

NOVA

IMS

Information
Management
School

9

MODELING: CLUSTERING

Machine Learning for Marketing

© 2020-2023 Nuno António

Instituto Superior de Estatística e Gestão da Informação
Universidade Nova de Lisboa

Acreditações e Certificações



Summary



1. Introduction

2. Clustering methods

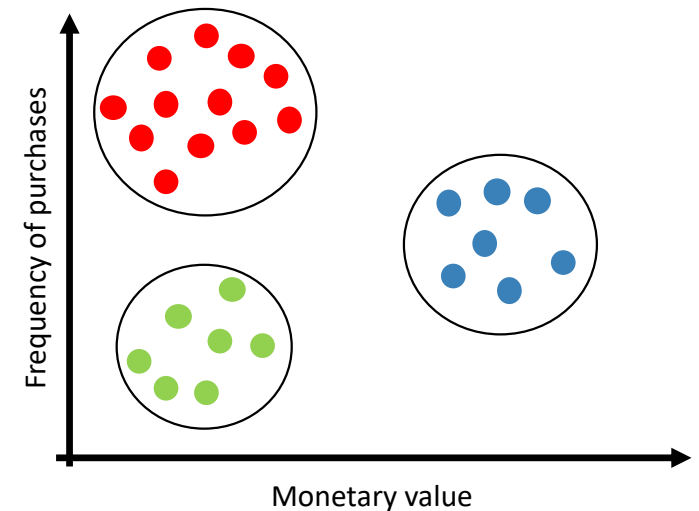
3. Evaluating clustering quality

Introduction

Modeling: Clustering

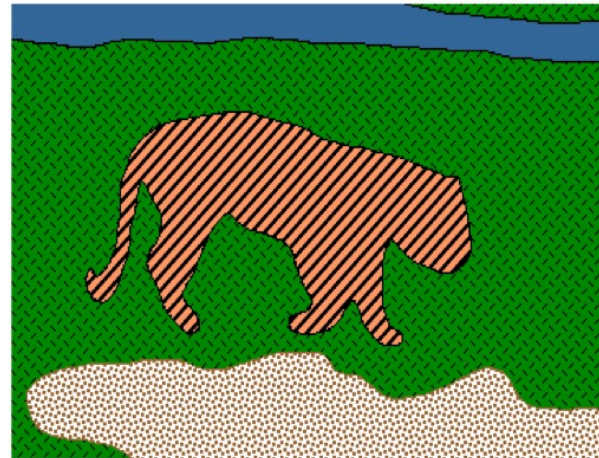
Clustering

- **Cluster analysis** or simply **clustering** is the process of partitioning data objects (instances) into subsets
- Each subset is a **cluster**, such that objects in a cluster are similar to one another, yet dissimilar to objects in other clusters
- Different clustering methods may generate different clusters on the same data
- Data objects are not made by humans, but by the method algorithm. Hence, clustering can lead to the discovery of unknown groups within the data



Typical applications

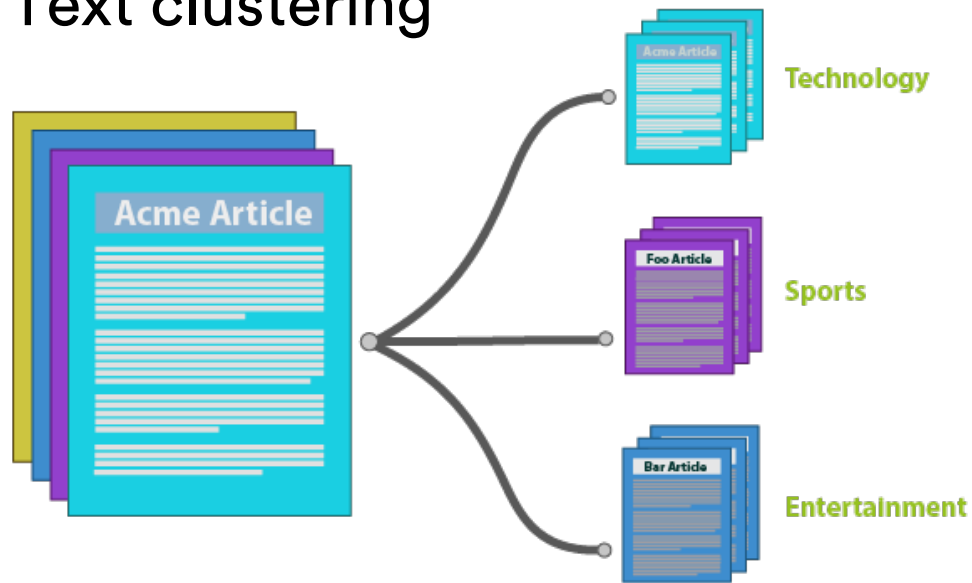
Image segmentation



<https://ai.stanford.edu/~syueung/cvweb/tutorial3.html>

Typical applications

Text clustering



<https://towardsdatascience.com/applying-machine-learning-to-classify-an-unsupervised-text-document-e7bb6265f52>

Typical applications

Customer segmentation



<https://www.baker-richards.com/insights/segmentation-what-how-and-why/>

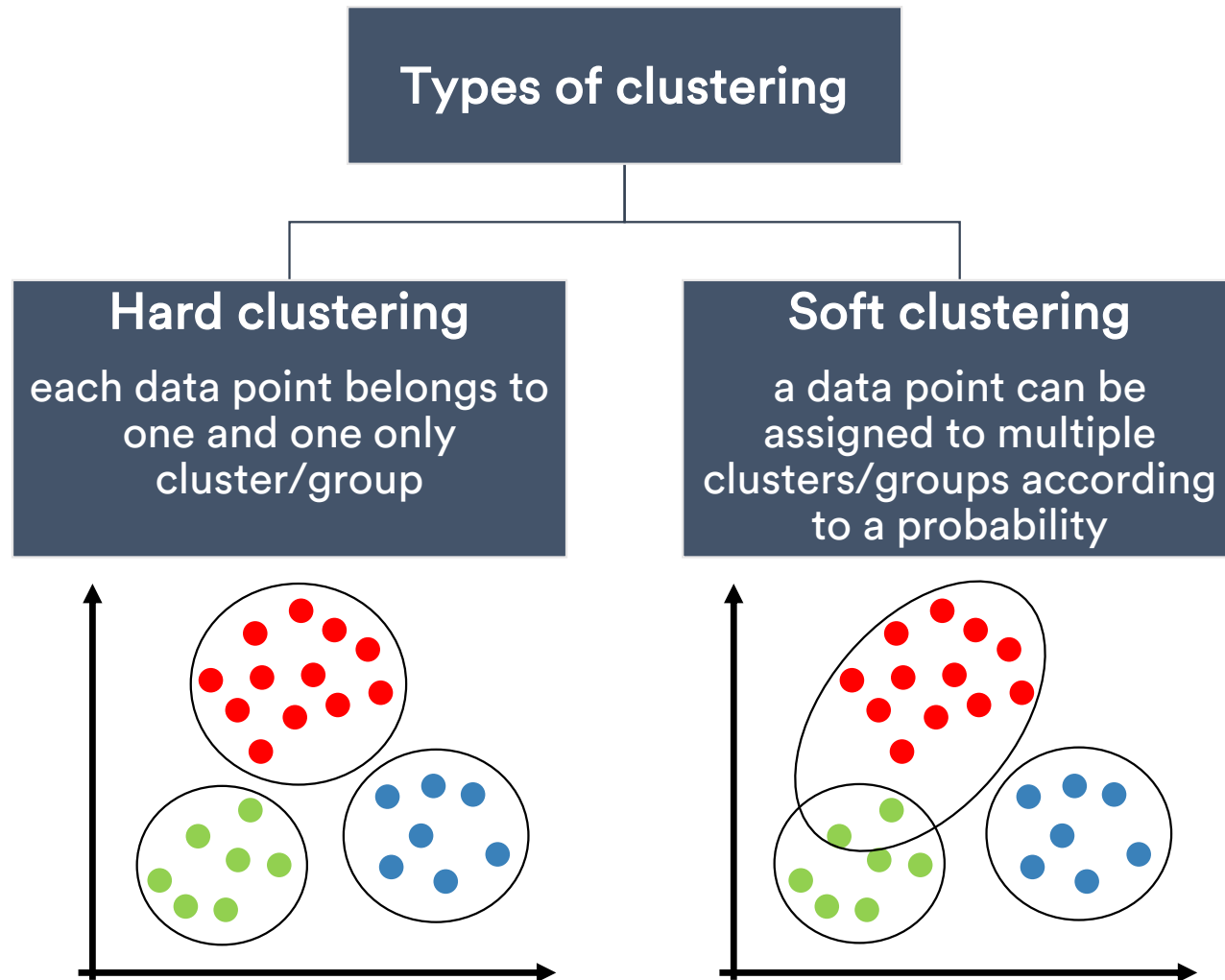
Typical applications

- Language clustering
- Gene clustering
- Product segmentation
- Among many others

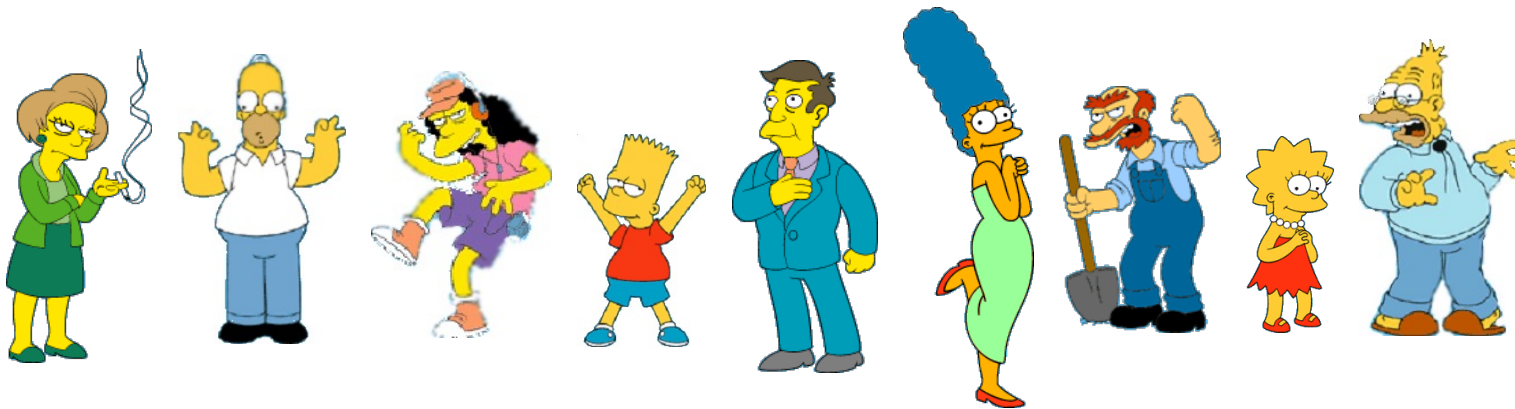
Applications in data analysis

- **Automatic classification:** as there is no label – is a form of *learning by observation*, rather than *learning by examples*
- **Data segmentation:** allows data to be split into partitions according to their similarity
- **Outlier detection:** to identify data objects that are “far away” from any cases. Many times, these are the more interesting patterns (e.g., fraud detection or VIP customers)

