TÉCNICO+
FORMAÇÃO AVANÇADA

Clustering (1/2)

Introduction to clustering

DASH: Data Science e Análise Não Supervisionada

Rui Henriques, rmch@tecnico.ulisboa.pt
Instituto Superior Técnico, Universidade de Lisboa

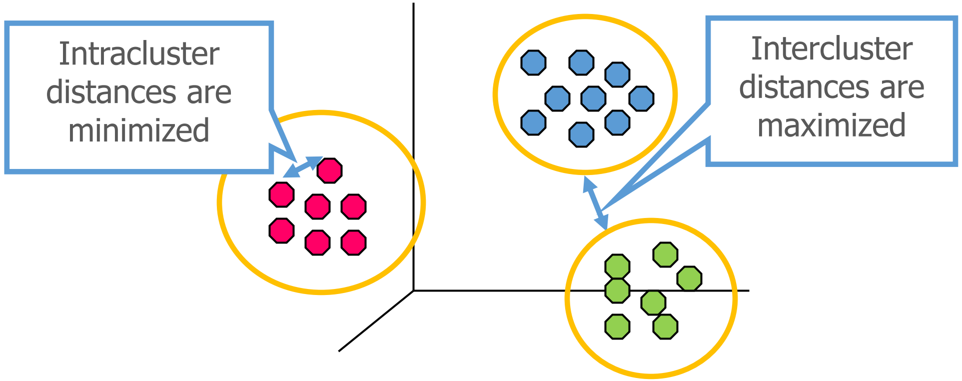
Outline

- Introduction to clustering
- Multivariate similarity metrics
- Approaches
  - hierarchical
  - density-based
- From multivariate to complex data structures
- Evaluation
  - intrinsic metrics
  - extrinsic metrics

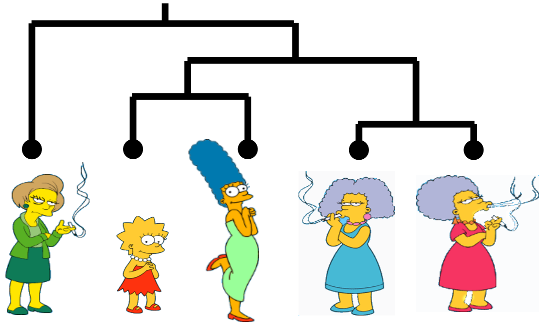
Clustering

Cluster: group of observations

Cluster analysis: group observations into clusters according to their (dis)similarity: observations in the same cluster are more similar than those in different clusters



Motivation



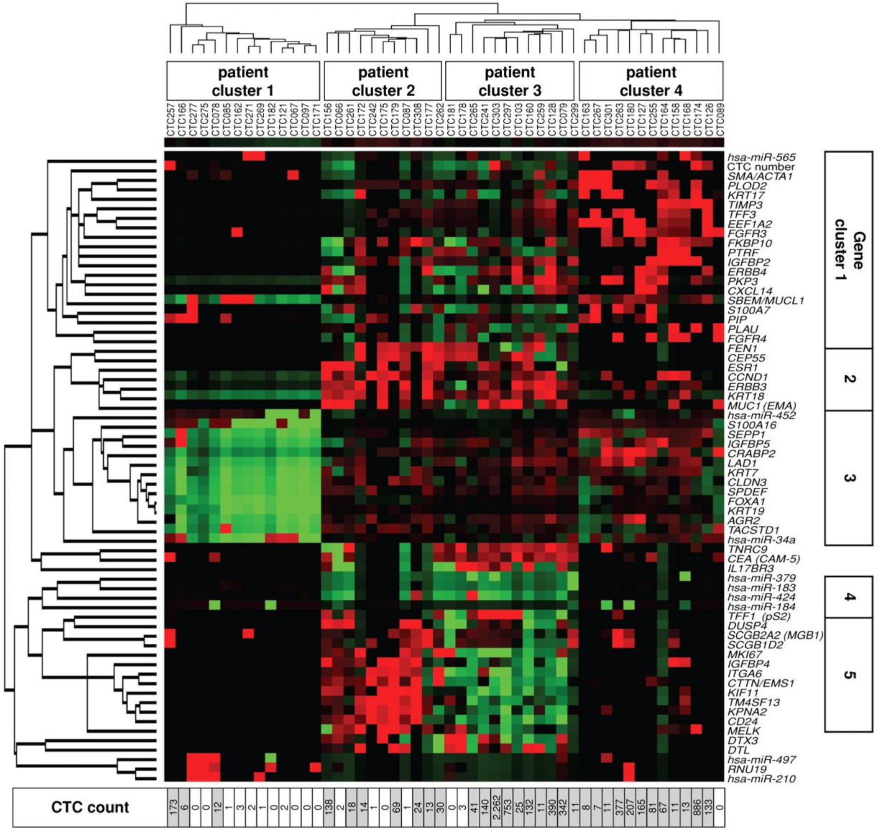
- **Patients** with a shared clinical condition:
  How to **understand disease**?
  - cancer types, dementia progression, risk groups
  - stratified diagnostics and therapeutics
- **Customers**: how to segment their profile for **personalized marketing**?
- **Webpages**, shopping products, **media**, **documents**:
  how to categorize them for **recommendations**?
- **Genes**, proteins and metabolites with different expression and concentration profile: how to understand their correlated behavior (biological functions)?
- **Students**, researchers, professors: how to improve science and **education**?

