Outline

Outline

1. Introduction

1.1. Objectives

Objectives of the lesson

- Know the basics of clustering.
- Know the fundamental characteristics of partitional, hierarchical and density-based clustering.
- Know how to apply the different types of clustering.
- Interpret the results.

Objectives of unsupervised learning

- Unsupervised learning methods do not need labelled data.
- They try to discover the structure of the data.
- It can be a goal, or a means to an end.
- There are different objectives within the framework of unsupervised learning.

Objectives of unsupervised learning

- Clustering: given two unlabelled examples (no field or class label), group them according to predefined criteria.
- Hierarchy generation: given data at the same level, generate hierarchies that organise the data.
- Dimensionality reduction: given data, reduce the dimension or number of attributes characterising the data.
- Visualisation: given data with complex representation, allow its visualisation.

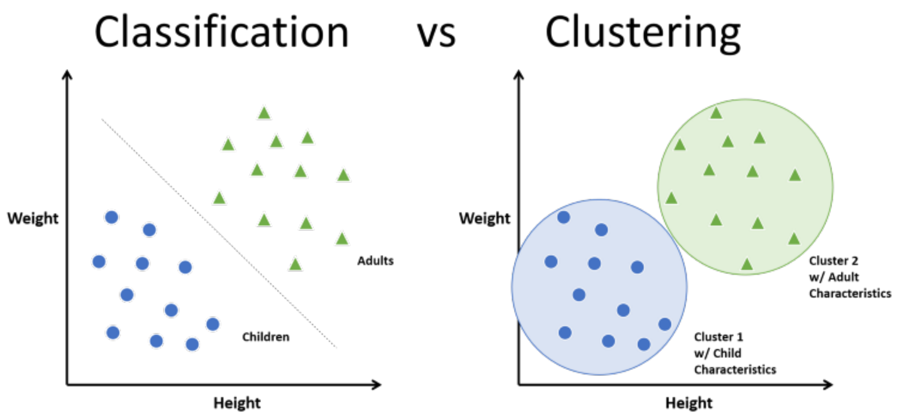
1.2. Clustering

Clustering definition

- Objective:
  - To group instances or patterns by creating similar clusters.
  - Segment a heterogeneous population into some homogeneous subgroups or clusters.
  - That is, given two unlabelled examples (no class field), group them according to some predefined criteria. Such a criterion usually comes from:
    - Parametric learning: assumed parameters, e.g., the data follow a specific probability density function.
    - Non-parametric learning: some measure of distance.
- Cluster:
  - Group or set of objects or patterns.
  - Similar to each other (to those of its cluster).
  - Different from those of another cluster.

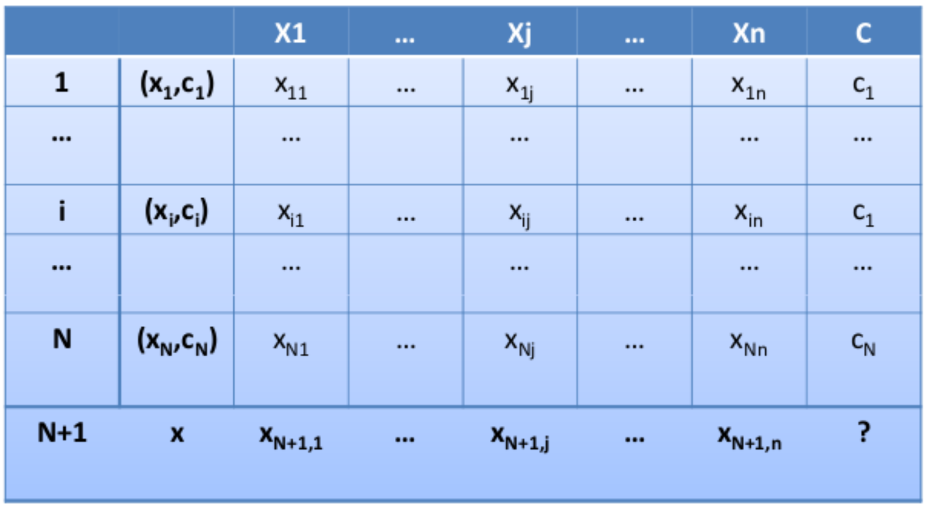
Clustering vs classification

- Both try to assign the appropriate class or cluster to a given example.
- However, in clustering, there are no predefined classes, so the aim is to group the data rather than to develop classifiers (no data partitioning will take place, and the evaluation of a given methodology will be more complicated).


Classification vs Clustering
Weight
Height
Children
Adults
Weight
Height
Cluster 1 w/ Child Characteristics
Cluster 2 w/ Adult Characteristics

Clustering vs classification

- In classification, we are given the label.
- The objective is to learn a relationship between the label and the rest of the variables, which leads us to predict the label of new examples ($N + 1$).

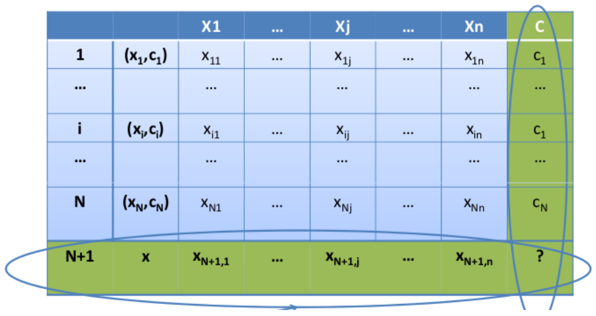


		X1	...	Xj	...	Xn	C
1	(x_1,c_1)	x_{11}	...	x_{1j}	...	x_{1n}	c_1
...		...		...		...	...
i	(x_i,c_i)	x_{i1}	...	x_{ij}	...	x_{in}	c_1
...		...		...		...	...
N	(x_N,c_N)	x_{N1}	...	x_{Nj}	...	x_{Nn}	c_N
N+1	x	$x_{N+1,1}$	...	$x_{N+1,j}$	...	$x_{N+1,n}$	?

Clustering vs classification

- In clustering, natural groupings are found in an unlabelled (unsupervised) data set.

- Describe them in terms of classes or groups of data with strong internal similarities (similar).
- Homogeneity within classes and heterogeneity between classes.
- "Art of finding clusters".
- Sometimes, it is asked to predict the label of a new example after the labelling process.



		X1	...	Xj	...	Xn	C
1	(x_1,c_1)	x_{11}	...	x_{1j}	...	x_{1n}	c_1
...		...		...		...	...
i	(x_i,c_i)	x_{i1}	...	x_{ij}	...	x_{in}	c_1
...		...		...		...	...
N	(x_N,c_N)	x_{N1}	...	x_{Nj}	...	x_{Nn}	c_N
N+1	x	$x_{N+1,1}$	...	$x_{N+1,j}$	...	$x_{N+1,n}$	?

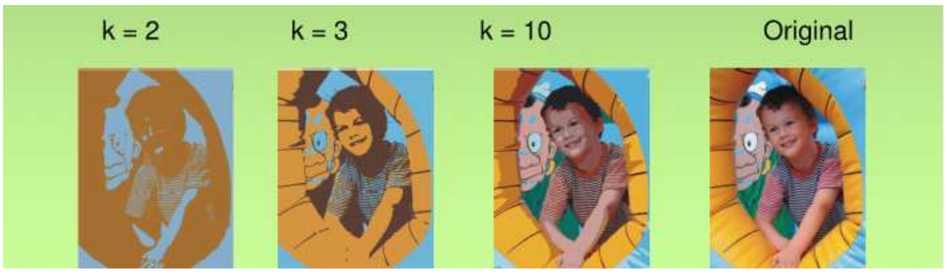
Applications

There are different fields where clustering is applied:

- Marketing: identifying groups of customers with similar behaviour or product positioning.
- Internet: grouping of texts, videos, images...
- Social networks: identifying communities.
- Recommendation systems: recommending products similar to personal tastes.

Example of real application

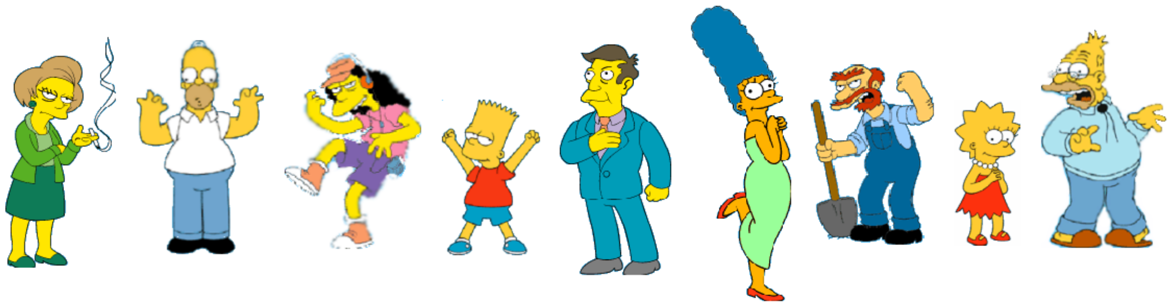
- Segment = partition the image into regions of homogeneous visual appearance.
- Point = pixel with intensity R,G,B, given by the centroid to which the pixel has been assigned.
- The image is represented using a palette of only k colours.
- It does not take into account the spatial proximity of pixels and works quite well!



1.3. Key Issues

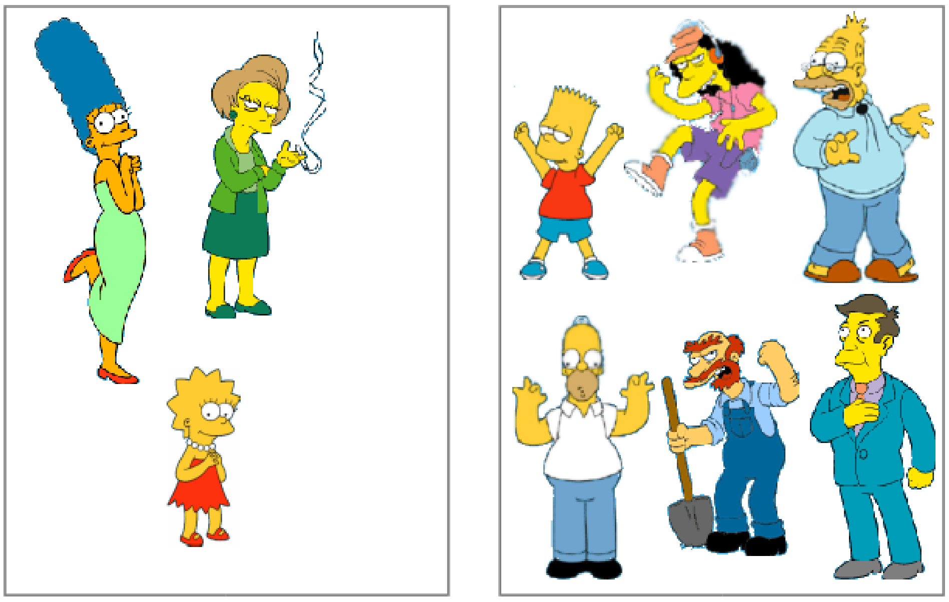
Subjectivity

What would be the natural way to group the Simpsons characters?