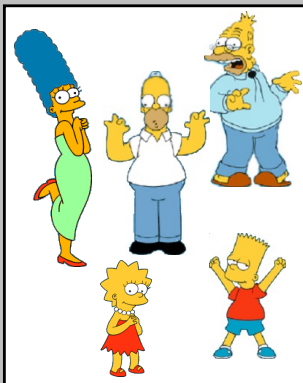
Types of clustering



There is more than one way of grouping objects



Clustering is subjective



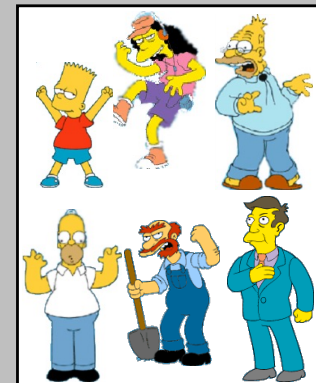
Simpson's Family



School Employees



Females



Males

Requirements for cluster analysis (1/2)

- **Scalability:** ability to scale well to large datasets
- **Ability to deal with different type of attributes:** more and more clustering application is done with non-numeric data, such as images, documents, and other types of data
- **Discovery of clusters arbitrary shape:** algorithms based on distance tend to find spherical clusters with similar size and density. However, a cluster could be of any shape
- **Requirements for domain knowledge to determine input parameters:** some algorithms require users to provide domain knowledge in the form of inputs, such as the number of clusters
- **Ability to deal with noisy data:** clustering methods should be robust to noise, such as outliers and missing data

Requirements for cluster analysis (2/2)

- **Incremental clustering and insensitivity to order:** algorithms should be insensitive to objects order and when possible, should be able to incorporate new data to do incremental updates
- **Capability of clustering high-dimensionality data:** algorithms should be able to work with highly dimensional data, such as documents, who can be very sparse and skewed
- **Constraint-based clustering:** real-world applications may need to perform clustering under constraints (e.g., clusters should be analyzed by city)
- **Interpretability and usability:** clustering should be interpretable, comprehensible, and usable

Clustering methods

Modeling: Clustering

Partitioning methods

Clustering methods

General characteristics

- Find hard clusters of spherical shape
- Based on the notion that similarity between data objects/points is derived by the closeness (distance) of a data object/point to the centroid of the cluster
- May use mean, medoid, or other measure to represent cluster center

STRENGTHS

- Simple and easy to understand
- Effective for small- to medium-size datasets

WEAKNESSES

- Requires domain knowledge to define k (the number of clusters)

Distance between objects/data points

- Eucledian distance

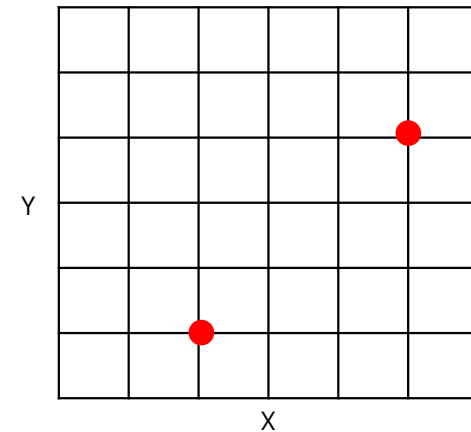
$$\sqrt{\sum_{i=1}^n (x_i - y_i)^2} = \sqrt{(2 - 5)^2 + (1 - 4)^2} = \sqrt{9 + 9} = 4.24$$

- Manhattan distance

- $\sum_{i=1}^n |(x_i - y_i)| = |2 - 5| + |1 - 4| = 6$

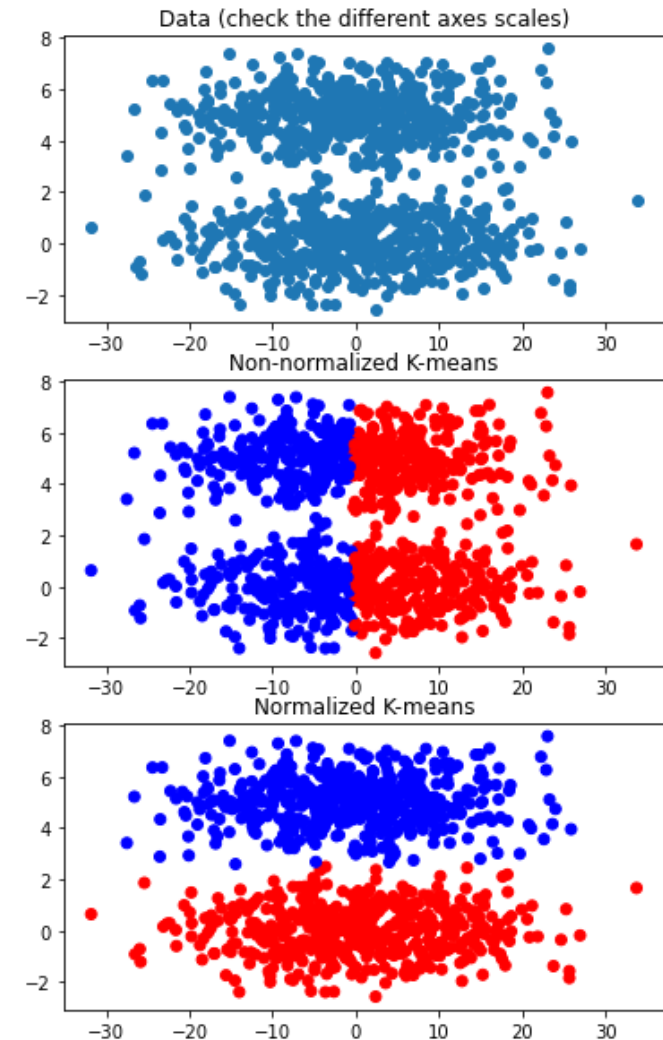
- Hamming distance

- Minkowski distance



Should normalization be done?

If you have variables in different scales (e.g., height in meters and weight in kgs) one will influence more the distance than the other. If that matters for your analysis, you should normalize the data



High dimensional data and clustering

- Difficult to find of true clusters:
 - Irrelevant and redundant features
 - All points are equally close
 - Makes modeling slower
- Solution – Dimension reduction:
 - Feature selection
 - PCA
 - And other dimension reduction techniques

Partitioning clustering algorithms

- K-Means
- K-Medoids
- Fuzzy C-Means
- CLARA
- CLARANS

K-Means algorithm

Introduction

- Hard partitioning algorithm: each data point falls only into one partition
- Iterative: although it converges most of the time, it does not guarantee convergence

STRENGTHS

- Simple to implement
- Scales to large datasets
- Easily adapts to new examples

WEAKNESSES

- Requires domain knowledge to define k (the number of clusters)
- Clusters have different sizes and densities
- Clusters can be dragged to outliers
- Performance is affected in high dimensionality data

K-Means algorithm

Convergence objective

$c^{(i)}$ = cluster index (1, 2, ..., K) to which instance $x^{(i)}$ is assigned to

μ_k = cluster k centroid

$\mu_{c^{(i)}}$ = centroid of cluster to which instance $x^{(i)}$ was assigned to

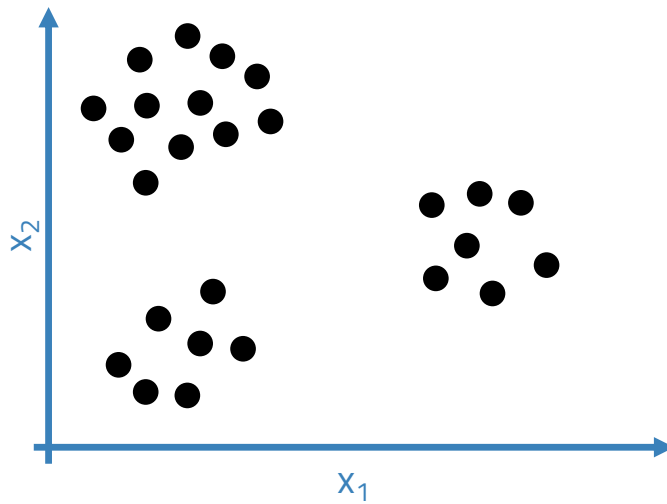
$$J(c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_k) = \frac{1}{m} \sum_{i=1}^m \|x^{(i)} - \mu_{c^{(i)}}\|^2$$
$$\min_{\substack{c^{(1)}, \dots, c^{(m)}, \\ \mu_1, \dots, \mu_k}} J(c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_k)$$

K-Means algorithm

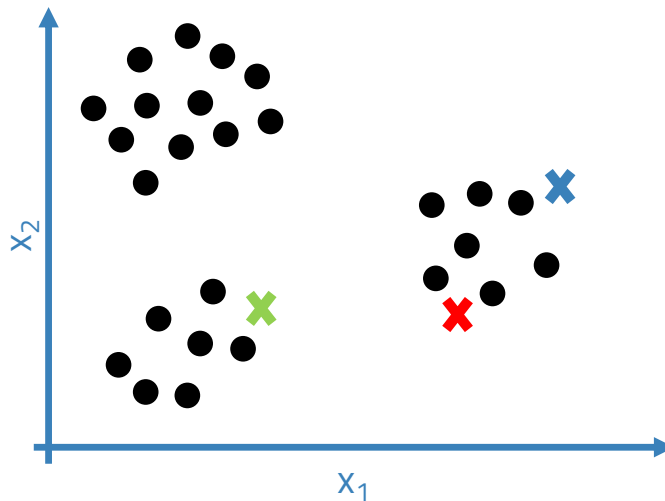
Convergence objective

1. Randomly initialize K centroids of clusters $(\mu_1, \mu_2, \dots, \mu_k)$
2. Repeat {
 - For $i=1$ until m
 - $c^{(i)} := \text{index (from 1 to K) of nearby centroid to } x^{(i)}$
 - For $k=1$ until K
 - $\mu_k := \text{mean of the data points assigned to cluster k}$}