Motivation

- **ENDS**
  - **Insight** into the underlying structure/regularities of data
  - **Preprocessing** step for other tasks
  - **Supporting prediction** by stratifying populations (exercise: *how*?)
  - **Improving efficiency** by using clusters as a proxy for observations
  - Many others…

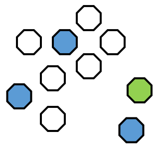
- Application **DOMAINS**
  - **Information retrieval**: document and webpage clustering
  - **Marketing**: customer groups according to profile and product-receptivity
  - **Insurance**: policy holders with different average claim costs
  - **Medicine**: risk groups, personalized medicine
  - **Biology**: philogenetics, pathways, regulatory modules
  - Others: city-planning, land use, seismic studies, atmospheric conditions

Illustration



Clustering modes

- **Unsupervised** (*default*)
  - cluster observations without knowning their labels
- *Semi-supervised*
  - cluster observations when:
    - the labels of some observations may be known *or*
    - pairs of observations are known to belong to the same cluster



- *Supervised*
  - cluster observations when targets are considered, e.g.:
    - label added as an additional input variable
    - cluster class-conditional observations

Clustering modes

- Deterministic versus probabilistic cluster stances
  - **hard** solutions: each observation either belongs or not to a given cluster
  - **soft** solutions: each observation has a probability (membership) of belonging to a given cluster
    - fuzzy and model-based clustering
- Separation of clusters: **exclusive** *versus* **non-exclusive** (overlapping clusters)
- **Complete** versus **partial** (observations may not belong to any cluster)
- **Uniform** versus **weighted**
  - variables can be weighted based on data semantics/domain knowledge
  - observations can be weighted based on relevance criteria

Motivation

Two major factors impact solutions: ***distance + approach***

- **distance metrics** depend on the:
  - **variable domains**
    - *numeric* and *ordinal* (e.g. Euclidean)
    - *nominal* (e.g. Hamming)
    - *non-iid attributes*
  - **data structure**: tabular, time series, image, spatiotemporal data, events…
- **approach**
  - partitioning
  - hierarchical
  - density-based
  - model-based

Outline

- Introduction to clustering
- **Multivariate similarity metrics**
- Approaches
  - hierarchical
  - density-based
- From multivariate to complex data structures
- Evaluation
  - intrinsic metrics
  - extrinsic metrics

Focal point: distances

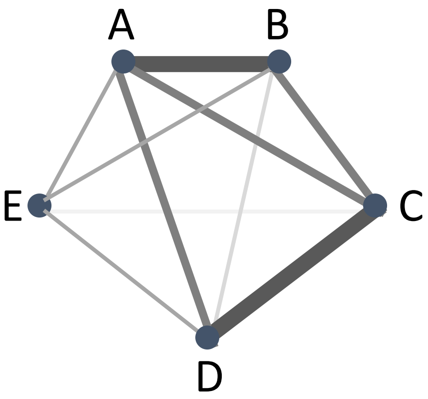
- well-established distances can be applied yet…
  …best distances are generally ***customized*** to the problem domain (background knowledge)
  - e.g. demographic $\text{dist}(ind_1, ind_2) = \frac{age_1 - age_2}{20} + \mathbf{1}[region_1 = region_2] \times 0.8 + 1[sex_1 = sex_2] \times 1.2 + \cdots$
- apply distance to produce pairwise **distance matrices** between observations (and/or clusters)
- similarity matrix = − distance matrix

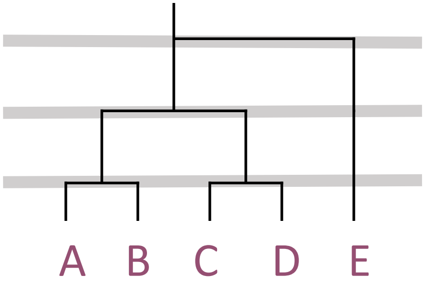
	A	B	C	D	E	F
A	0	0.71	5.66	3.61	4.24	3.20
B	0.71	0	4.95	2.92	3.54	2.50
C	5.66	4.95	0	2.24	1.41	2.50
D	3.61	2.92	2.24	0	1.00	0.50
E	4.24	3.54	1.41	1.00	0	1.12
F	3.20	2.50	2.50	0.50	1.12	0

Clustering as a graph-based task

- Proximity between all data observations defines a weighted graph
- Nodes are the observations, edges capture their distances
- Clustering = breaking the graph into connected components
- Minimize the edge weight between clusters AND maximize the edge weight within clusters
  - How? Incremental grouping using thresholds

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





Distances and metrics

A distance function is a **metric** if the following conditions are met:

- non-negative

  $d(x, y) \geq 0$

- distance to point itself is zero

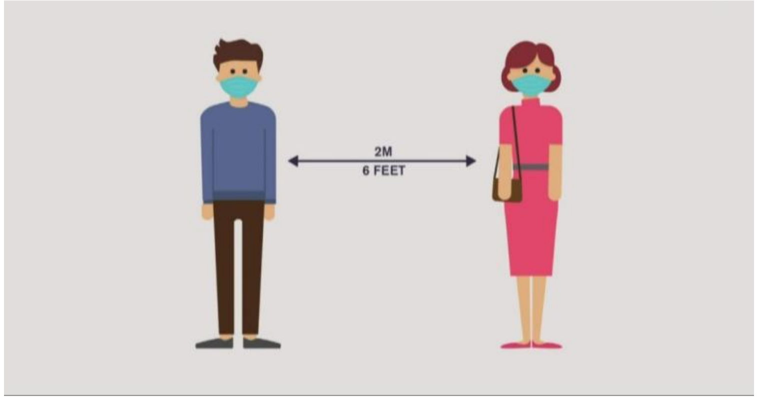
  $d(x, x) = 0$

- symmetry

  $d(x, y) = d(y, x)$

- triangular inequality

  $d(x, y) \leq d(x, z) + d(z, y)$



Common distance metrics
(numeric data)