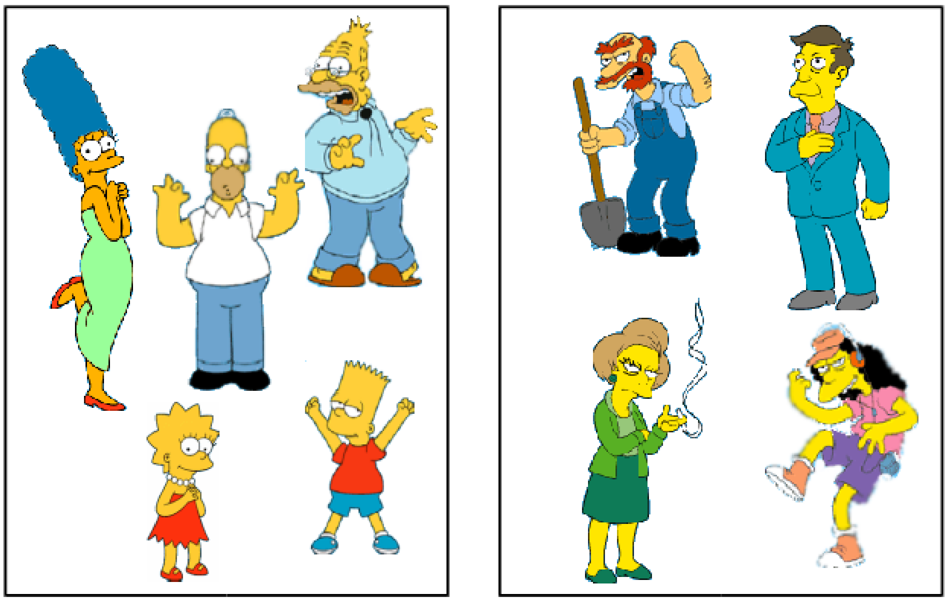
Subjectivity

Women vs Men



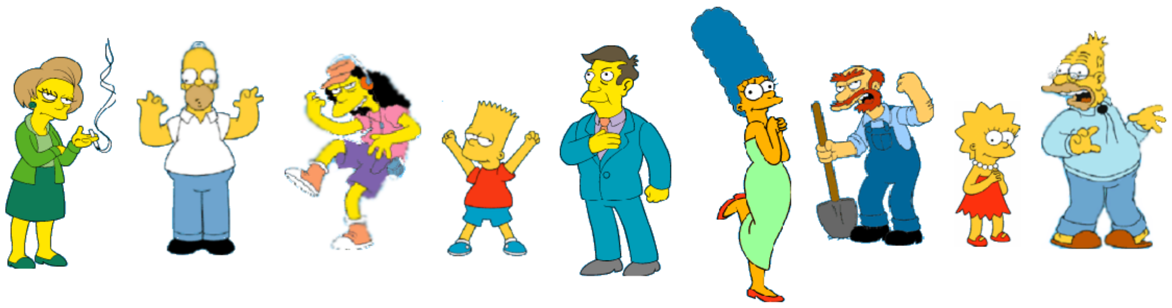
Subjectivity

Simpsons family vs school workers

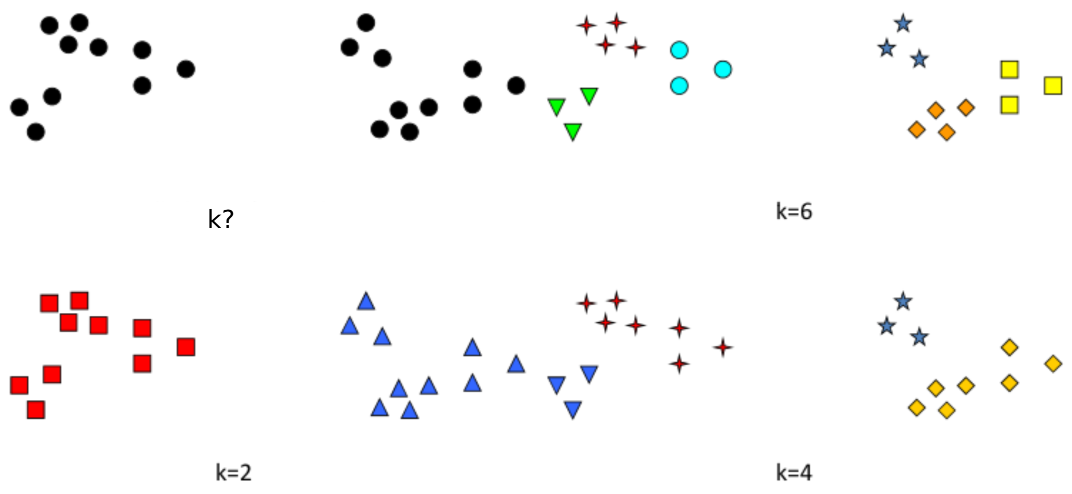


Subjectivity

- Depending on the objective, one option is better than another.
- Clustering is subjective.



Number of clusters



Similarity meassure

- We talk about similarity between instances, but what is similarity, and how do we measure it?
- Usually, it is expressed in terms of distances, where $d(i,j) > d(i,k)$ tells us that the object i is more similar to k than to j.
- There should be closer distances within the cluster than outside the group.
- The definition of the similarity/distance metric will differ depending on the type of data and the semantic interpretation we make. In other words, the similarity between objects is subjective.

Distances

- Minkowski distance (Lp-norm):

$$d(\mathbf{x},\mathbf{y}) = \left(\sum_{i=1}^{k} |x_i - y_i|^p\right)^{1/p}.$$

- Euclidean distance ($p = 2$): square root of the squared differences of the characteristics of two patterns.

$$d(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}.$$

Distances

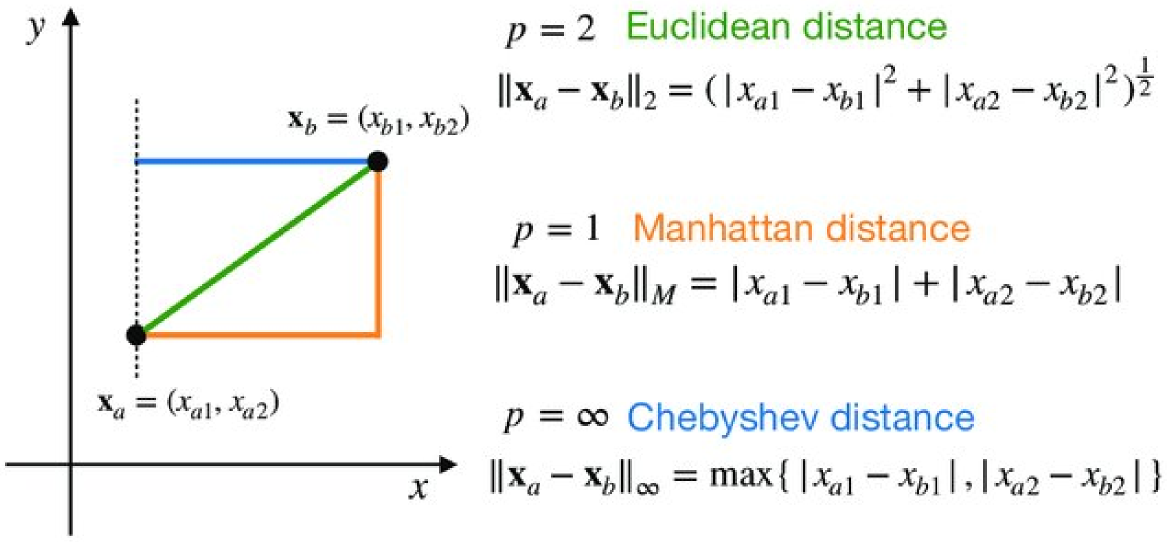
- Manhattan distance ($p = 1$): sum of the absolute differences in the characteristics of the two patterns.

$$d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{k} |x_i - y_i|.$$

- Tchebychev distance ($p \to \infty$): the greatest of its differences along any of its characteristic dimensions.

$$d(\mathbf{x}, \mathbf{y}) = max_{i=1}^{k}(|x_i - y_i|).$$

Distances


y
x
$\mathbf{x}_b = (x_{b1}, x_{b2})$
$\mathbf{x}_a = (x_{a1}, x_{a2})$
$p = 2$ Euclidean distance
$\|\mathbf{x}_a - \mathbf{x}_b\|_2 = (|x_{a1} - x_{b1}|^2 + |x_{a2} - x_{b2}|^2)^{\frac{1}{2}}$
$p = 1$ Manhattan distance
$\|\mathbf{x}_a - \mathbf{x}_b\|_M = |x_{a1} - x_{b1}| + |x_{a2} - x_{b2}|$
$p = \infty$ Chebyshev distance
$\|\mathbf{x}_a - \mathbf{x}_b\|_\infty = \max\{|x_{a1} - x_{b1}|, |x_{a2} - x_{b2}|\}$

Source: Granular Classification for Imbalanced Datasets: A Minkowski Distance-Based Method

1.4. Validation

Validation

External and internal validation are the two most important categories for clustering validation:

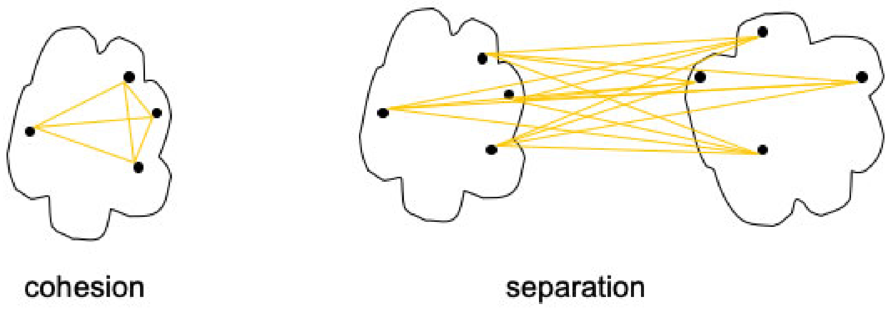
- Internal validation measures clustering solely based on information from the data. They assess how good the structure of the clustering is.

- External validation measures the quality of the clustering with external information beforehand. The class labels are known, which are usually not available.

Internal validation

Since clustering aims to obtain groups of objects that are similar to each other but different to objects in other clusters, the internal validation metrics are based on:

- Cohesion: each cluster member should be as close as possible to the other members of the same group.
- Separation: clusters should be widely separated from each other.