K-Means algorithm Visualization



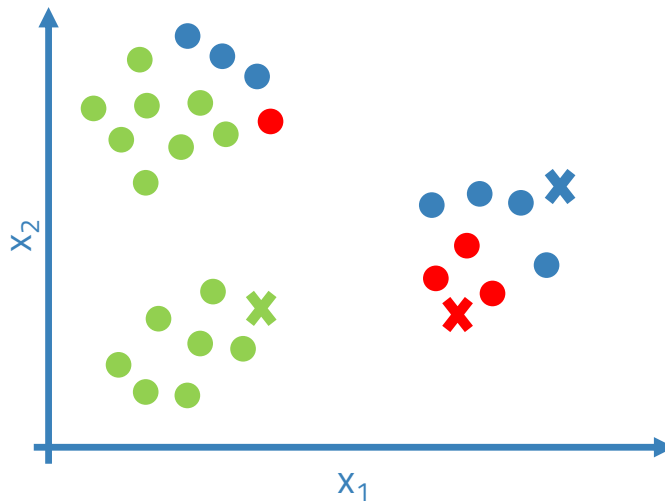
K-Means algorithm Visualization



Step 1:

1. Define the number of clusters (K)
2. Arrange K points randomly - centroids (e.g., $K=3$)

K-Means algorithm Visualization



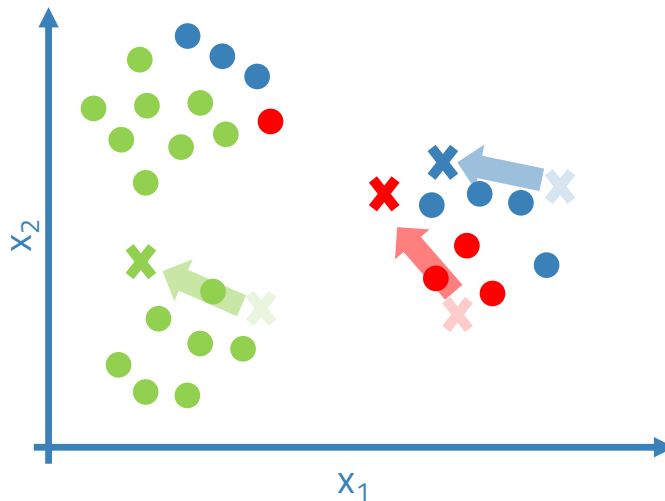
Step 2:

1. For each point, calculate the distance with each centroid (e.g., Euclidean)

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

2. Assign each point to the centroid closest to it

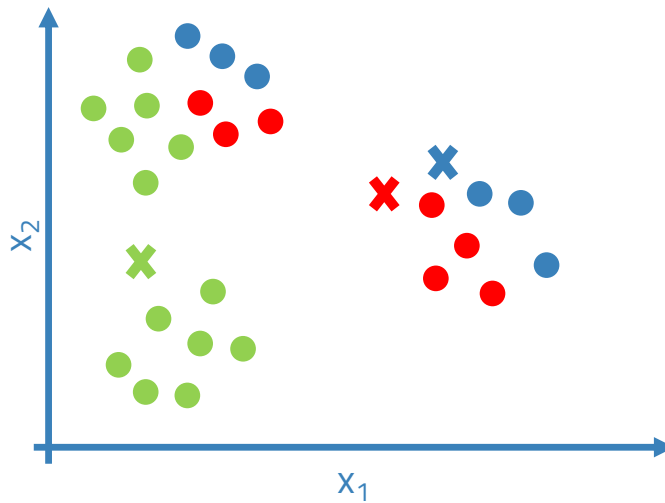
K-Means algorithm Visualization



Step 3:

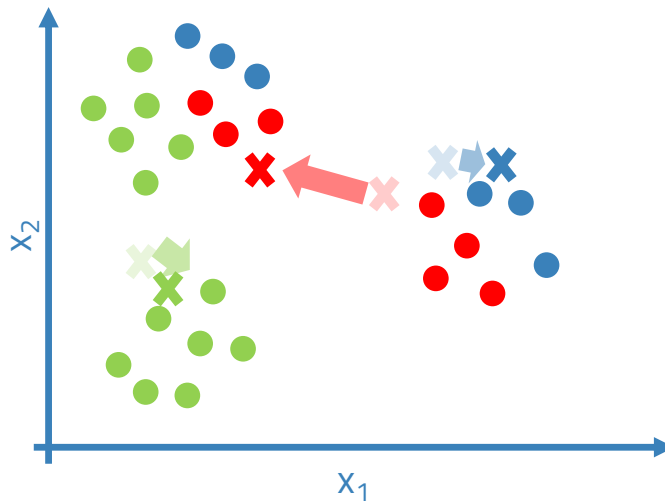
1. Calculate the average of the points assigned to each cluster
2. Change the placement of each centroid to the previous calculated average

K-Means algorithm Visualization



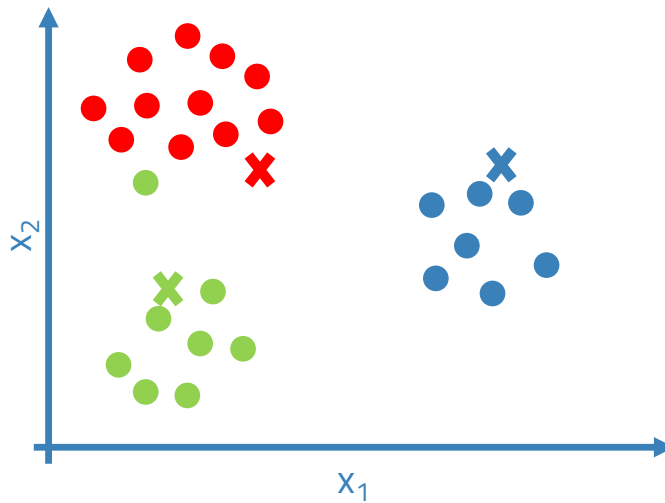
Repeat steps 2 and 3 until
convergence

K-Means algorithm Visualization



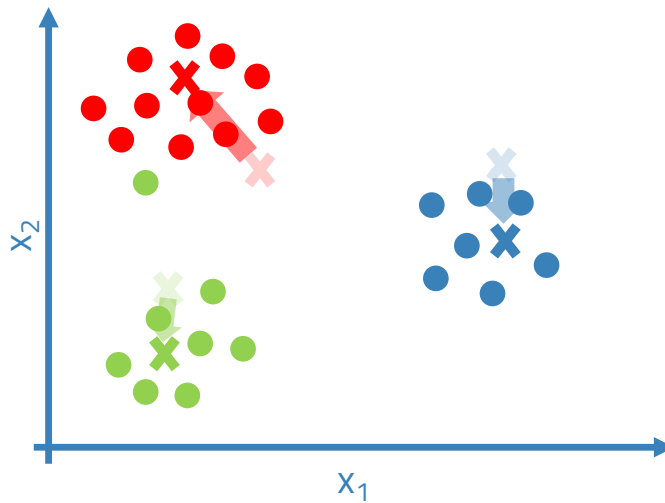
Repeat steps 2 and 3 until
convergence

K-Means algorithm Visualization



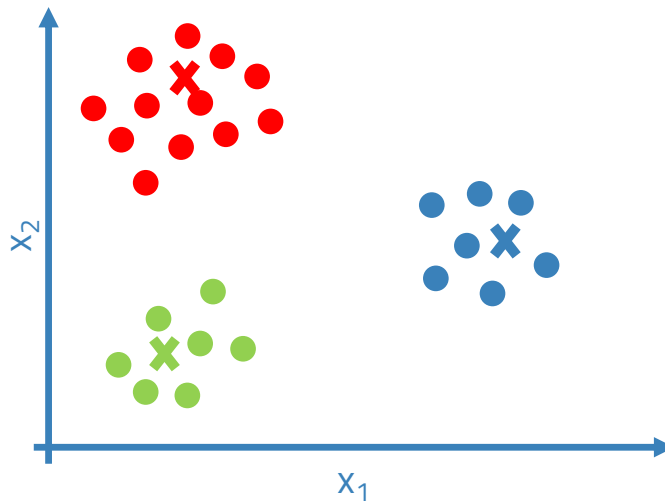
Repeat steps 2 and 3 until
convergence

K-Means algorithm Visualization



Repeat steps 2 and 3 until
convergence

K-Means algorithm Visualization



Repeat steps 2 and 3 until
convergence

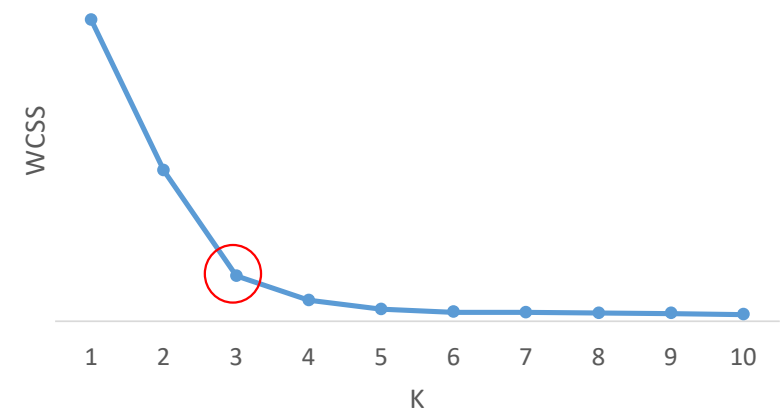
K-Means algorithm

K selection: “Elbow method”

1. Calculate K-Means clustering for a range of K (e.g., 1 to 10)
2. In each K calculate the WCSS (Within-Cluster Sums of Squares), which is the sum of the squares of the distances from each data point (p) to the centroid (c_i) of its cluster (C_i), for all clusters (k)

$$E = \sum_{i=1}^k \sum_{p \in C_i} \text{dist}(p, c_i)^2$$

3. Graph the WCSS value of each K cluster
4. Select the K where there is a curve in the graph, as it indicates that from there the WCSS stops rapidly decreasing



K-Means algorithm

K selection: other techniques

- **Silhouette:** measures how similar a data point is to its own cluster (cohesion) compared to other clusters (separation). Range varies between -1 and +1. The higher the value, the better
 - **Gap statistic:** compares the total within intra-cluster variation for different values of K with their expected values under null reference distribution of the data. The best K is the one that maximizes the gap statistic
 - **Dunn index:** used to identify compact and well separated clusters. The higher the value the better ($Dunn\ Index = \frac{\min. inter\ cluster\ distance}{\max. intra\ cluster\ distance}$)
 - Among others
- ⚠ For outliers or anomaly detection:
- Select a low number of clusters (2 or 3); or
 - Define a K so high that each cluster have a small number of instances (except a pattern of outliers)