Minkowski distance

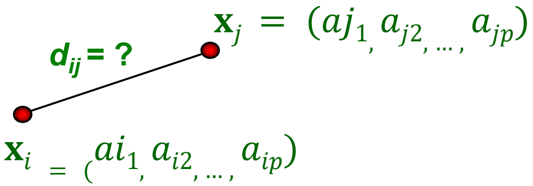
$$d(i,j) = \sqrt[q]{\underbrace{|a_{i1} - a_{j1}|^q}_{1^{st}\text{ dimension}} + \underbrace{|a_{i2} - a_{j2}|^q}_{2^{nd}\text{ dimension}} + \ldots + \underbrace{|a_{ip} - a_{jp}|^q}_{p^{th}\text{ dimension}}}$$

Euclidean distance $(q = 2)$

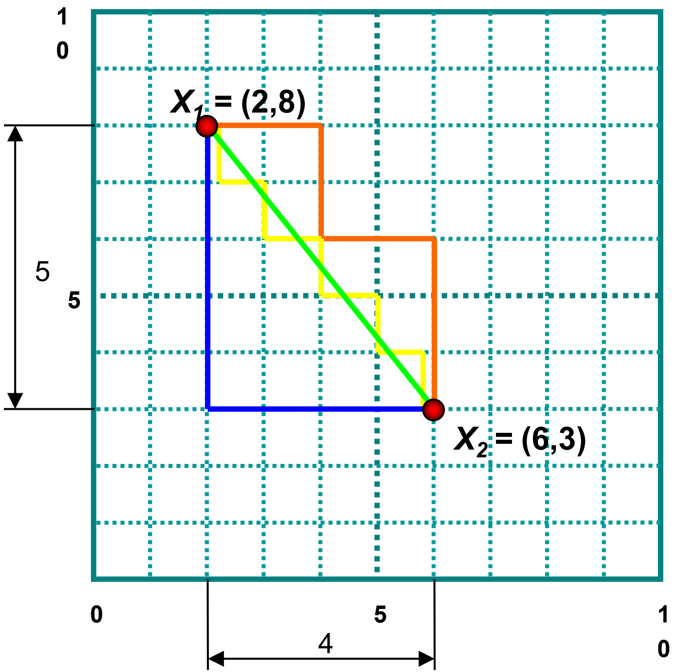
$$d(i,j) = \sqrt{|a_{i1} - a_{j1}|^2 + |a_{i2} - a_{j2}|^2 + \ldots + |a_{ip} - a_{jp}|^2}$$

Manhattan distance $(q = 1)$

$$d(i,j) = |a_{i1} - a_{j1}| + |a_{i2} - a_{j2}| + \ldots + |a_{ip} - a_{jp}|$$



Common distance metrics
(numeric data)



2D example

$x_1 = (2,8)$

$x_2 = (6,3)$

Euclidean distance

$$d(1,2) = \sqrt{|2-6|^2 + |8-3|^2} = \sqrt{41}$$

Manhattan distance

$$d(1,2) = |2-6| + |8-3| = 9$$

Chebyshev distance
(numeric data)

- when $q \rightarrow \infty$, the metric highly penalizes maximum attribute errors
- useful if the worst case must be avoided:

$$d_\infty(\mathbf{x},\mathbf{y}) = \lim_{q\to\infty}\left(\sum_{i=1}^{n}|x_i - y_i|^q\right)^{1/q} = \max(|x_1 - y_1|, |x_2 - y_2|, \dots, |x_n - y_n|)$$

Example:

$$d_\infty((2,8),(6,3)) = \max(|2-6|, |8-3|) = \max(4,5) = 5$$

Correlation

- positive (negative): two variables vary in the same (opposite) way
  - maximum value of 1 (-1) if X and Y are perfectly direct (inverse) correlated
- *recall*: **Pearson** and **Spearman** coefficients for numeric data
  - how to handle categorical or mixed data?
- example: gene expression data clustering

  g1 = (1,2,3,4,5)

  g2 = (100,200,300,400,500)

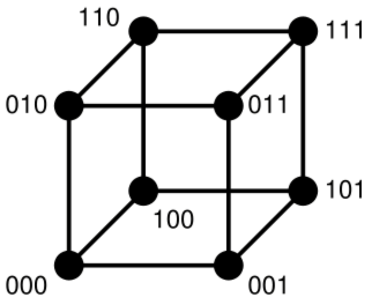
  g3 = (5,4,3,2,1)

  Which genes are similar according to correlation coefficients?