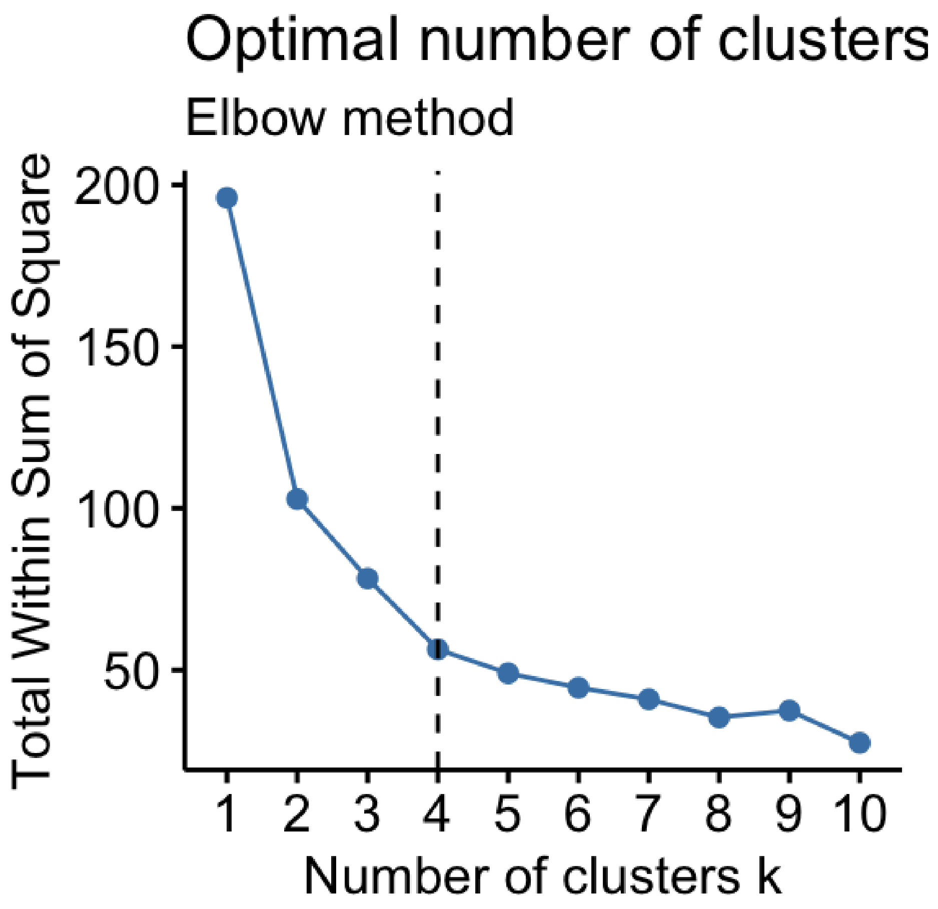


Internal validation: SSE

Sum of Squared Errors or Sum of Squared Within

$$SSE = \sum_{i=1}^{k} \sum_{x \in C_i} d(\mathbf{m_i}, \mathbf{x})^2,$$

where k is the number of clusters, $\mathbf{x}$ is a pattern of C_i, and $\mathbf{m_i}$ is the centroid of cluster C_i.



Internal validation: DB

Davies-Bouldin index

$$DB = \frac{1}{k}\sum_{i=1}^{k} max_{i \neq j} \frac{\alpha_i + \alpha_j}{d(\mathbf{m_i}, \mathbf{m_j})},$$

where α_i is the average distance of all elements in cluster C_i to centroid $\mathbf{m_i}$, and $d(\mathbf{m_i}, \mathbf{m_j})$ is the distance between centroids C_i and C_j. This index has to be minimised.

Internal validation: other metrics

- Calinski and Harabasz.
- Dunn.
- Silhouette.
- COP.

It would be good to have a look at all of them.

External validation: RI

Rand Index: is a measure of the similarity between two data clusterings.

$$RI = \frac{a+b}{\binom{n}{2}},$$

where a is every time a pair of elements is grouped together by the two clustering assignations, b is every time a pair of elements is not grouped together, and n is the number of possible pairs.

External validation: ARI

Adjusted Rand Index: is a measure of the similarity between two data clusterings. Concretely, it is the corrected-for-chance version of the Rand index.

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - [\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}]/\binom{n}{2}}{\frac{1}{2}[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}] - [\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}]/\binom{n}{2}}.$$

	Y_1	Y_2	$\dots$	Y_k	$Sums$
X_1	n_{11}	n_{12}	$\dots$	n_{1k}	a_1
X_2	n_{21}	n_{22}	$\dots$	n_{2k}	a_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
X_k	n_{k1}	n_{k2}	$\dots$	n_{kk}	a_k
$Sums$	b_1	b_2	$\dots$	b_k	

External validation: MI

Mutual Information: is a measure of the mutual dependence of two assignations for comparing clusterings.

$$MI = \sum_i \sum_j p_{ij} \ln \frac{p_{ij}}{p_i p_j},$$

where $p_{ij} = \frac{n_{ij}}{n}$, $p_i = \frac{n_i}{n}$ and $p_j = \frac{n_j}{n}$.

Seeks information about Adjusted Mutual Information.

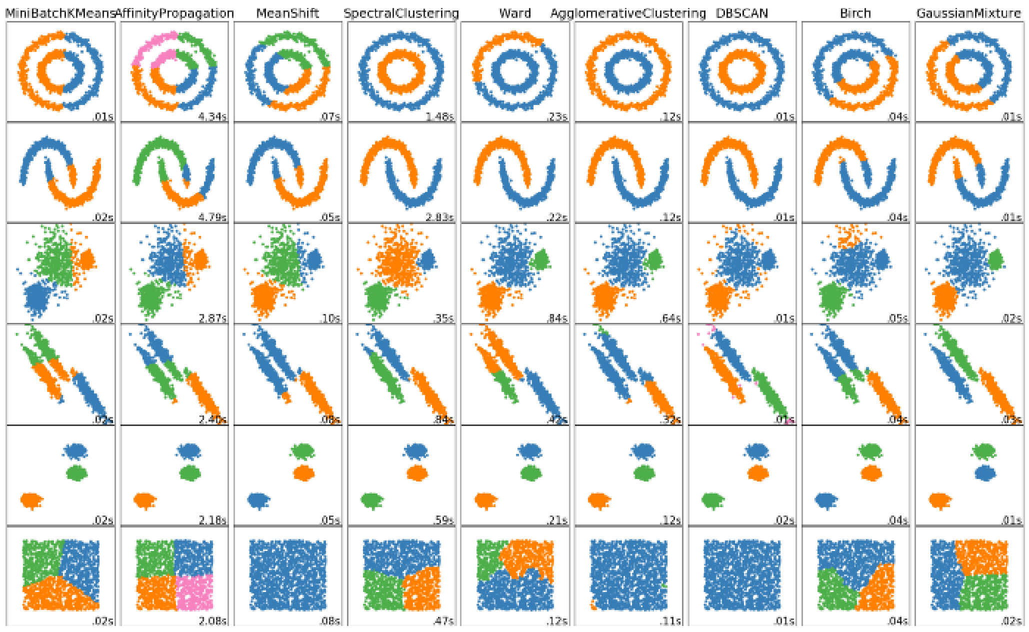
1.5. Different Techniques

Types

- Hierarchical: assumes that data can be naturally grouped in a tree structure.
- Non-hierarchical:
  - Partitional: form natural partitions of the data into a possibly prefixed number of clusters. Similar problem to classification but with unlabelled data. It is usually based on optimisation techniques.
  - Probabilistic: usually assumes that the conditional cluster densities have some known parametric shape (e.g. Gaussian) and thus estimates the unknown parameters (by maximum likelihood).

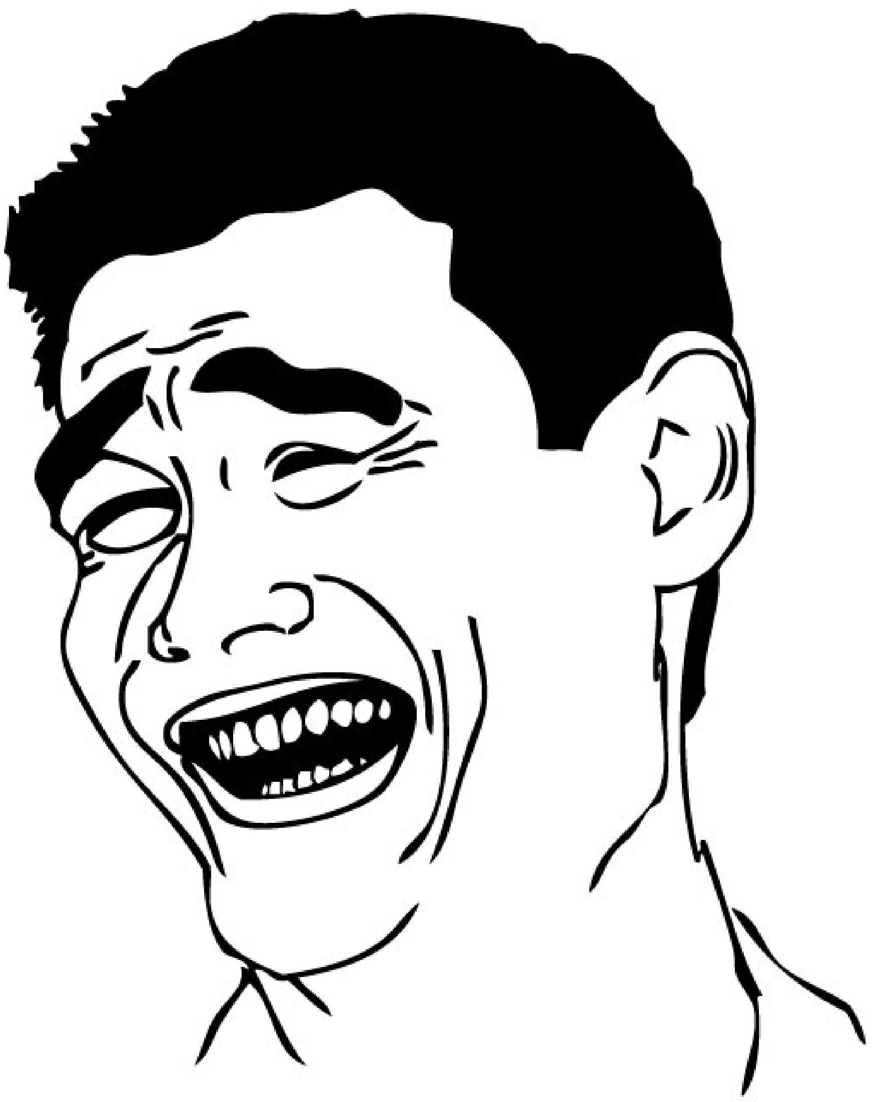
Types

Different types of clustering will lead to different clusters → Study the problem well.



Best technique

- Be scalable to the size of the data.
- Handle different data types.
- Identify clusters with arbitrary shapes.
- Have a minimum number of parameters.
- Be tolerant to noise and outliers.