Hierarchical methods

Clustering methods

General characteristics

- Create a hierarchical decomposition (i.e., multiple levels – “tree”) of a given dataset
- Cannot correct erroneous merges or splits
- May incorporate other techniques like micro-clustering or consider objects “linkages”
- Hierarchical methods can be distance-based or density- and continuity-based

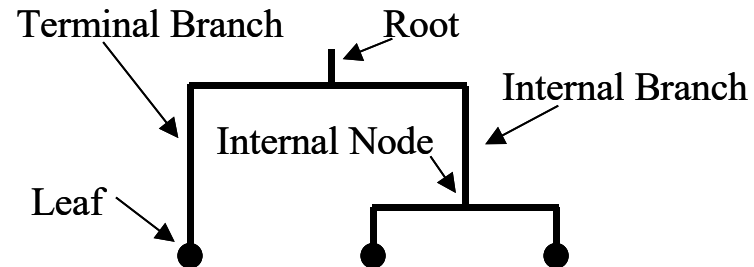
STRENGTHS

- Easy to implement
- No need to specify the number of clusters in advance
- Interpretable
- Often reveal finer details in relationships

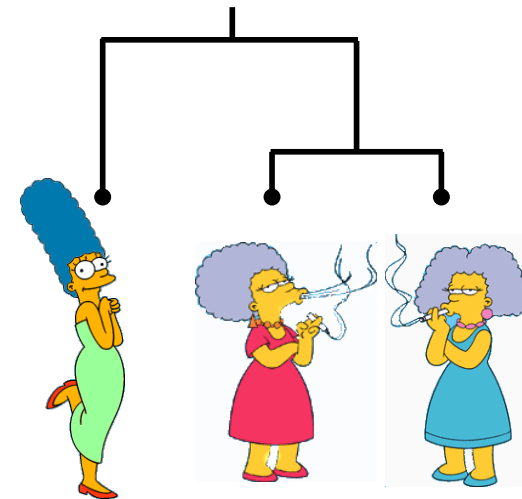
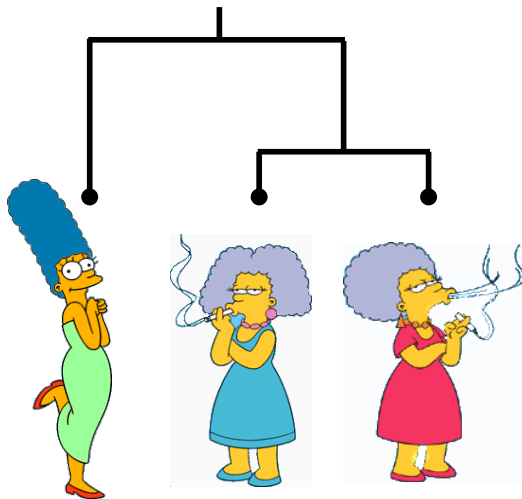
WEAKNESSES

- Tend to be difficult to scale
- Only finds spherical-shaped clusters
- Interpretation of results is (very) subjective
- Computationally expensive

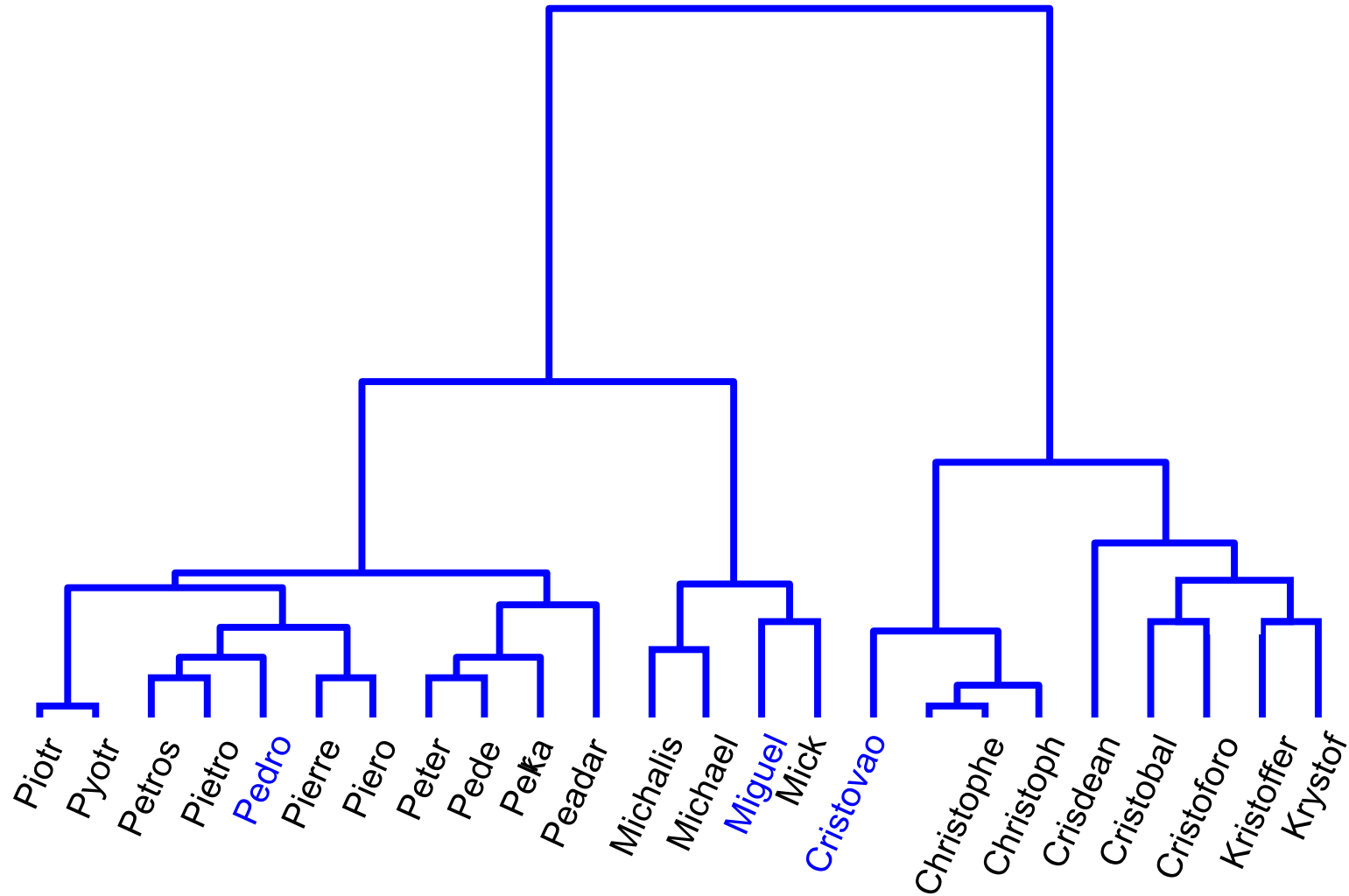
Dendrogram: a useful tool for summarizing similarity measurements



The similarity between two objects in a dendrogram is represented as the **height** of the lowest internal node they share

