it is not fully translated.
- **January 2003**
  WHO receives an email describing a "strange contagious disease" that has "already left more than 305 people dead" in Guangdong Province.
- **February 2003**
  Dott. LJL, physician and professor of nephrology at Zhongshan University, leaves to Hong Kong.

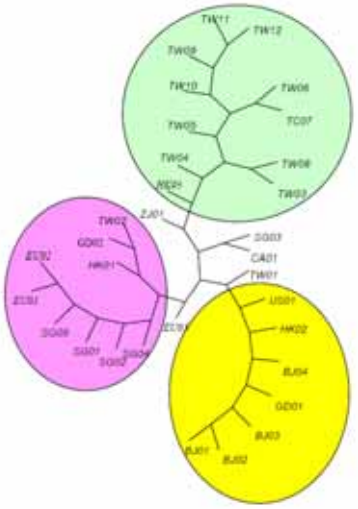
53

SARS 1: origin and evolution

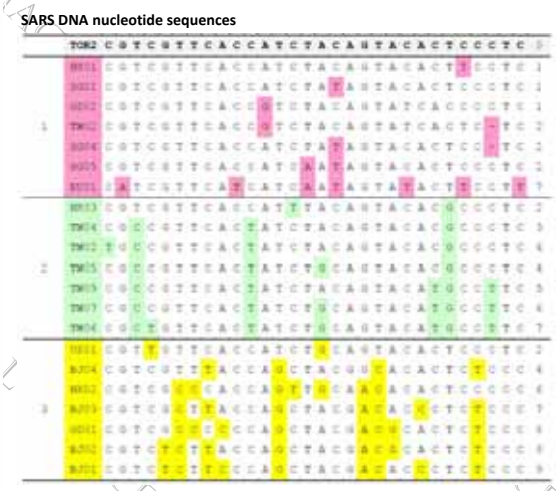


54

Phylogenetic analysis of the virus



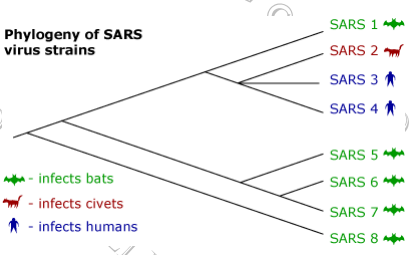
SARS DNA nucleotide sequences



55

Phylogenetic analysis of the virus

* Caged animals, including Himalayan palm civets and a raccoon dog, from wet markets in mainland China are the reservoir for the origin of the SARS 1 epidemic
* Several lines of evidence suggested that the civet may have served only as an amplification host for SARS 1 virus
* A coronavirus similar to SARS 1 was found in nearly 40 per cent of wild Chinese bats they examined
* Genetic analysis of the bat SARS virus showed it was closely related to the human SARS 1 coronavirus
* The researchers could not determine how the bats were originally infected or they transmitted the SARS coronavirus to other mammals including civets



56