- Be independent of the order of presentation of training patterns.
- Allow working with spaces with many different dimensions.
- Be interpretable.

Steps for clustering analysis

1. Select a similarity measure.
2. Choose the technique to use (hierarchical, non-hierarchical).
3. Choose a method/algorithm within the technique.
4. [Decide how many clusters to make] (depends).

5. Interpret the clusters formed (deduce the properties that divide the individuals in this way).

2. Partitional Clustering

2.1. Problem Definition

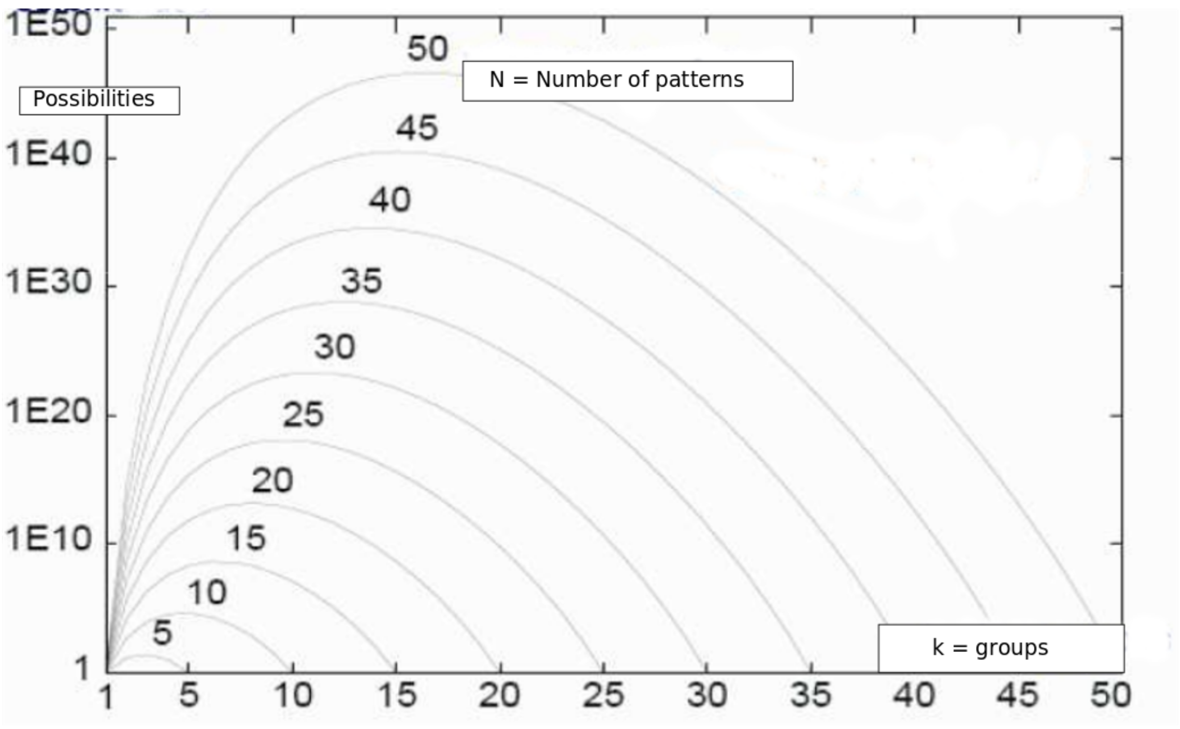
Partitional clustering

- Objective: partitioning of objects (patterns) into disjoint clusters.
- Let $D = \{x_1, x_2, ..., x_n\}$ be a set of N objects, we partition D into k clusters $C_1, \ldots, C_k$.
- It is assumed that the number of clusters k is known or believed to be known, as this is necessary for the algorithms.
- The number of possible groupings with N objects in k groups is a second-species Stirling number:

$$\left\{ N \atop k \right\} = \frac{1}{k!} \sum_{j=0}^{k} (-1)^{k-j} \binom{k}{j} j^N.$$

- It implies that the cardinality of the search space is NP-complex.

Partitional clustering



2.2. K-means

K-means

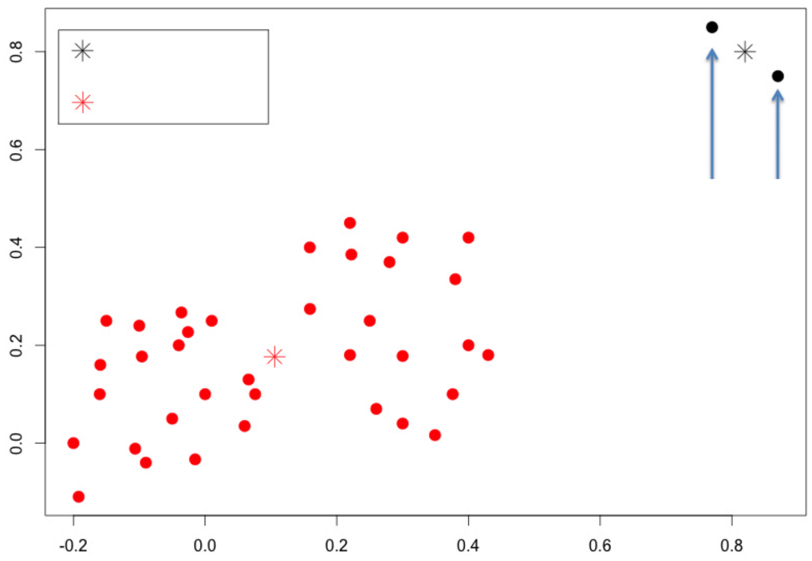
- It requires the number of clusters k as a parameter.
- The clusters are modelled using a set of k hyperspheres.
- Each cluster C_i has an associated centroid $\mathbf{m_i}$.
- The objective is to generate a set of prototypes that minimises a given distortion or error measure.
- The algorithm converges to a (primarily local) optimum.

Algorithm

1. Select the initial k seeds, i.e. the initial centroids: existing patterns or points within the pattern space can be used.
2. Assign each pattern to the nearest cluster (centroid): using one of the aforementioned distances.
3. Update the centroids: the k-means algorithm calculates the centroid as the midpoint of all the patterns in the cluster.
4. Repeat the process from step 2 until the centroids do not change or the pattern assignment is the same, indicating that the algorithm has converged.

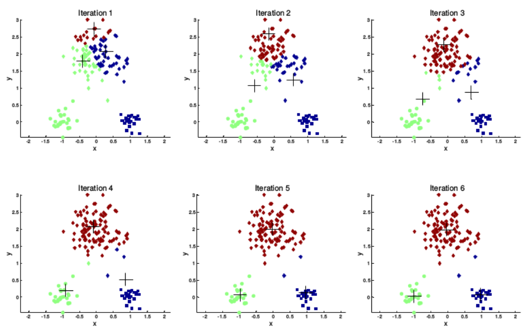
Pros and cons

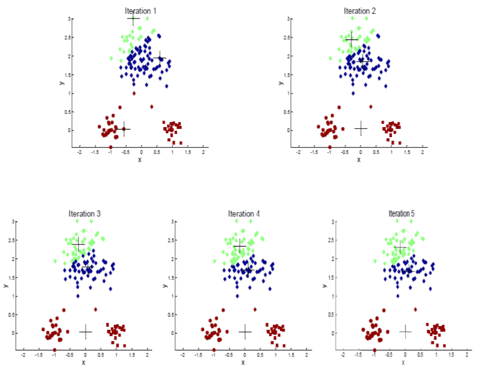
- It is efficient.
- We need to know the number of groups k.
- Unable to detect noise or outliers.



Pros and cons

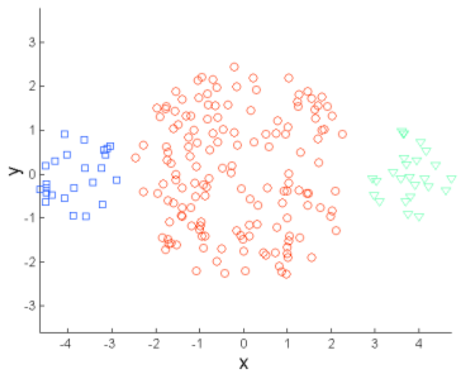
- The final result depends on the initialisation, and it is also very sensitive to it.
- Ends in a local optimum.

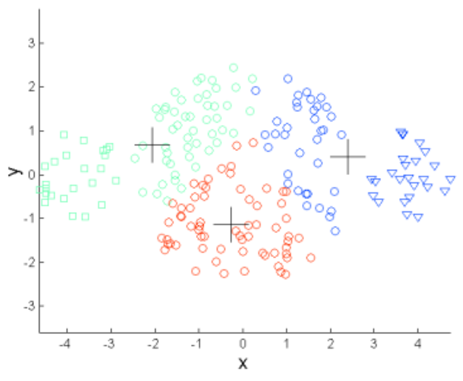




Pros and cons

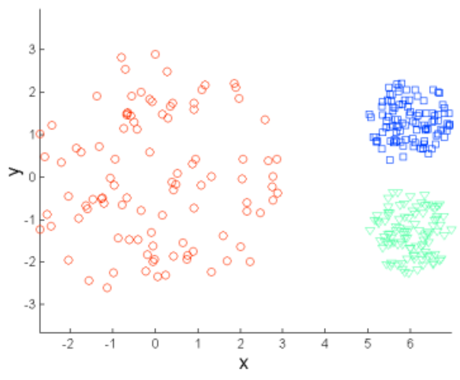
- Unable to detect clusters of different sizes.

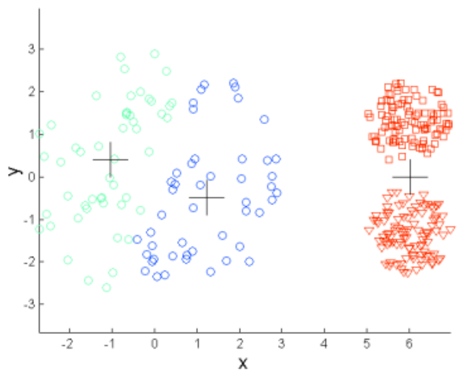




Pros and cons

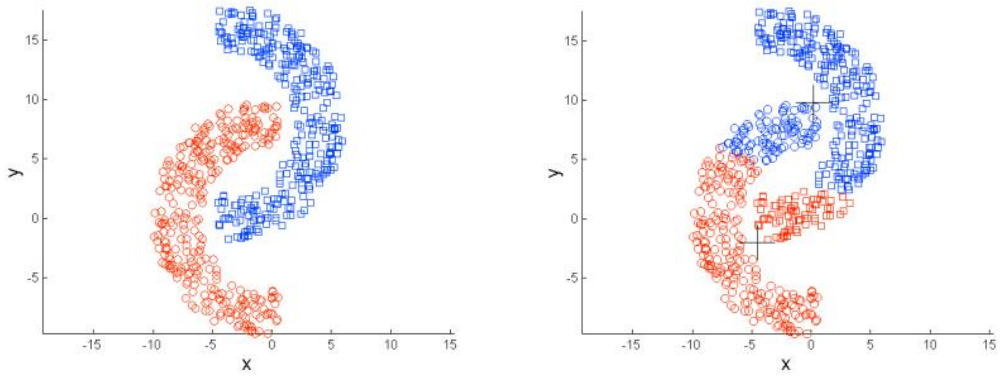
- Unable to detect clusters of different densities.