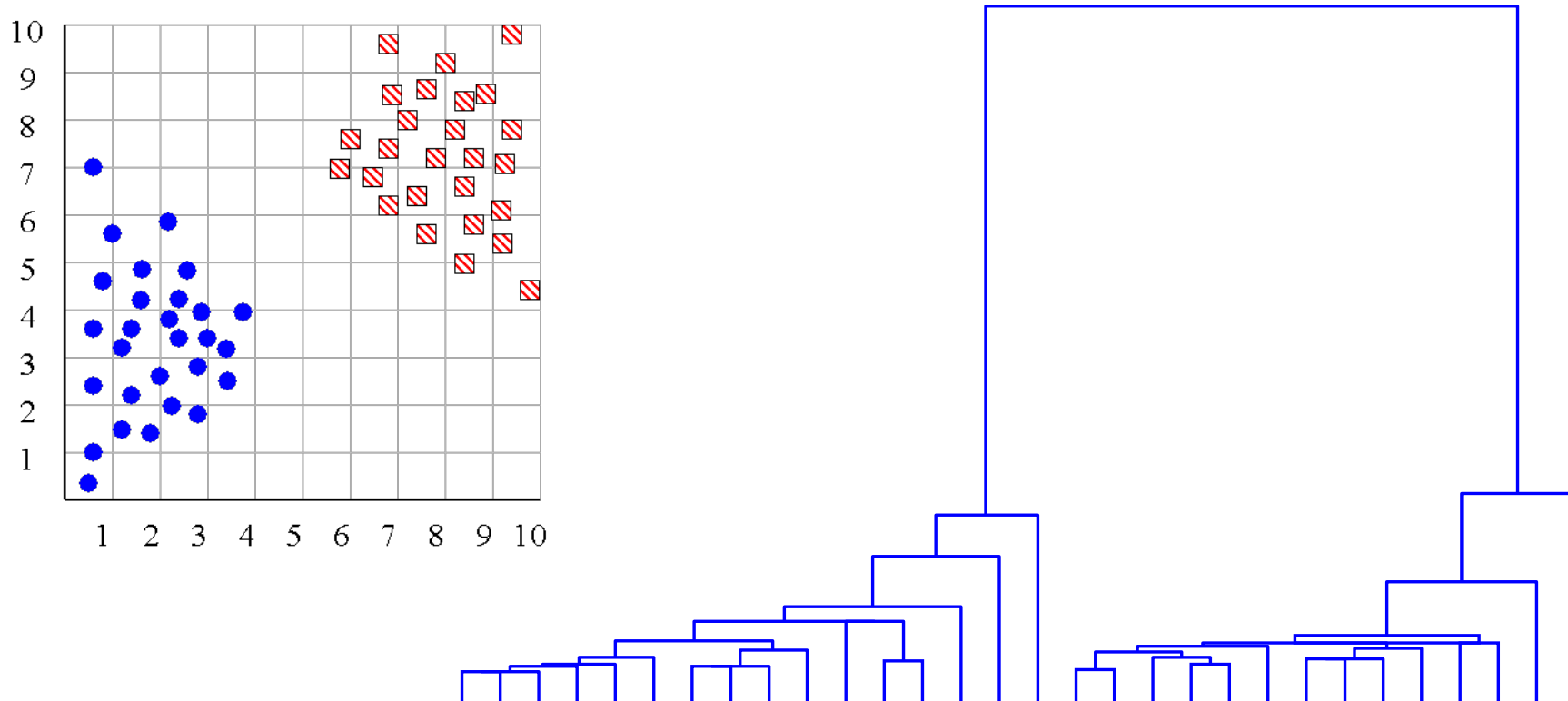


Demo: hierarchical distance using string edit distance



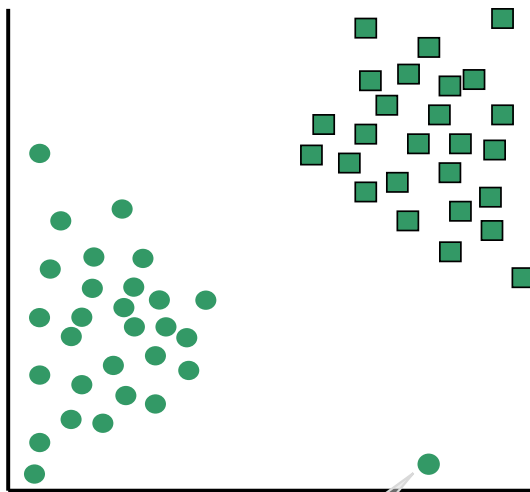
Dendrogram

A dendrogram can help determine the “correct” number of clusters

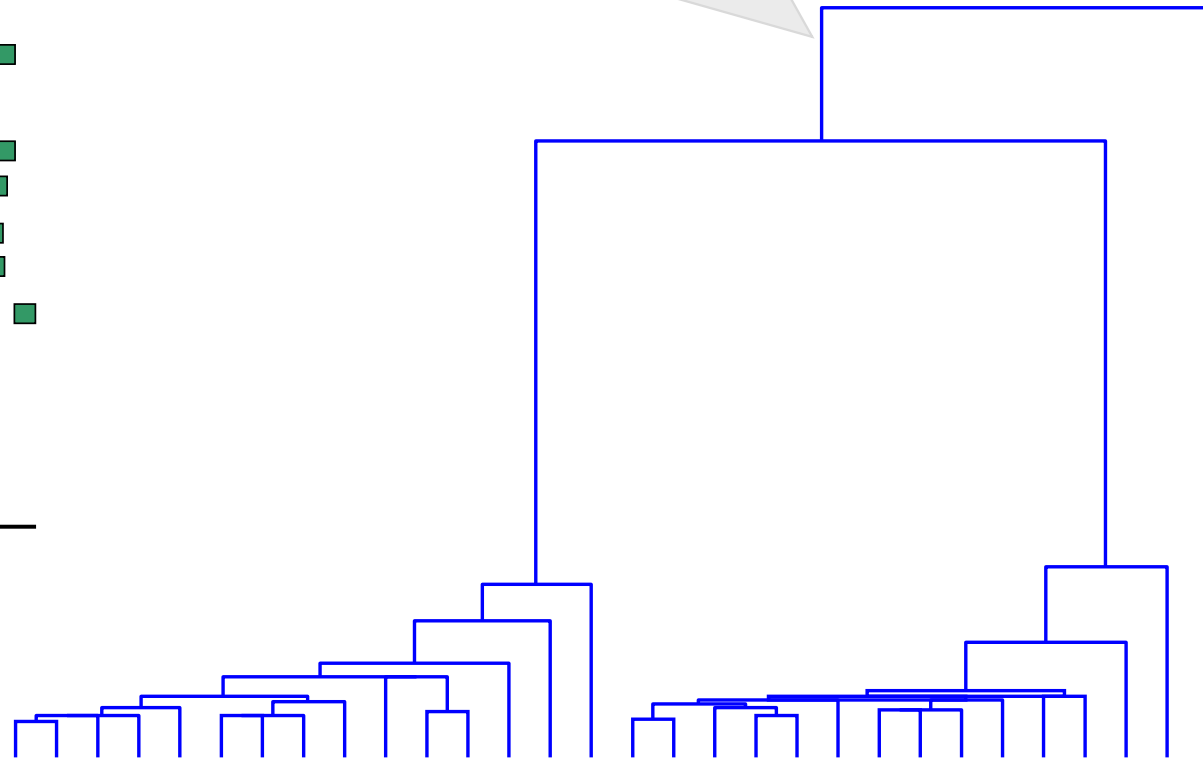


Using dendrograms to detect outliers

The single isolated branch is suggestive of a data point that is very different to all others



Outlier



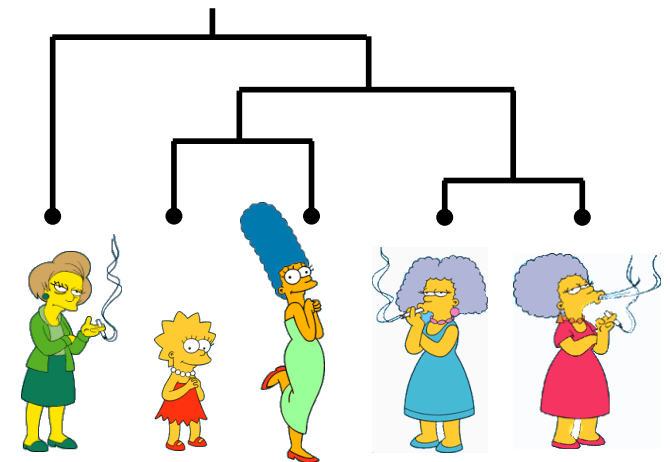
Hierarchical clustering

Since we cannot test all possible trees, we will have to use heuristics to search of all possible trees. We could do this:

- **Bottom-Up (agglomerative):** Typically starts by letting each object form its own clusters and iteratively merges clusters into larger and larger clusters
- **Top-Down (divisive):** Starts with all objects in one cluster. It then divides the root cluster into several smaller subclusters, and recursively partitions those clusters into smaller ones

The number of dendrograms with n leafs = $(2n-3)! / [(2^{(n-2)}) (n-2)!]$

Number of Leafs	Number of Possible Dendrograms
2	1
3	3
4	15
5	105
...	...
10	34,459,425













Building a hierarchical cluster

We begin with a distance matrix which contains the distances between every pair of objects in our database

$$D(\text{Mrs. Simpson}, \text{Lisa Simpson}) = 8$$

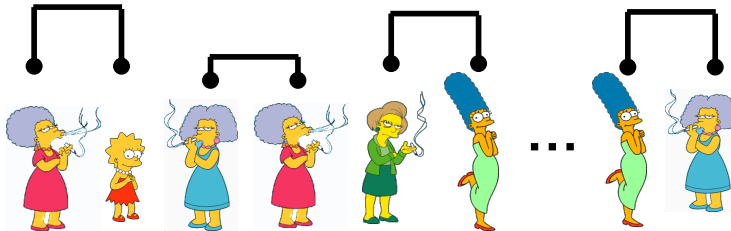
$$D(\text{Marge Simpson}, \text{Bart Simpson}) = 1$$

				
0	8	8	7	7
	0	2	4	4
		0	3	3
			0	1
				0
				

Bottom-up (agglomerative)

Typically starts by letting each object form its own cluster and iteratively merges clusters into larger and larger clusters

Consider
all possible
merges...



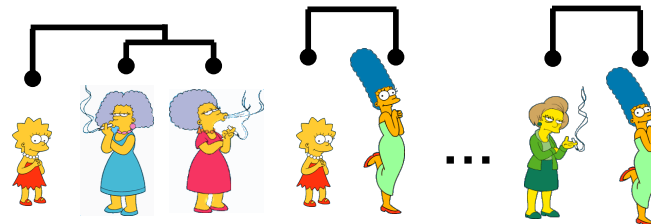
Choose
the
best



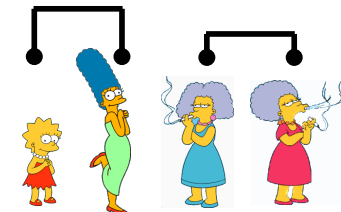
Bottom-up (agglomerative)

Typically starts by letting each object form its own cluster and iteratively merges clusters into larger and larger clusters

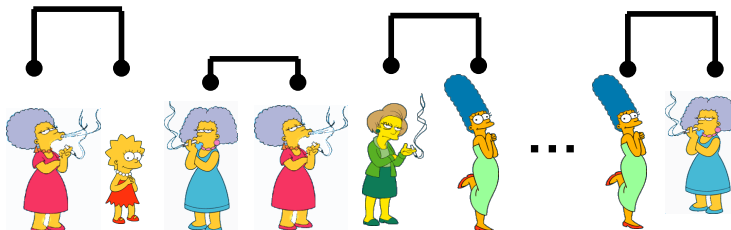
Consider
all possible
merges...



Choose
the
best



Consider
all possible
merges...



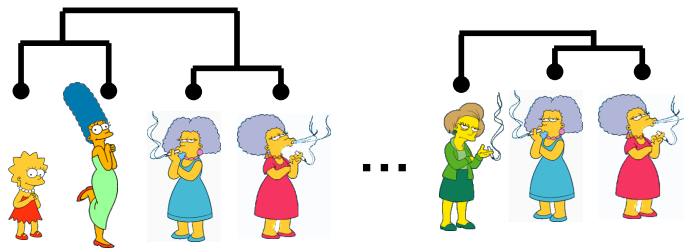
Choose
the
best



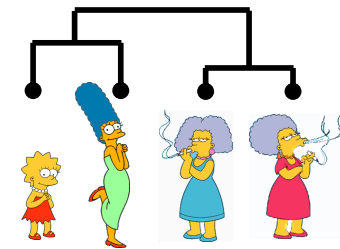
Bottom-up (agglomerative)

Typically starts by letting each object form its own cluster and iteratively merges clusters into larger and larger clusters

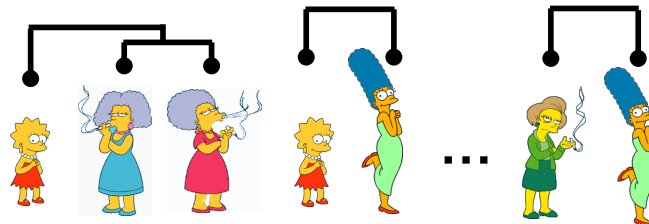
Consider all possible merges...



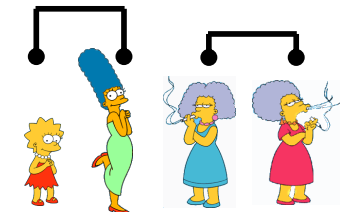
Choose the best



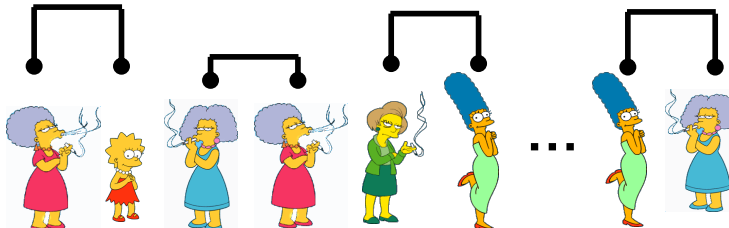
Consider all possible merges...