Hamming distance
(binary and categorical data)

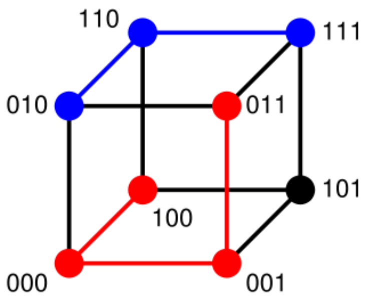
- number of different attribute values
- distance of (1011101) and (1001001) is 2
- distance between (toned) and (roses) is 3

3-bit binary cube



100->011 has distance 3 (red path)

010->111 has distance 2 (blue path)



Outline

- Introduction to clustering
- Multivariate similarity metrics
- **Approaches**
  - hierarchical
  - density-based
- From multivariate to complex data structures
- Evaluation
  - intrinsic metrics
  - extrinsic metrics

Approaches

Partitioning:

- Create partitions and iteratively update them (e.g. *k*-means, *k*-modes, *k*-medoids)

Hierarchical:

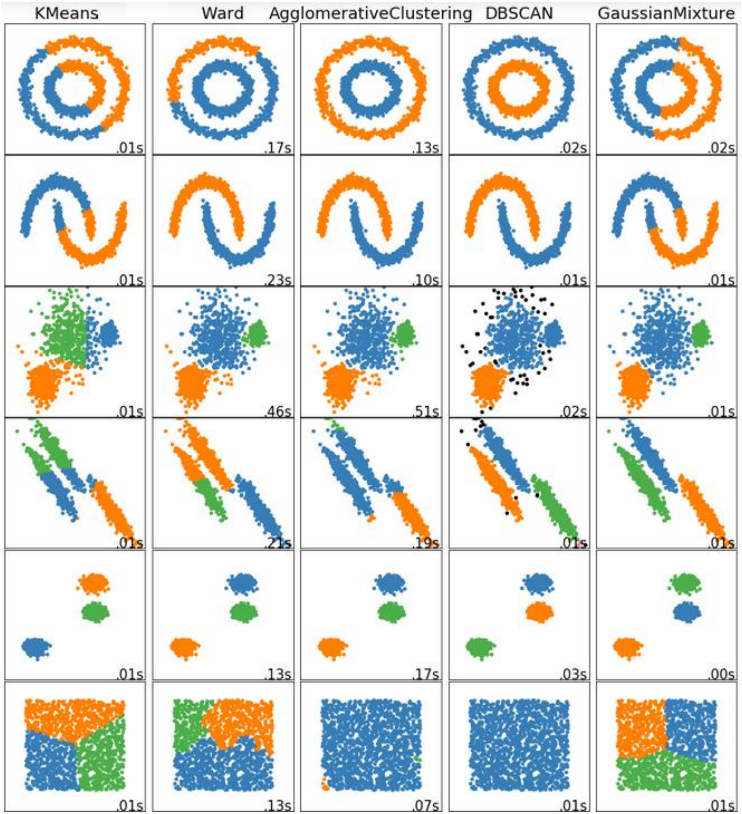
- Create hierarchical decomposition of data points (e.g. Diana, Agnes)

Density-based:

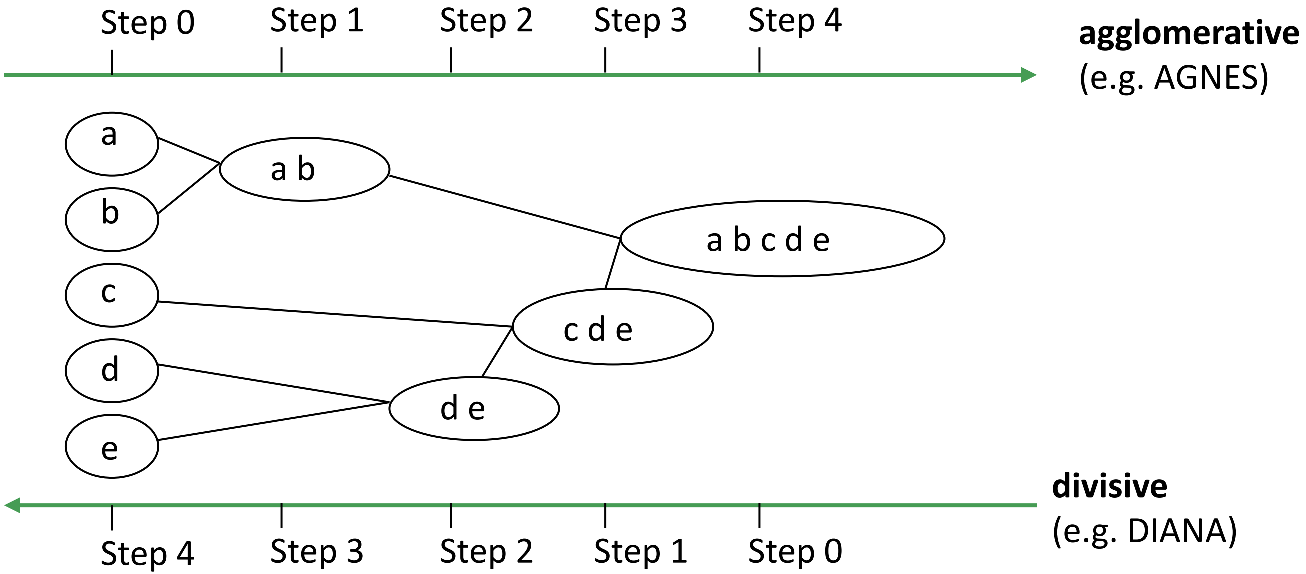
- Group points based on connectivity and density (e.g. DBSACN, DenClue)

Model-based:

- Data are seen as a mixture of distributions (e.g. EM)



Hierarchical clustering



Hierarchical clustering

- **Agglomerative** (bottom-up)
  - initialize each point as its own cluster
  - iteratively merge clusters
- **Divisive** (top-down)
  - initialize all data points into one cluster
  - large clusters are successively divided