Pros and cons

- Unable to detect non-convex clusters.



2.3. K-means Variations

Variations

The different methods vary in the way they perform:

- Initialisation: selection of initial centroids:
  - The k first patterns.
  - Random.
  - Chosen by the researcher.
  - Chosen by a heuristic that ensures they are as far apart as possible.
- Assignment of each pattern to its cluster:
  - The centroids are not updated until all reallocations are completed.
  - The centroids (where it came from and where it goes to now) are recalculated each time an object is reallocated.
  - Using batches.

K-means++

- Modify the initialisation strategy to maximise the distance between the different centroids.
- Algorithm:
  1. Choose one center uniformly at random among the data points.
  2. For each data point $\mathbf{x}$ not chosen yet, compute $d(\mathbf{x}, \mathbf{m})$, the distance between $\mathbf{x}$ and the nearest center that has already been chosen.
  3. Choose one new data point at random as a new center, using a weighted probability distribution where a point $\mathbf{x}$ is chosen with probability proportional to $d(\mathbf{x}, \mathbf{m})^2$.
  4. Repeat Steps 2 and 3 until k centers have been chosen.
  5. Apply k-means.

K-medians

- Instead of using the centroid as the centre, use the median.
- To do this, the median is calculated for each attribute, and we return a point formed by all the medians.
- This point can be virtual.
- For instance:

  $$(0,1),(1,0),(2,2) \rightarrow (median\{0,1,2\}, median\{1,0,2\}) = (1,1).$$
- The Manhattan distance ($L1$-norm) is minimised instead of the Euclidean distance ($L2$-norm).

K-medoids

- Instead of using the centroid as the centre, use the medoid.
- A medoid is an object in a group (it is not virtual, it exists) whose average distance to all objects in the group is minimal.
- It is the most centrally located point in the whole group.
- Most complex to calculate!

2.4. K-means Example

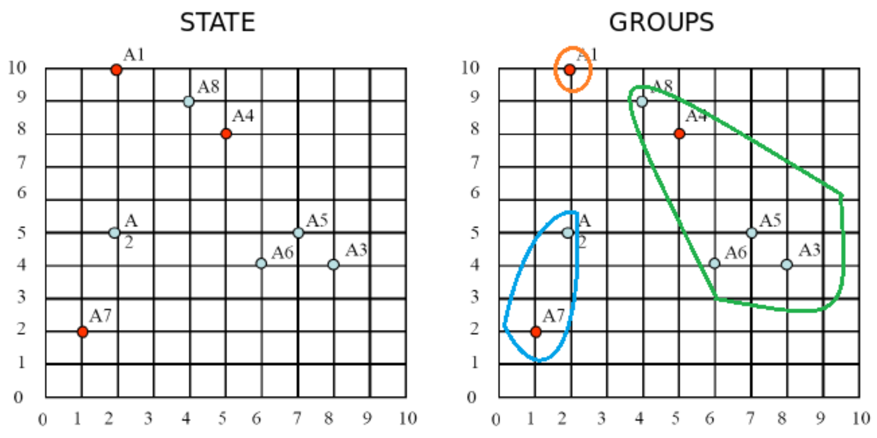
K-means example

- A total of 8 two-dimensional patterns are asked to be grouped into three clusters ($k = 3$). The patterns are as follows: $A1(2,10)$, $A2(2,5)$, $A3(8,4)$, $A4(5,8)$, $A5(7,5)$, $A6(6,4)$, $A7(1,2)$ and $A8(4,9)$.
- The initial centroids are points: $A1$, $A4$ and $A7$.
- The distance metric used will be the Euclidean distance.
- You are requested to:
  - Plot the clusters created and the position of the centroids after each iteration.
  - The value of the SSE metric.

K-means example

Initial state

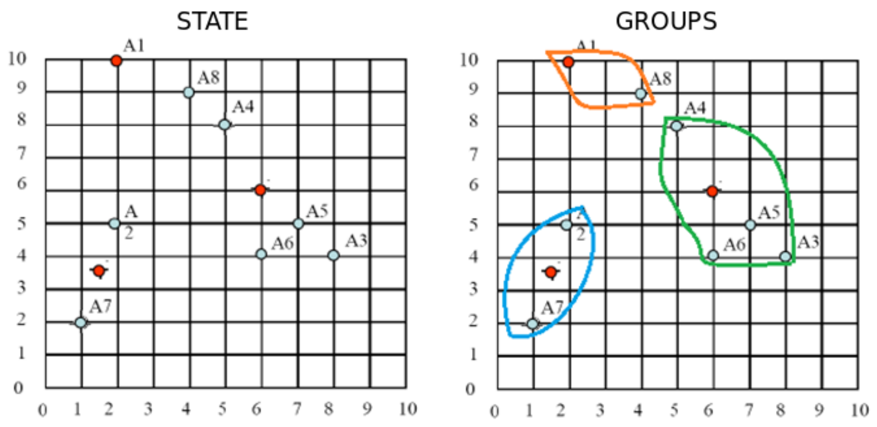
- $C_1 = (2,10)$.
- $C_2 = (1,2)$.
- $C_3 = (5,8)$.



K-means example

Iteration 1

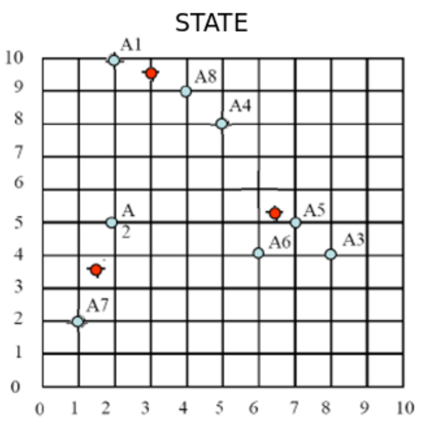
- $C_1 = (2, 10)$.
- $C_2 = ((1 + 2)/2, (2 + 5)/2) = (1.5, 3.5)$.
- $C_3 = ((4 + 5 + 6 + 7 + 8)/5, (9 + 8 + 4 + 5 + 4)/5) = (6, 6)$.

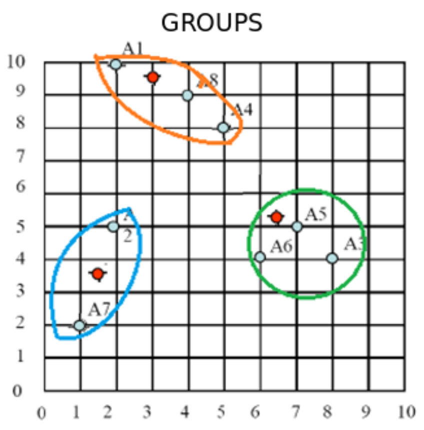


K-means example

Iteration 2

- $C1 = ((2 + 4)/2, (10 + 9)/2) = (3, 9.5)$.
- $C_2 = ((1 + 2)/2, (2 + 5)/2) = (1.5, 3.5)$.
- $C_3 = ((5 + 6 + 7 + 8)/4, (8 + 4 + 5 + 4)/4) = (6.5, 5.25)$.

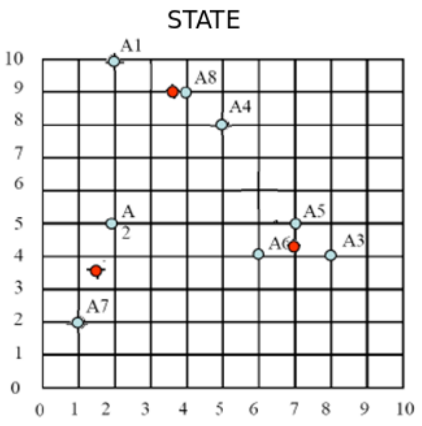


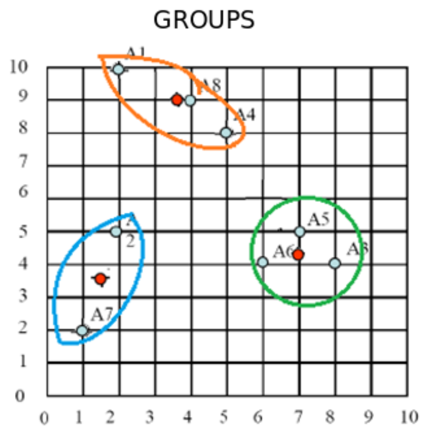


K-means example

Iteration 3

- $C1 = ((2+4+5)/3, (10+9+8)/3) = (3.66, 9)$.
- $C_2 = (1.5, 3.5)$.
- $C_3 = ((6+7+8)/3, (4+5+4)/3) = (7, 4.33)$.