Choose the best



Consider all possible merges...

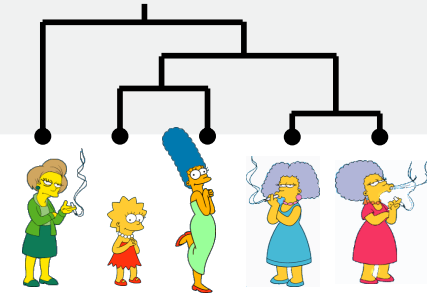


Choose the best

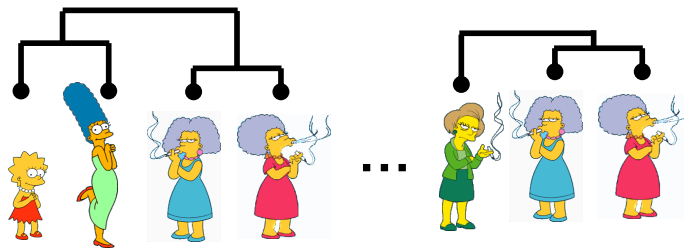


Bottom-up (agglomerative)

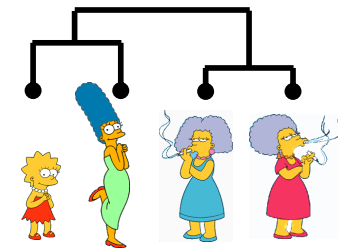
Typically starts by letting each object form its own cluster and iteratively merges clusters into larger and larger clusters



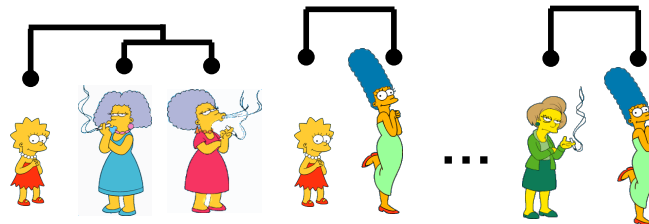
Consider
all possible
merges...



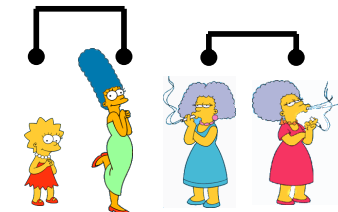
Choose
the
best



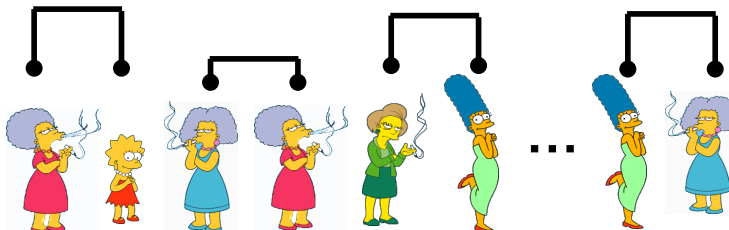
Consider
all possible
merges...



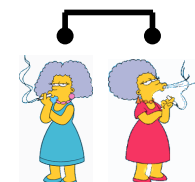
Choose
the
best



Consider
all possible
merges...



Choose
the
best

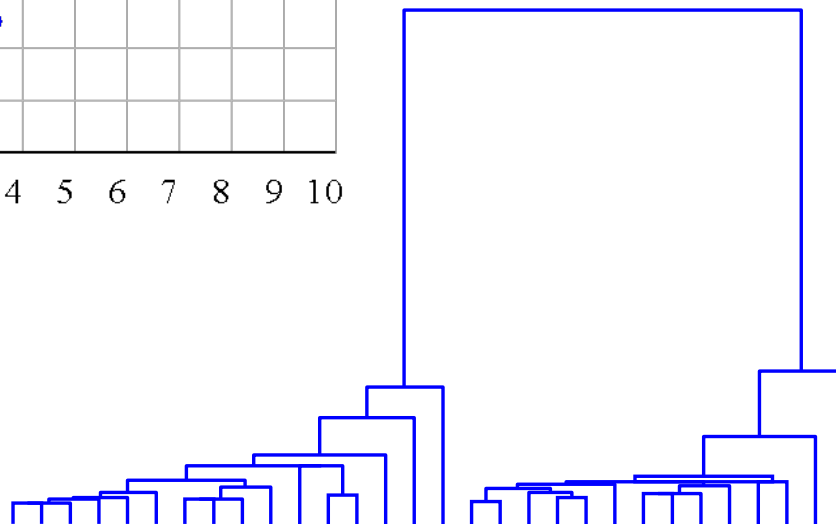
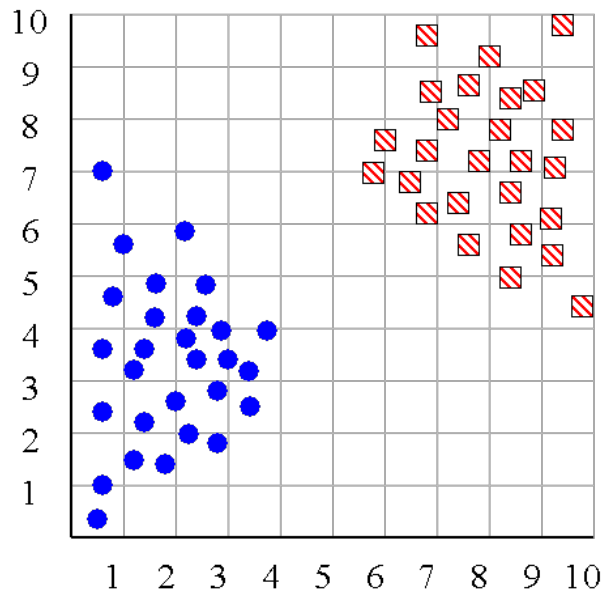


Measuring the distance between two clusters

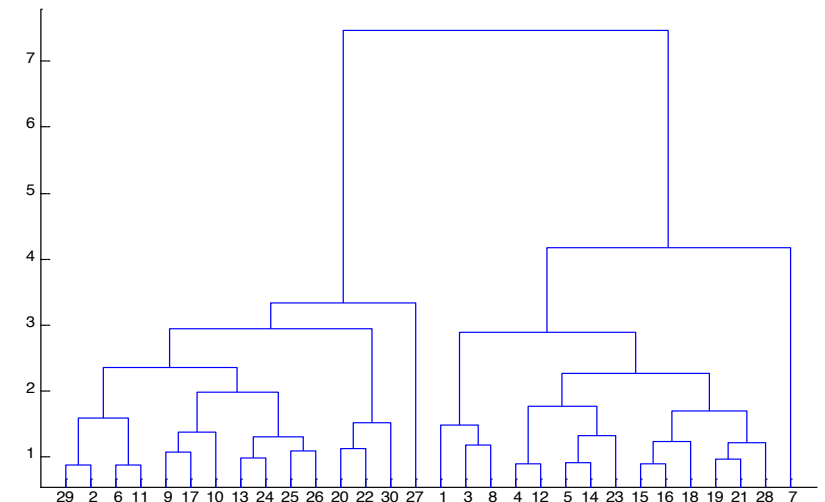
Measuring the distance between two clusters is not so obvious as measuring the distance between two objects. Several methods exists:

- **Single linkage (nearest neighbor):** the distance between two clusters is determined by the distance of the two closest objects (nearest neighbors) in the different clusters
- **Complete linkage (furthest neighbor):** the distances between clusters are determined by the greatest distance between any two objects in the different clusters (i.e., by the "furthest neighbors")
- **Group average linkage:** the distance between two clusters is determined as the average distance between all pairs of objects in the two different clusters

Measuring the distance between two clusters



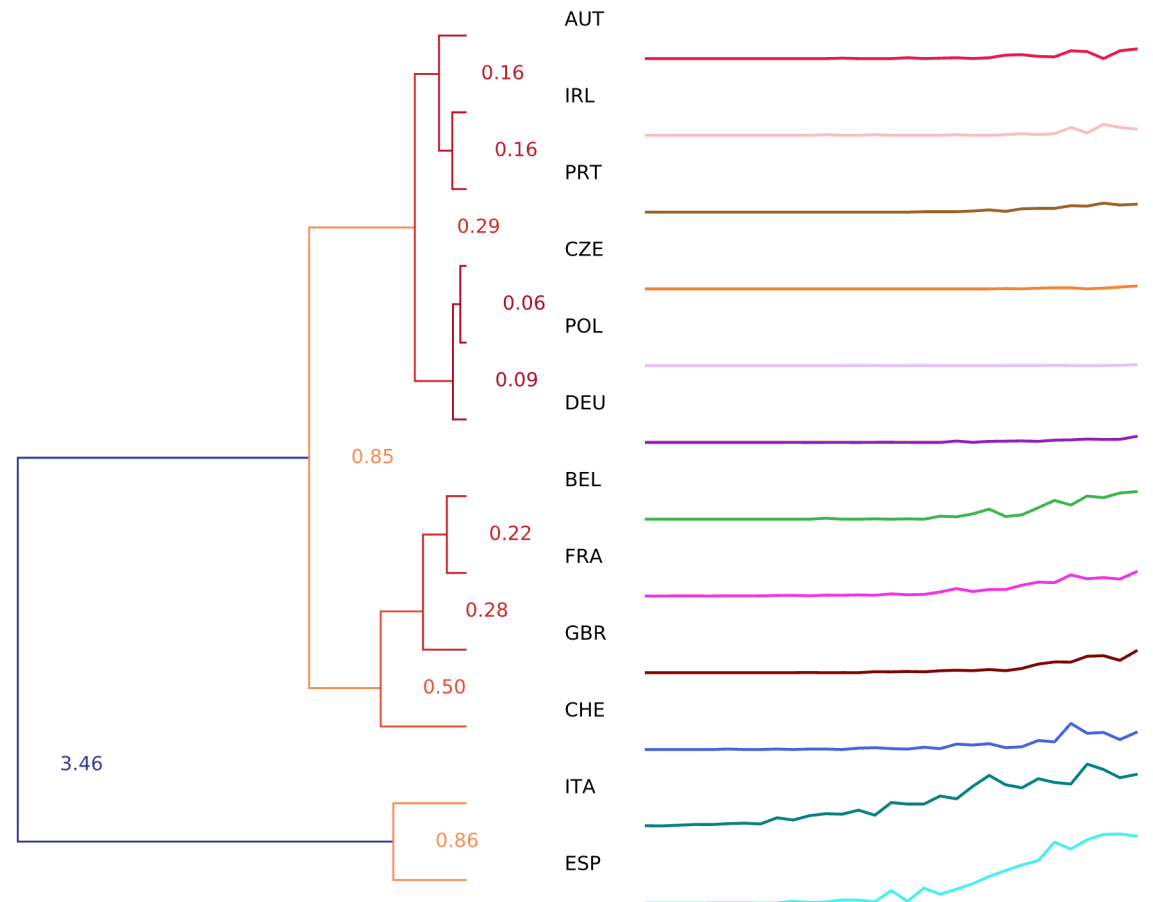
Single linkage



Average linkage

Hierarchical clustering algorithms

- BIRCH: Multiphase Hierarchical Clustering Using Clustering Feature Trees
- Chameleon: Multiphase Hierarchical Clustering Using Dynamic Modeling
- Probabilistic Hierarchical Clustering