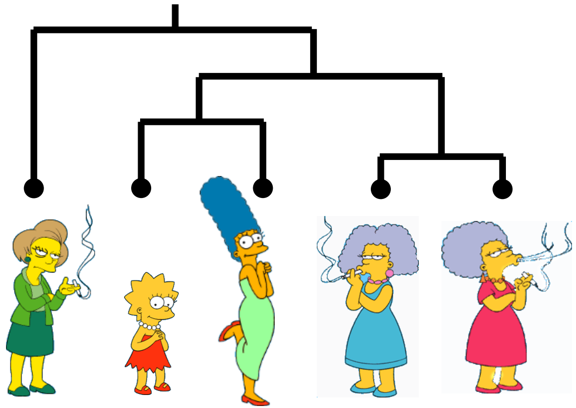
Hierarchical clustering

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

cannot test all possible trees

$\Longrightarrow$ heuristic searches



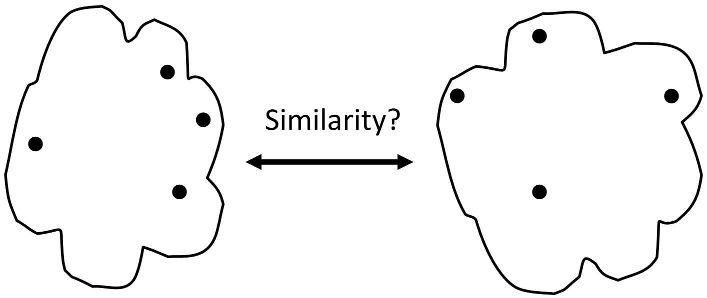
Cluster distance

- **Single link**: smallest distance between observations
- **Complete link:** largest distance between observations
- **Average link:** average distance between observations

$$d(c_i, c_j) = \frac{1}{|c_i||c_j|} \sum_{x_i \in C_i} \sum_{x_j \in C_j} d(x_i, x_j)$$

- **Centroid link:** distance between centroids
- **Ward**'s distance: similarity based on the error increase when two clusters are merged (sum of squared distances of points to closest centroid)

Cluster distance

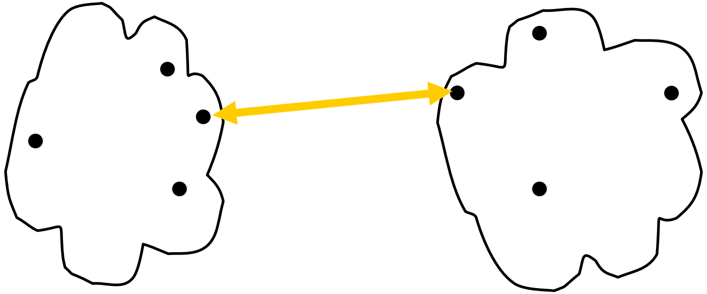


- MIN (single link)
- MAX (complete link)
- Average link
- Centroid link
- Ward's method

	x_1	x_2	x_3	x_4	x_5	…
x_1						
x_2						
x_3						
x_4						
x_5						
⋮						

similarity matrix

Cluster distance

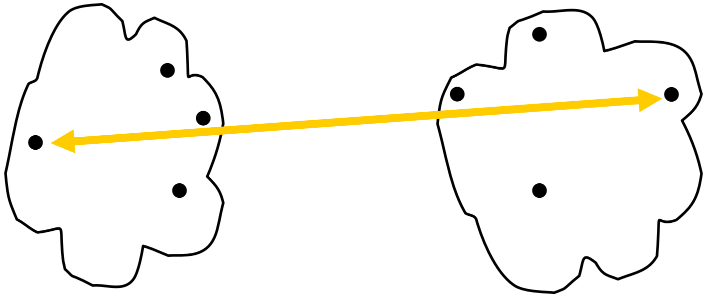


	x_1	x_2	x_3	x_4	x_5	…
x_1						
x_2						
x_3						
x_4						
x_5						
⋮						

similarity matrix

- **MIN** (**single link**)
- MAX (complete link)
- Average link
- Centroid link
- Ward’s method

Cluster distance

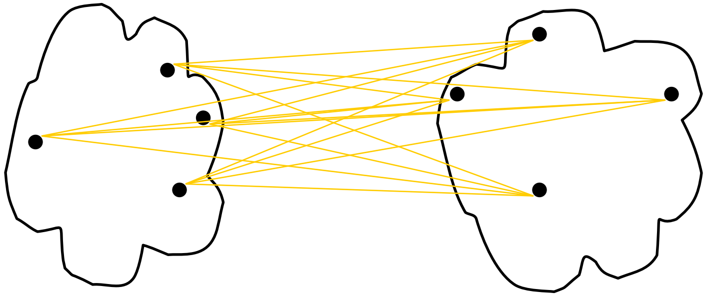


	x_1	x_2	x_3	x_4	x_5	…
x_1						
x_2						
x_3						
x_4						
x_5						
⋮						

similarity matrix

- MIN (single link)
- **MAX (complete link)**
- Average link
- Centroid link
- Ward’s method

Cluster distance

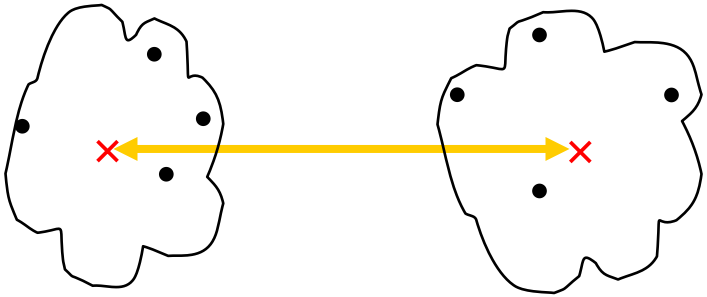


	x_1	x_2	x_3	x_4	x_5	…
x_1						
x_2						
x_3						
x_4						
x_5						
⋮						

similarity matrix

- MIN (single link)
- MAX (complete link)
- **Average link**
- Centroid link
- Ward's method