K-means example

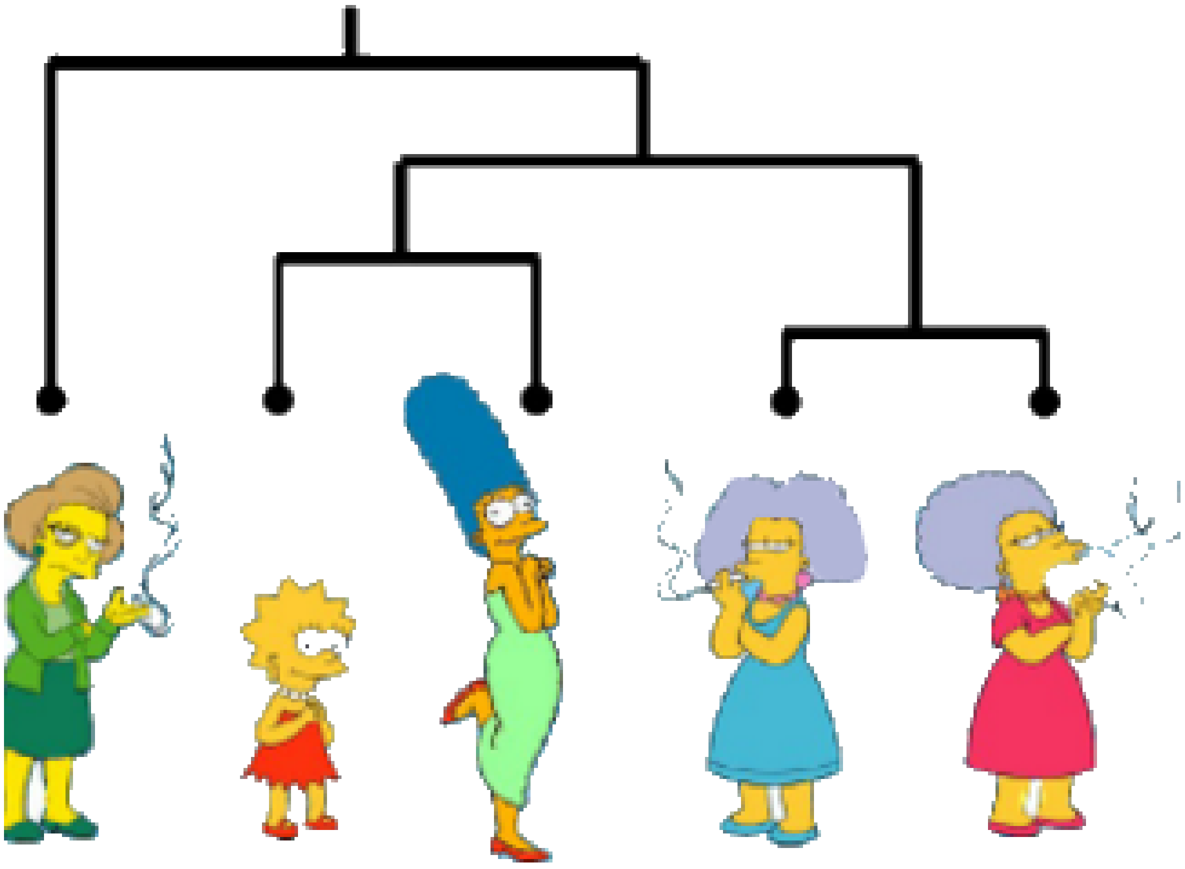
SSE

Point	Cluster	$d(\mathbf{x}, \mathbf{m})$
$A1(2, 10)$	$C1(3.66, 9)$	1.938
$A2(2, 5)$	$C2(1.5, 3.5)$	1.581
$A3(8, 4)$	$C3(7, 4.33)$	1.053
$A4(5, 8)$	$C1(3.66, 9)$	1.672
$A5(7, 5)$	$C3(7, 4.33)$	0.670
$A6(6, 4)$	$C3(7, 4.33)$	1.053
$A7(1, 2)$	$C2(1.5, 3.5)$	1.581
$A8(4, 9)$	$C1(3.66, 9)$	0.340
	SSE	14.333

3. Hierarchical Clustering

3.1. Initial Concepts

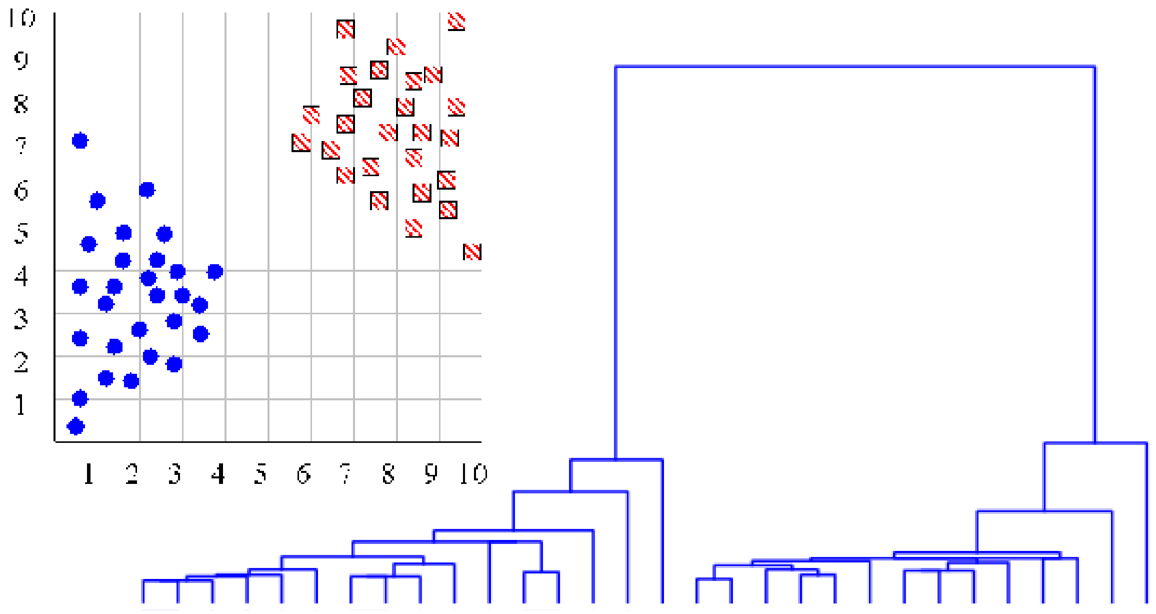
Dendogram



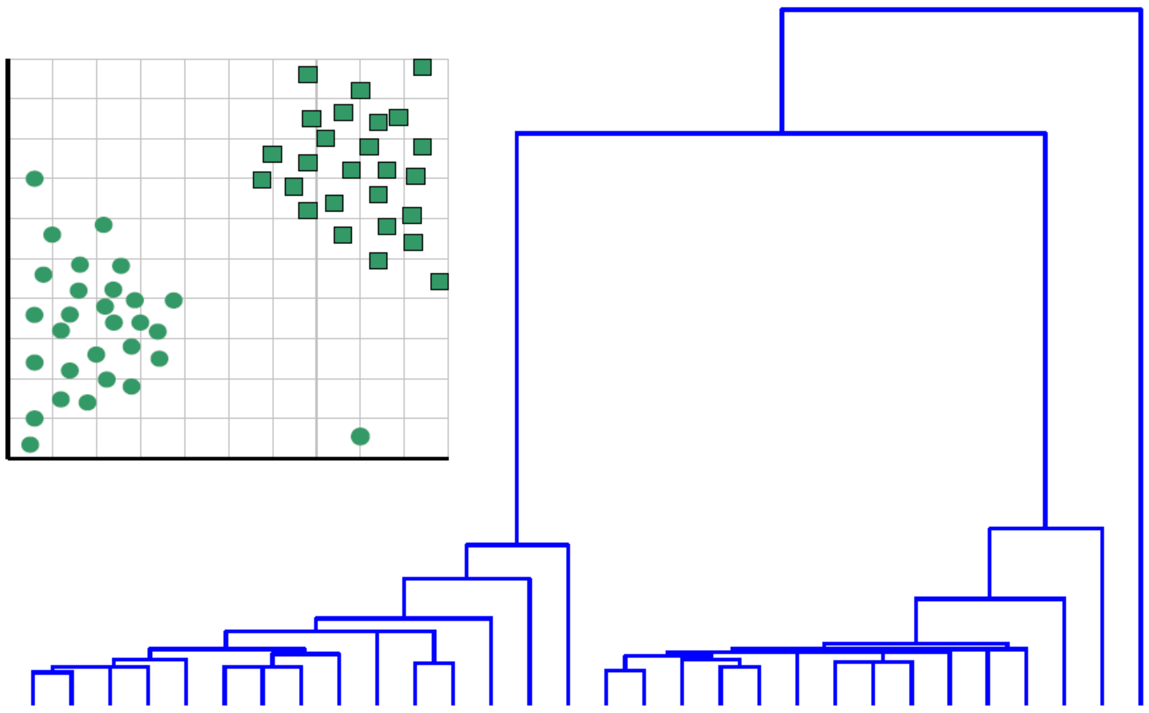
- Hierarchical clustering tries to group objects into hierarchies.
- The dendogram allows us to represent the similarity between objects and how the clusters are linked by levels.
- The height indicates the distance between the objects/ clusters joined in each iteration.

Dendogram

- It allows us to decide the value of k.



- It allows us to detect outliers.



As can be seen, the output is a hierarchy between clusters, where the cut-off level will lead to different clustering. Thus, it is not necessary to fix the value of k.

Types

There are two types of hierarchical clustering:

- Agglomerative: the starting situation is usually one cluster per pattern. In each step, the two closest clusters are merged. The process is repeated until a single cluster is achieved.
- Divisive: this is the opposite process, starting with a single cluster that is divided at each step until there is one cluster per pattern. This process requires a more significant number of calculations, so, as a general rule, the agglomerative method is applied.

Agglomerative algorithm

1. Initially, each node represents one example.
2. Repeat $N-1$ times (N being the number of examples):
   - The two closest examples/clusters are grouped.
   - We delete the rows corresponding to those two examples/groups.
   - Add a new row and column corresponding to the new distances to the new group.
3. Not only groups or classes are generated, but also a hierarchy between them (trees called dendograms).

Example

Group 5 districts according to their homogeneity in household equipment levels.

Distric	Telephone	Washer	Dishwasher
1	0.9	0.8	0.7
2	0.6	0.5	0.3
3	0.7	0.7	0.4
4	0.8	0.7	0.7
5	0.7	0.5	0.3

Example

1. The five clusters are initially the five patterns.
2. Calculate the distances matrix using Euclidean distance:
   $d(\mathbf{x_1}, \mathbf{x_2}) = \sqrt{(0.9-0.6)^2 + (0.8-0.5)^2 + (0.7-0.3)^2} = 0.58.$

	1	2	3	4	5
1	0	0.58	0.37	0.14	0.54
2		0	0.24	0.49	0.10
3			0	0.32	0.22
4				0	0.46
5					0

3. Repeat:
   - Grouped two closest examples: 2, 5.
   - New distances matrix.
     How to calculate the distance between each point and the cluster (2,5)?

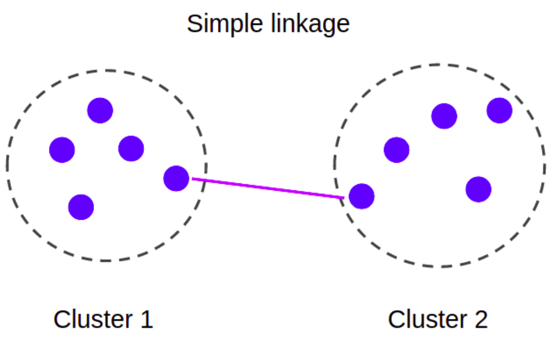
3.2. Distance between Clusters

Distance between clusters

Different methods to calculate the distance between clusters:

- Single linkage.
- Complete linkage.
- Average linkage.
- Centroid linkage.
- Ward linkage.