Density-based methods

Clustering methods

General characteristics

- Clusters are dense regions of objects in space that are separated by low-density regions
- Cluster density: each data point must have a minimum number of data points within its “neighborhood”
- May filter out outliers

STRENGTHS

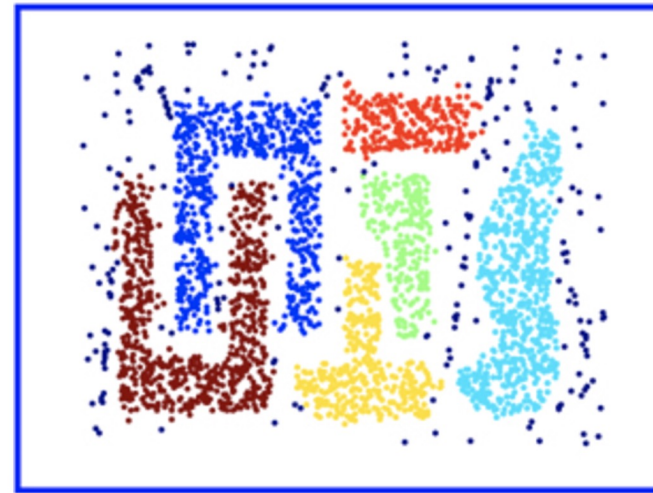
- Can discover arbitrarily shaped clusters
- Find cluster surrounded by other clusters

WEAKNESSES

- Struggles with high dimensionality data
- Results are not good with sparse datasets or varying densities
- Sampling affects density measures

Density-based algorithms

- DBSCAN: Density-Based Clustering Based on Connected Regions with High Density
- OPTICS: Ordering Points to Identify the Clustering Structure
- DENCLUE: Clustering Based on Density Distribution Functions



DBSCAN visualization – source: <https://medium.com/@elutins/dbscan-what-is-it-when-to-use-it-how-to-use-it-8bd506293818>

Grid-based methods

Clustering methods

General characteristics

- Use a multiresolution grid data structure
- Unlike other methods, instead of being data-driven, grid-based methods partitions the space into cells, independent of the distribution of the objects

STRENGTHS

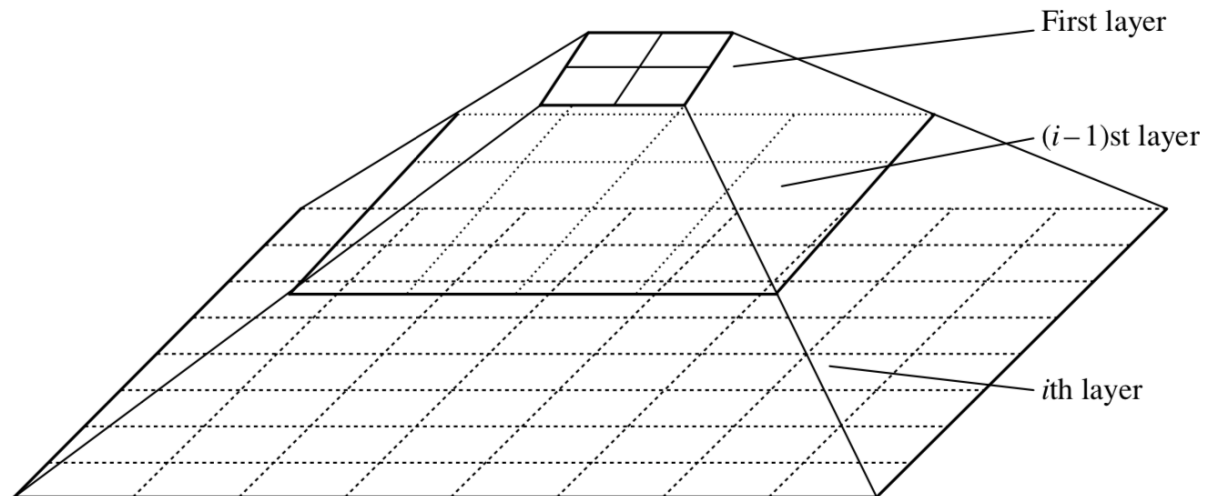
- Fast processing time (typically independent of the number of data objects, yet dependent on grid size)

WEAKNESSES

- Clustering does not take in consideration the data objects distribution

Grid-based algorithms

- STING: Statistical Information Grid
- CLIQUE: An Apriori-like Subspace Clustering Method



Evaluating clustering quality

Modeling: Clustering

Extrinsic methods

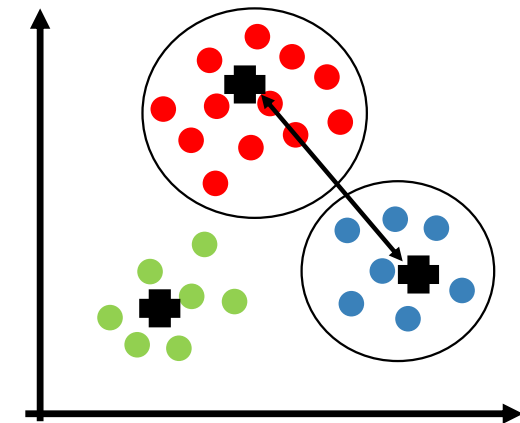
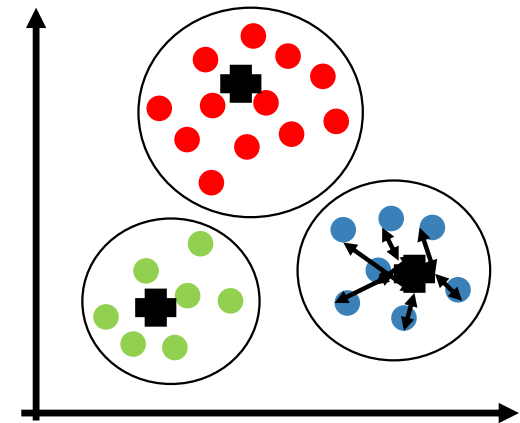
Extrinsic: when the “ground truth” is available (supervised method). These measures, such as *Bcubed Precision* and *Bcubed Recall*, should validate four criteria:

- **Cluster homogeneity:**
 - A clustering satisfies homogeneity if all of its clusters contain only instances which are members of a single class
- **Cluster completeness:**
 - A clustering satisfies completeness if all instances that are members of a given class are of the same cluster
- **Rag bag (miscellaneous):**
 - Putting a heterogeneous object into a pure cluster should be penalized more than putting it into a rag bag
- **Small cluster preservation:**
 - Splitting a small class into pieces is more harmful than splitting a large class into pieces

Intrinsic methods

Intrinsic: when the “ground truth” is not available (unsupervised method). Evaluate how well the clusters are separated and how compact they are (e.g., the *silhouette coefficient*).

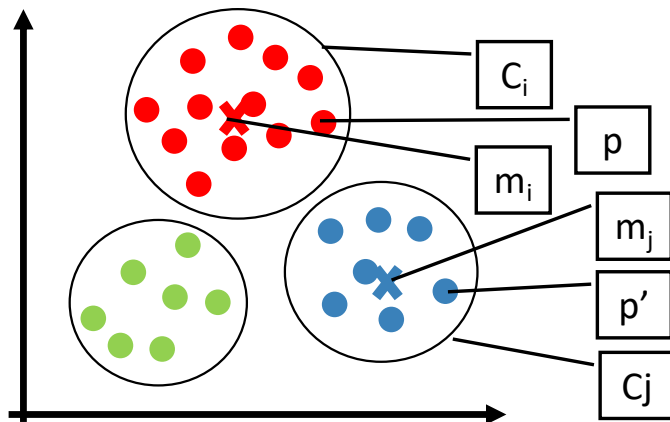
- **Intra-cluster distance:** the distance between data points of the same cluster should be small
- **Inter-cluster distance:** the distance between data points of different clusters should be large