Cluster distance



- MIN (single link)
- MAX (complete link)
- Average link
- **Centroid link**
- Ward's method

	x_1	x_2	x_3	x_4	x_5	…
x_1						
x_2						
x_3						
x_4						
x_5						
$\vdots$						

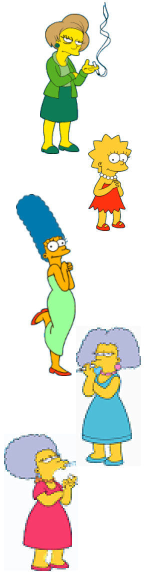
similarity matrix

Hierarchical clustering

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

d() = 8

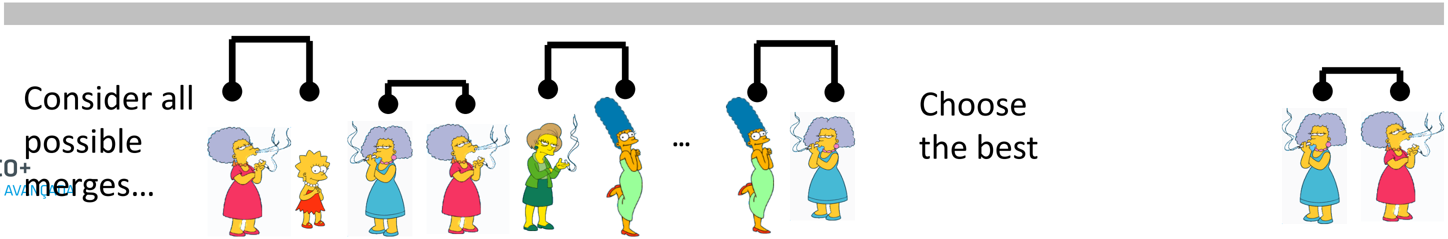
d() = 1



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

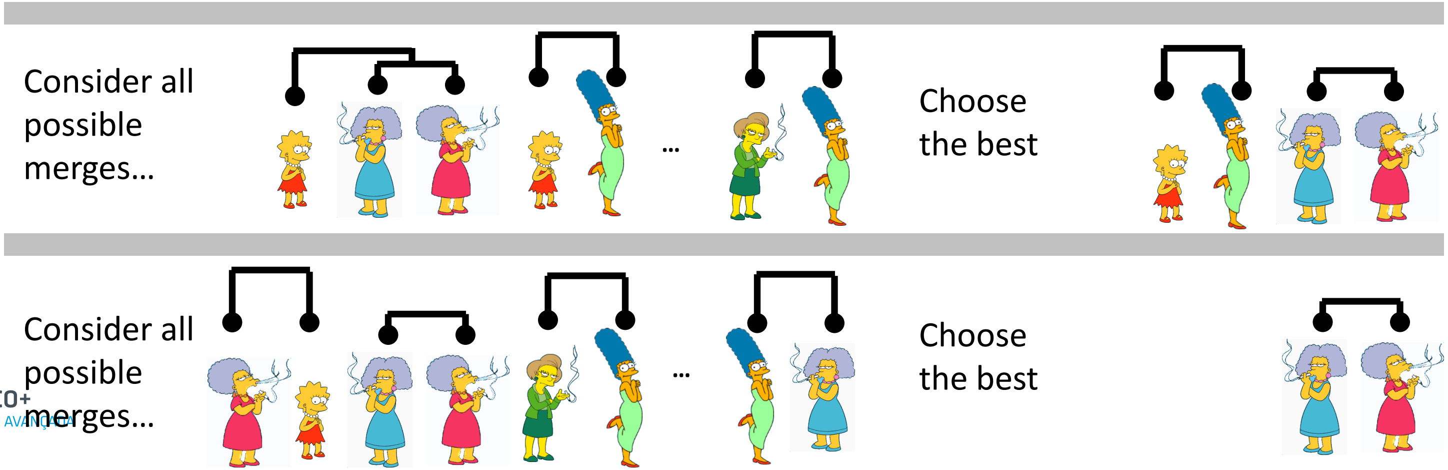
Hierarchical clustering

Bottom-up (agglomerative): Starting with each point as a cluster, find best pair. Repeat until all clusters are fused

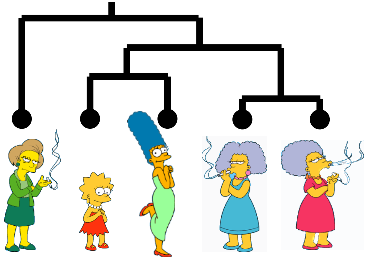


Hierarchical clustering

Bottom-up (agglomerative): Starting with each point as a cluster, find best pair. Repeat until all clusters are fused

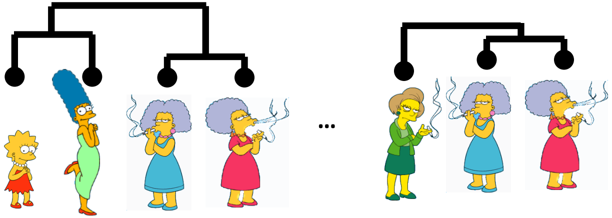


Hierarchical clustering

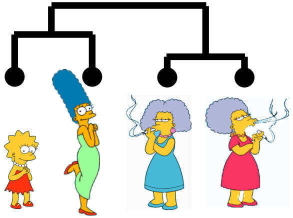


Bottom-up (agglomerative)