Single linkage



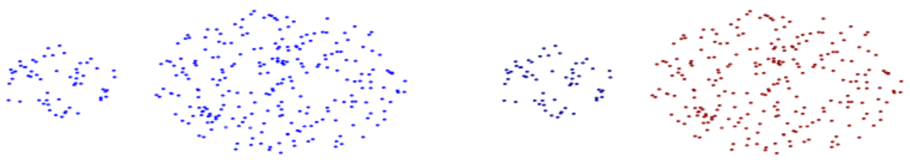
The distance between 2 clusters is the minimum distance between all possible pairs of objects in both clusters:

$$d(C_1, C_2) = min_{x \in C_1, x' \in C_2} d(\mathbf{x}, \mathbf{x}').$$

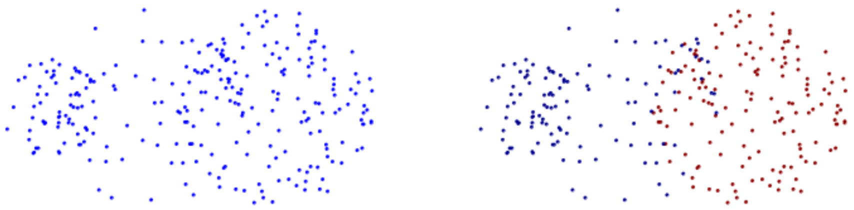
Example: $d(1,(2,5)) = min(d(1,2), d(1,5)) = min(0.58, 0.54) = 0.54$.

Single linkage: pros and cons

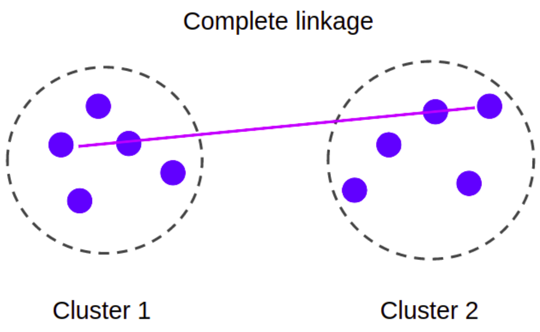
- Pros: It accepts non-circular shapes.



- Cons: Sensitive to noise and outliers.



Complete linkage



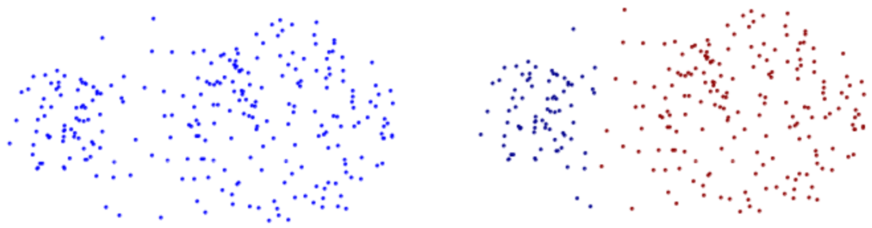
The distance between 2 clusters is the maximum distance between all possible pairs of objects in both clusters:

$$d(C_1, C_2) = max_{x \in C_1, x' \in C_2} d(\mathbf{x}, \mathbf{x}').$$

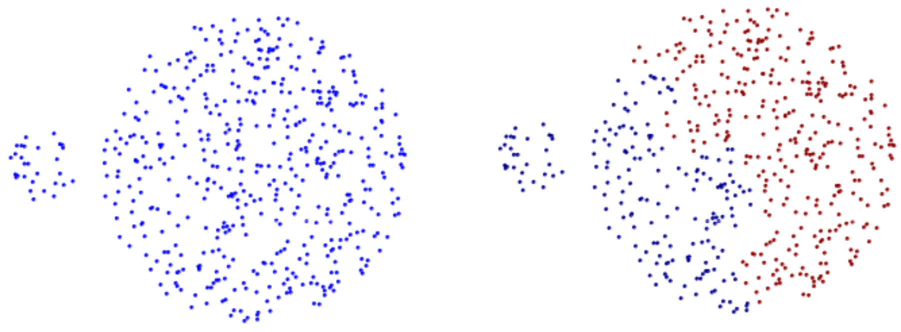
Example: $d(1,(2,5)) = max(d(1,2), d(1,5)) = max(0.58, 0.54) = 0.58$.

Complete linkage: pros and cons

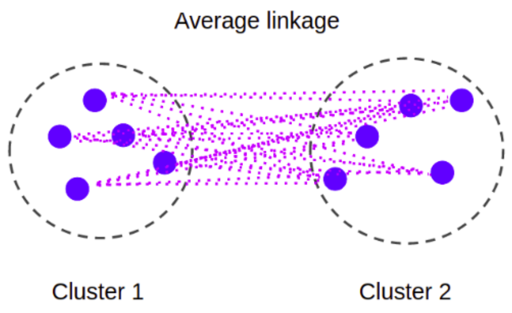
- Pros: Less sensitive to noise and outliers.



- Cons: Tends to “break up” large clusters.



Average linkage

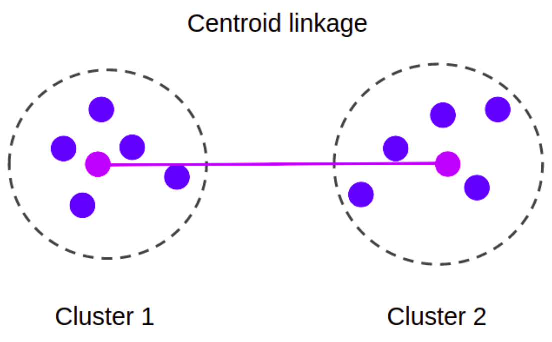


The distance between 2 clusters is the mean distance between all possible pairs of objects in both clusters:

$$d(C_1, C_2) = \frac{1}{|C_1||C_2|} \sum_{x \in C_1, x' \in C_2} d(\mathbf{x}, \mathbf{x}').$$

Example: $d(1,(2,5)) = \frac{1}{2}(d(1,2) + d(1,5)) = 0.56$.
Pros: Less sensitive to noise and outliers.
Cons: Tends to “break up” large clusters.

Centroid linkage

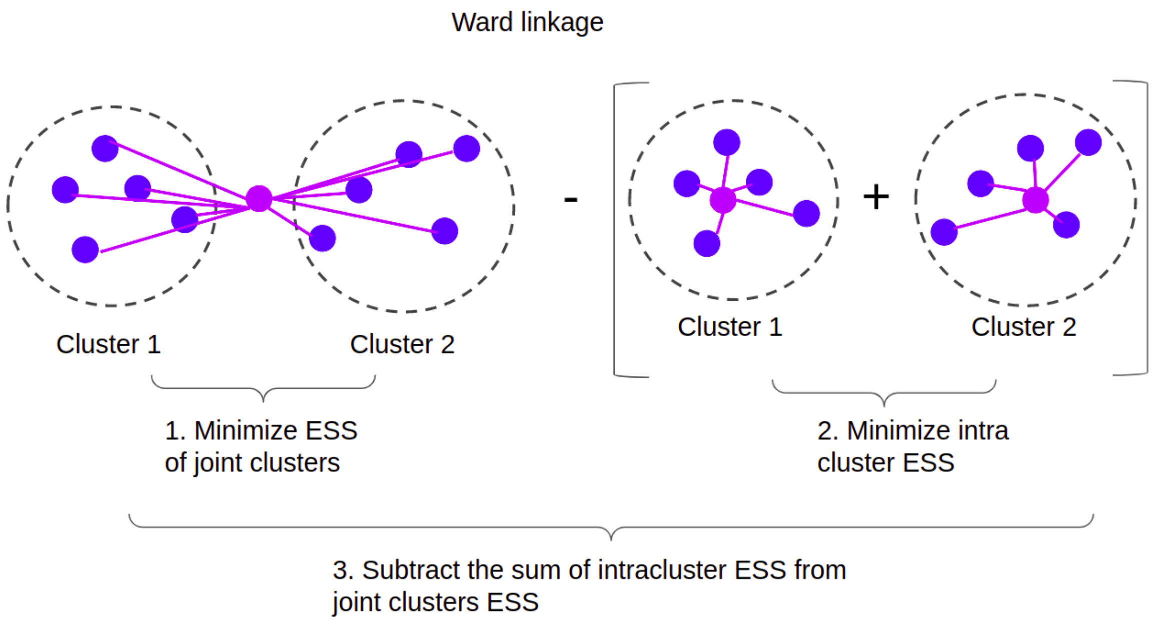


The distance between 2 clusters is the distance between the centroids of both clusters:

$$d(C_1, C_2) = d(\mathbf{m}_1, \mathbf{m}_2).$$

Example: $d(1,(2,5)) = d(1, centroid(2,5)) = d((0.9, 0.8, 0.7), (0.65, 0.5, 0.3)) = 0.56$.

Ward linkage

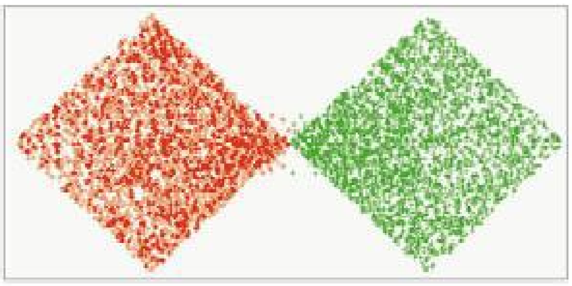
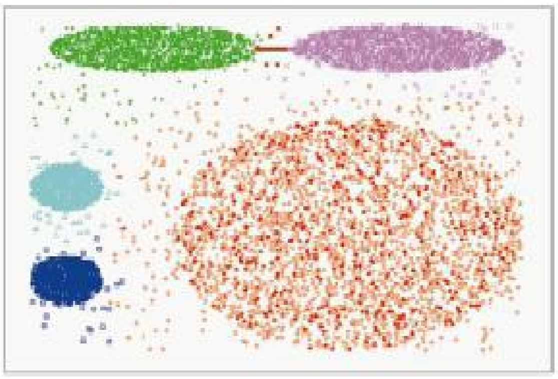
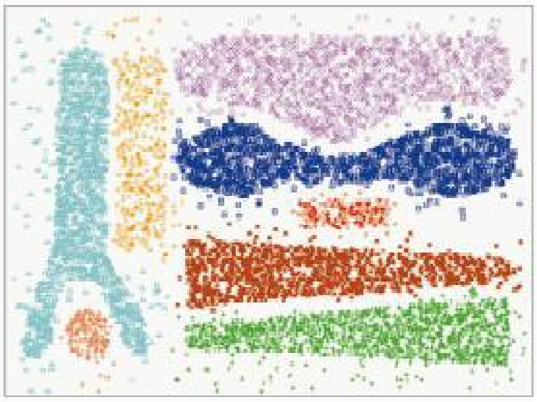
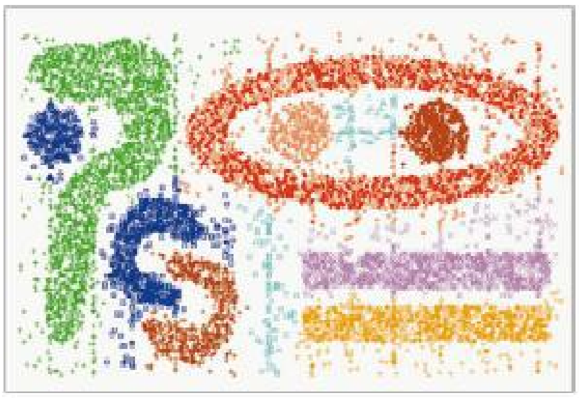

Ward linkage
Cluster 1
Cluster 2
-
+
Cluster 1
Cluster 2
1. Minimize ESS of joint clusters
2. Minimize intra cluster ESS
3. Subtract the sum of intracluster ESS from joint clusters ESS

3.3. Pros and Cons

Pros and cons

Although the advantages and disadvantages depend on the type of link used in these algorithms, in general, the following conclusions can be drawn:

- Low scalability.
- No need to set the value of k.
- Allows detection of outliers.
- Does not depend on initialisation.
- Allows detection of non-convex clusters.