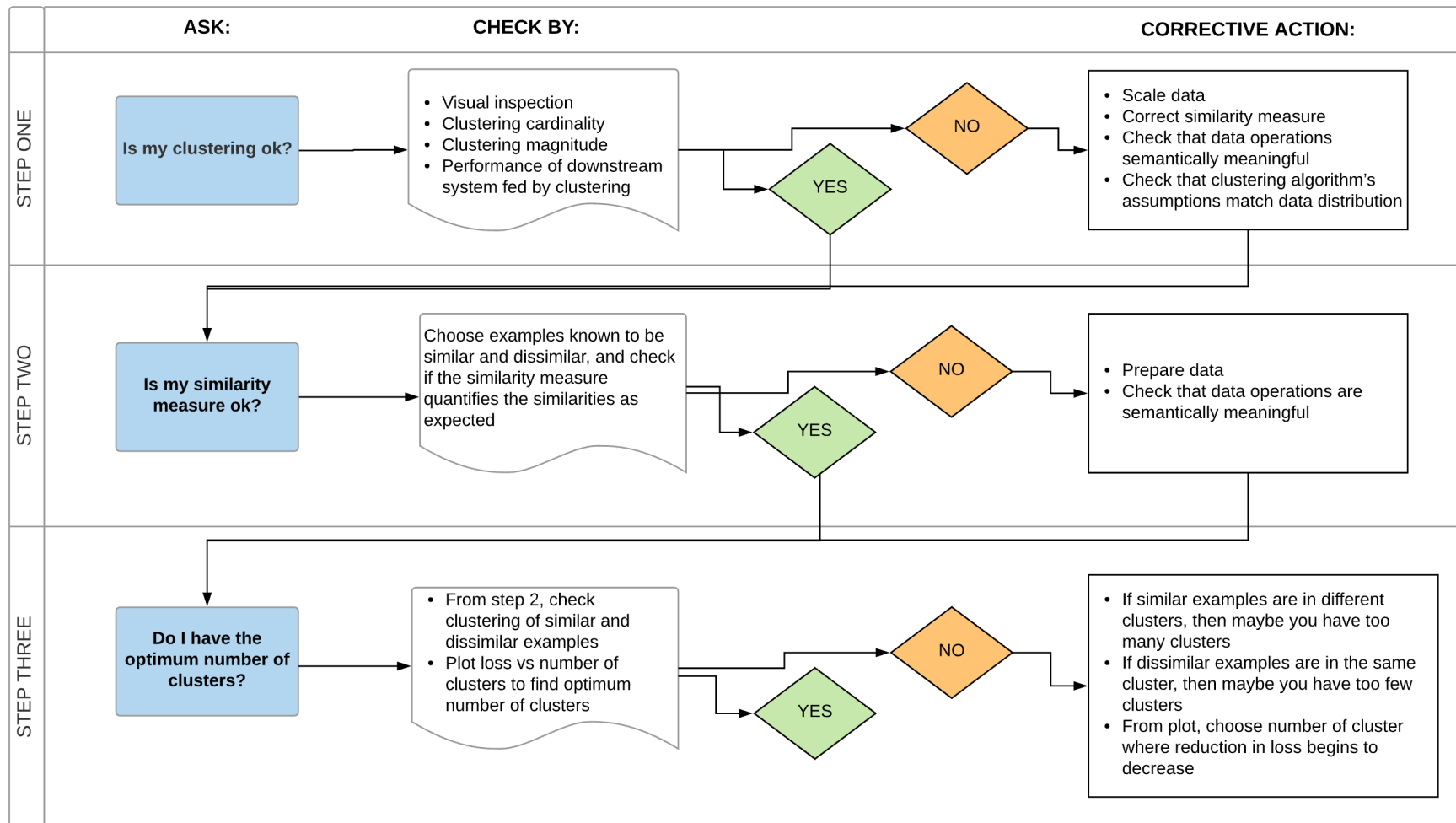
Distance measures in algorithmic methods

Measures for distance between clusters where $|p - p'|$ is the distance between to objects or data points; m_i is the mean of cluster C_i ; and n_i is the number of objects or data points in C_i . These measures are also known as *linkage measures*

- Minimum distance: $dist_{min}(C_i, C_j) = \min_{p \in C_i, p' \in C_j} \{|p - p'|\}$
- Maximum distance: $dist_{max}(C_i, C_j) = \max_{p \in C_i, p' \in C_j} \{|p - p'|\}$
- Mean distance: $dist_{mean}(C_i, C_j) = |m_i - m_j|$
- Average distance: $dist_{avg}(C_i, C_j) = \frac{1}{n_i n_j} \sum_{p \in C_i, p' \in C_j} |p - p'|$



How to evaluate the clustering quality

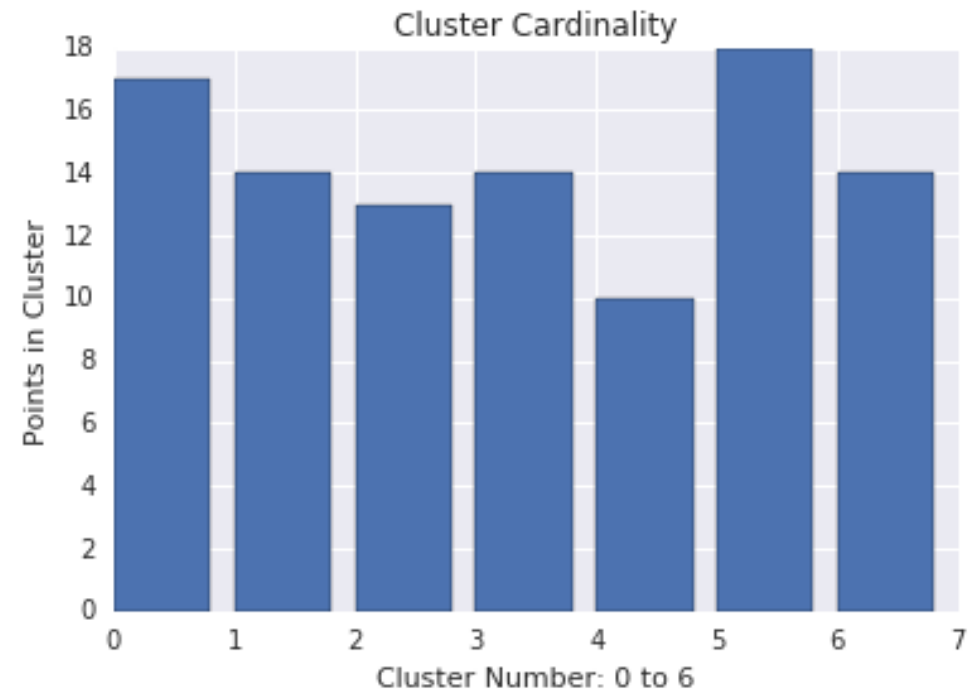


How to evaluate the clustering quality

Step 1: Cluster cardinality

Plot the cluster cardinality (number of objects per cluster) and investigate clusters that are outliers.

For example, in the figure, investigate cluster 5

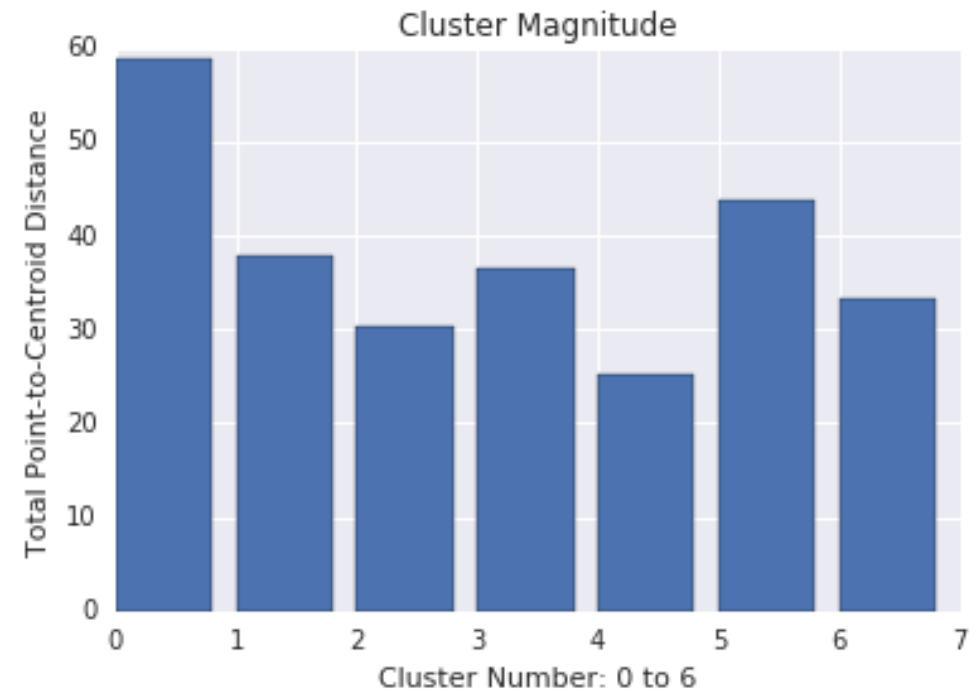


How to evaluate the clustering quality

Step 1: Cluster magnitude

Plot the cluster magnitude (sum of distances from all objects to the centroid of the cluster) and investigate clusters that are outliers or anomalies

For example, in the figure, investigate cluster 0

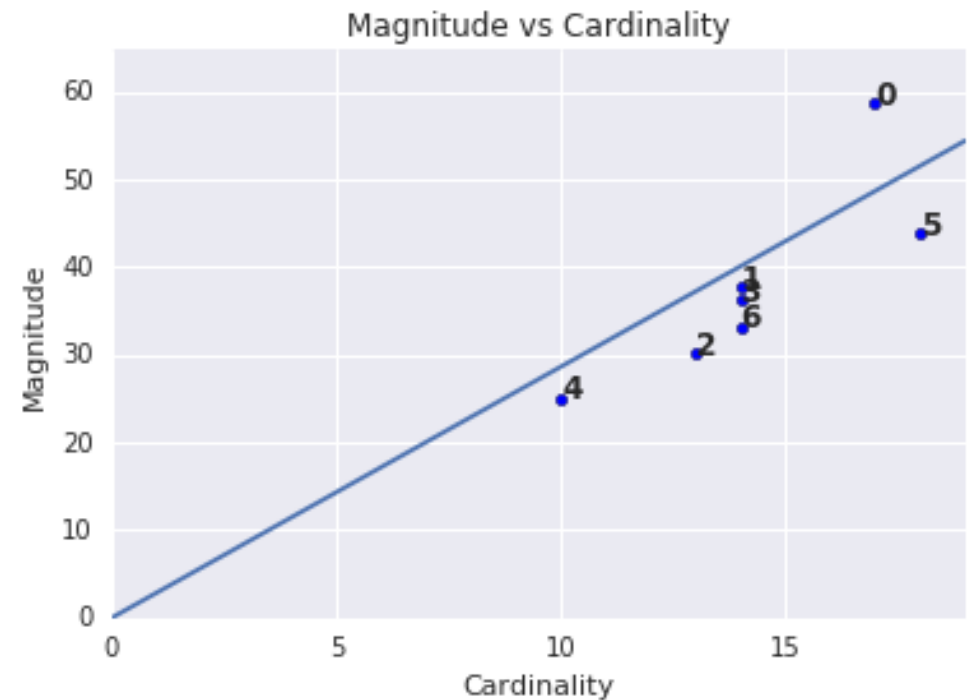


How to evaluate the clustering quality

Step 1: Magnitude vs. cardinality

A higher cardinality tends to result in a higher magnitude, which makes sense. Clusters are anomalous when cardinality does not correlate with magnitude.

For example, in the figure, cluster 0 seems to be anomalous



Considerations on how to improve clustering

- Step 1:
 - Check if data is normalized
 - Check the similarity measure employed
 - Check the data preparation
- Step 2:
 - Check for pairs of known object examples and compare them using different measures to understand which one is the best
- Step 3:
 - In algorithms where you need to define the number of clusters (e.g., k-means) check with different number of clusters.

Questions?

Hierarchical clustering examples adapted from Eamonn Keogh

Cluster evaluation examples adapted from developers.google.com

Machine Learning for Marketing

© 2020-2023 Nuno António (rev. 2023-02-09)

Acreditações e Certificações



Instituto Superior de Estatística e Gestão da Informação
Universidade Nova de Lisboa