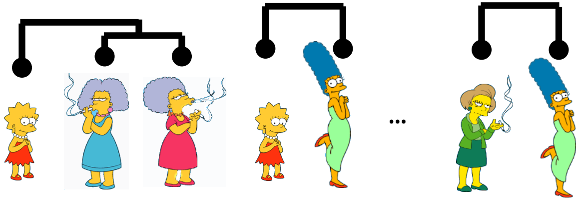
Consider all possible merges…



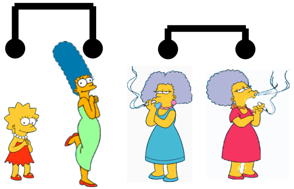
Choose the best



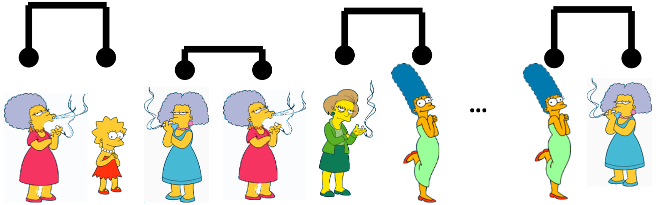
Consider all possible merges…



Choose the best



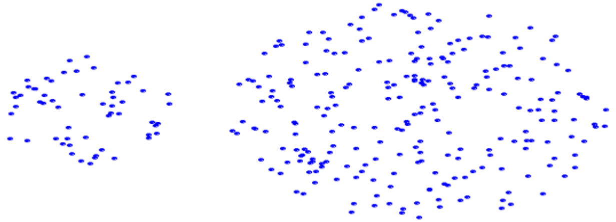
Consider all possible merges…



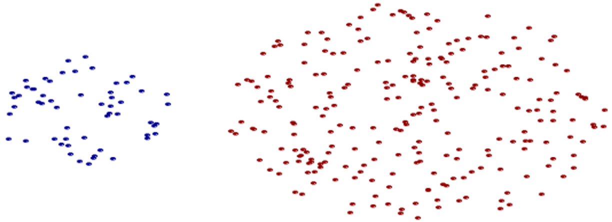
Choose the best

MIN: strengths and limitations

- Can handle non-elliptical shapes

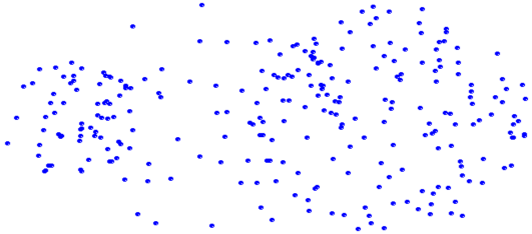


original points

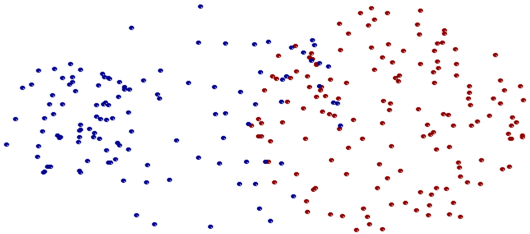


clusters

- Overlapping clusters and noise



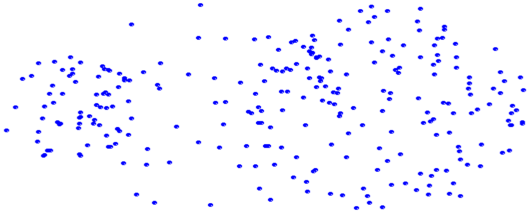
original points



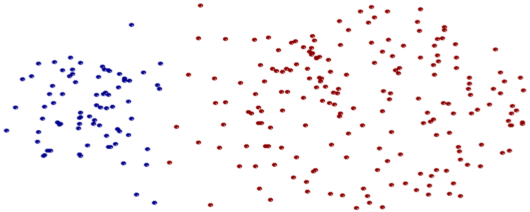
clusters

MAX: strengths and limitations

- Less susceptible to noise and outliers

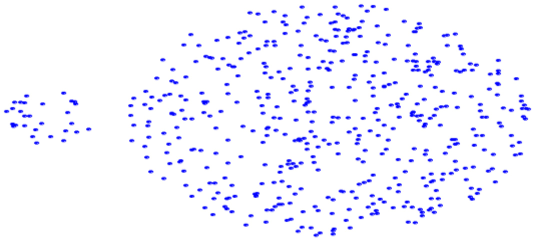