3.4. Hierarchical Clustering Example

Hierarchical clustering example

- Given the following matrix of distances between four patterns:

	A	B	C	D
A	0	1	4	5
B		0	2	6
C			0	3
D				0

- You are requested to:
  - Apply hierarchical clustering with the single linkage method.
  - Apply hierarchical clustering with the complete linkage method.

Hierarchical clustering example

Single linkage

1. Iteration 1:
   - We group C_A with C_B: $d(A, B) = 1$.

  - Resulting clusters: C_{AB}, C_C, C_D.

2. Iteration 2:
   - $d(C_{AB}, C_C) = min(d(A,C), d(B,C)) = min(4,2) = 2$.
   - $d(C_{AB}, C_D) = min(d(A,D), d(B,D)) = min(5,6) = 5$.
   - $d(C_C, C_D) = 3$.
   - We group C_{AB} and C_C.
   - Resulting clusters: C_{ABC}, C_D.
3. Iteration 3:
   - We join the two remaining clusters $C_{ABC}, C_D \to C_{ABCD}$.

Hierarchical clustering example
Complete linkage

1. Iteration 1:
   - We group C_A with C_B: $d(A,B) = 1$.
   - Resulting clusters: C_{AB}, C_C, C_D.
2. Iteration 2:
   - $d(C_{AB}, C_C) = max(d(A,C), d(B,C)) = max(4,2) = 4$.
   - $d(C_{AB}, C_D) = max(d(A,D), d(B,D)) = max(5,6) = 6$.
   - $d(C_C, C_D) = 3$.
   - We group C_C and C_D.
   - Resulting clusters: C_{AB}, C_{CD}.
3. Iteration 3:
   - We join the two remaining clusters $C_{AB}, C_{CD} \to C_{ABCD}$.

4. Density-Based Clustering

4.1. Introduction

Introduction

- Density-based clustering is based on the principle that dense clustering regions of points are separated from other dense regions by sparse regions.
- Unlike density-based clustering, the above algorithms cannot perform well when applied to tasks with arbitrarily shaped clusters or clusters within clusters.
- It can detect outliers, i.e. regions of low density whose points will not be included in any cluster.

4.2. DBSCAN

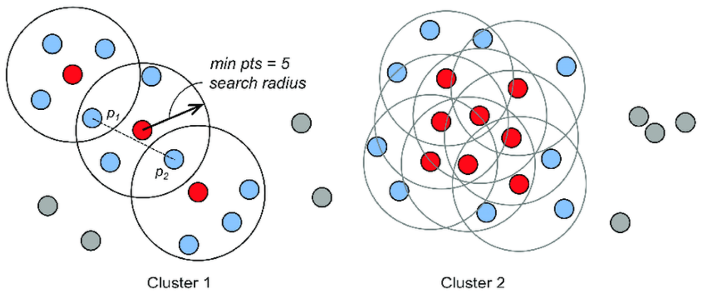
DBSCAN

- DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is the most well-known density-based clustering algorithm.
- It works on the idea that if a point belongs to a cluster, it should be surrounded by a lot of other points in that cluster.
- DBSCAN has two parameters:
  - ε: the radius of our neighborhoods around a data point $\mathbf{x}$.
  - $minPts$: the minimum number of data points we want in a neighborhood to define a cluster.
- The ε-neighbourhood of a point is defined as:

$$N_\varepsilon(\mathbf{x}) = \{\mathbf{y} \in X | d(\mathbf{x}, \mathbf{y}) < \varepsilon\}.$$

Type of points

- Using the previous parameters and the definition of $N_\varepsilon(\mathbf{x})$, DBSCAN categories the data points into three categories:
  - **Core Points**: A data point p is a core point if $N_\varepsilon(p)$ contains at least $minPts$: $N_\varepsilon(p) \geq minPoints$.
  - **Border Points**: A data point q is a border point if $N_\varepsilon(q)$ contains less than $minPts$ data points, but q is reachable from some core point p.
  - **Outlier**: A data point o is an outlier if it is neither a core point nor a border point.



Algorithm

```
DBSCAN(D,eps,MinPts)
  C=0
  for each unvisited point P in dataset D
    mark P as visited
    NeighborPts = regionQuery(P, eps)
    if sizeof(NeighborPts) < MinPts
      mark P as NOISE
    else
      C = next cluster
      expandCluster(P, NeighborPts, C, eps, MinPts)

expandCluster(P, NeighborPts, C, eps, MinPts)
  add P to cluster C
  for each point P' in NeighborPts
    if P' is not visited
      mark P' as visited
      NeighborPts' = regionQuery(P', eps)
      if sizeof(NeighborPts') >= MinPts
        NeighborPts = NeighbortPts joined with NeighborPts'
      if P' es not yet member of any cluster
        add P' to cluster C

regionQuery(P, eps)
  return all points within P's eps−neighborhood (including P)
```

4.3. Pros and Cons

Pros and cons

- Identify clusters arbitrarily.
- They are robust to the presence of noise.
- The k value does not need to be fixed, but ε and $minPts$ do.
- It is scalable, as it only needs a single run over the dataset.
- A problem can be that the border points belong to more than one cluster; therefore, it is necessary to decide to which group they belong, resulting in a non-deterministic algorithm.
- If there are large differences in densities, DBSCAN may not work well, as choosing a set of parameters that work well for all clusters is challenging.

4.4. DBSCAN Example

DBSCAN example

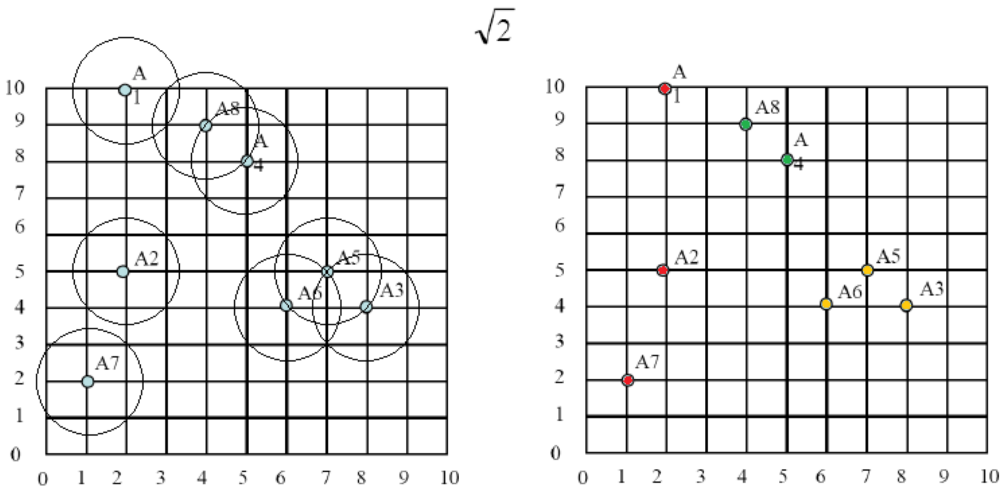
- Apply DBSCAN using the points of the k-means Example:

  a) Considering $minPts = 2$ and $\varepsilon = \sqrt{2}$.

  b) Considering $minPts = 2$ and $\varepsilon = \sqrt{10}$.

DBSCAN example

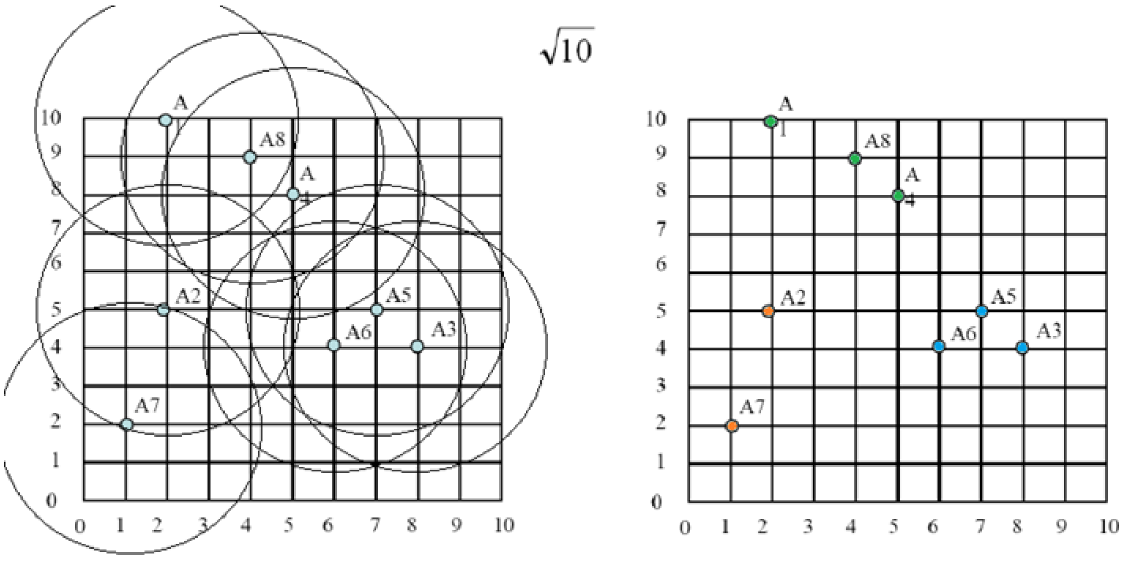
Considering $minPts = 2$ and $\varepsilon = \sqrt{2}$.



Outliers
Cluster 1
Cluster 2

DBSCAN example

Considering $minPts = 2$ and $\varepsilon = \sqrt{10}$.



Cluster 1
Cluster 2
Cluster 3

References

- Benítez-Iglésias, R., Escudero-Bakx, G., Kanaan-Izquierdo, S., Masip-Rodó, D. (2014). Inteligencia artificial avanzada. Editorial UOC.
- Bishop, CM. (2006). Pattern recognition and machine learning. Springer.
- Géron, A. (2019). Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow (Second Edition). O'Reilly.
- James, G.; Witten, D.; R Hastie, T.; Tibshirani, R.; Taylor, J. (2023). An Introduction to Statistical Learning with Applications in Python. Springer.
- Mohri, M., Talwalkar, A., Rostamizadeh, A. (2019). Foundations of machine learning. Cambridge: The MIT Press.

Questions?