original points

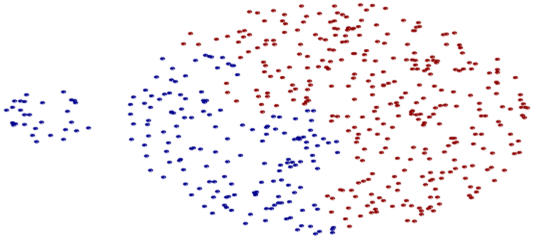


clusters

- Tends to break large clusters
- Biased towards globular clusters



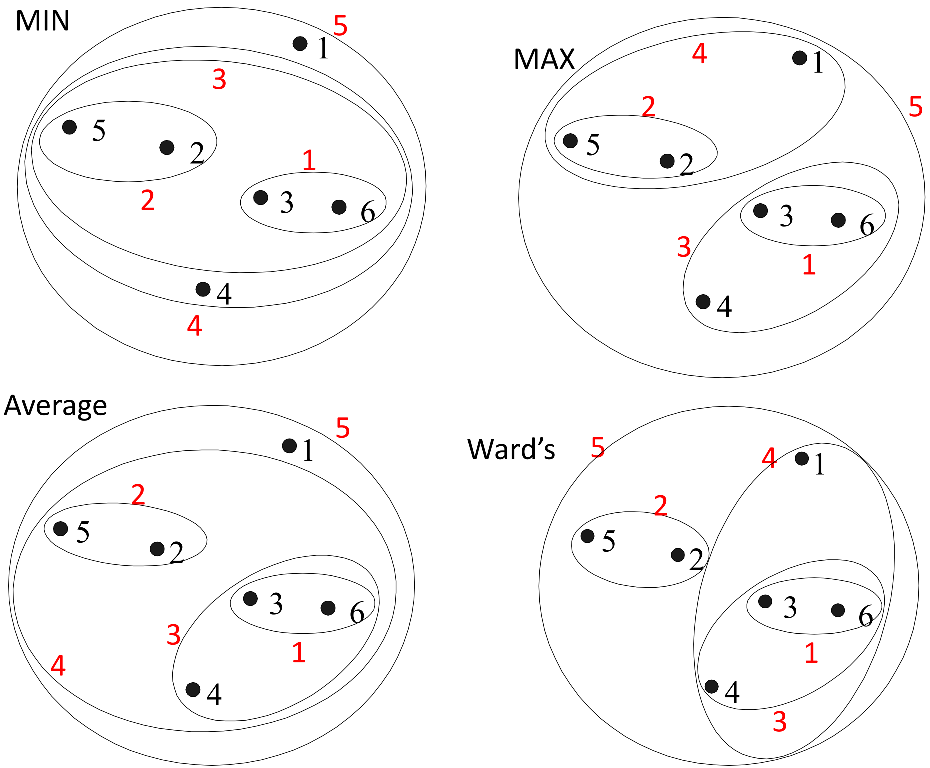
original points



clusters

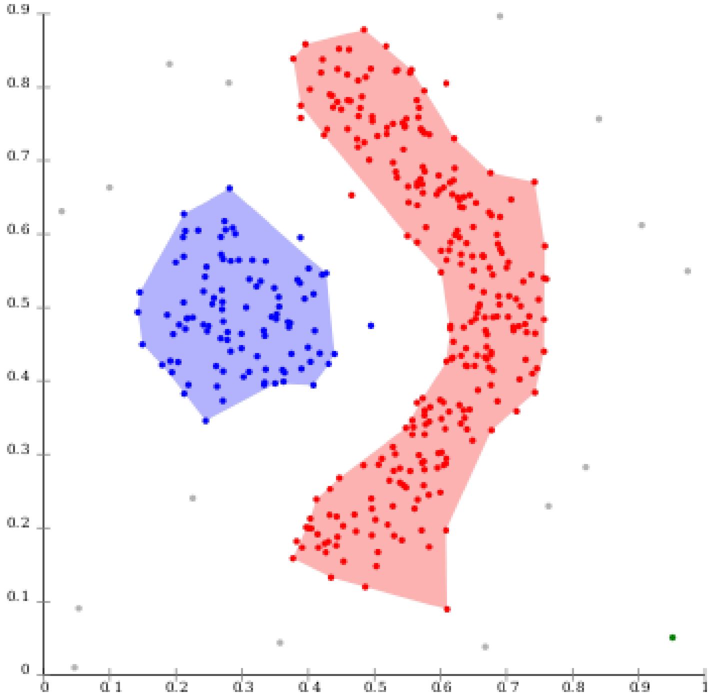
Hierachical clustering: comparison

- problems MIN and MAX link can be minimized under average/centroid/Ward link
  - *strength*: less susceptible to noise and outliers
  - *limitation*: biases towards globular clusters



DBSCAN (density-based clustering)

- clusters are defined as areas of higher density
- separation occurs in sparse areas
  - isolated data points here seen as outliers
- advantages? limitations?



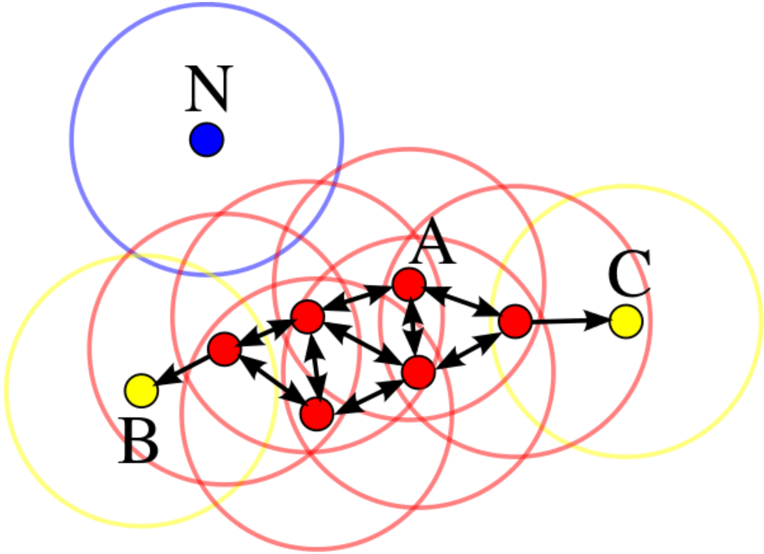
DBSCAN (density-based clustering)

- **parameters**
  - ε maximum distance
  - *p* minimum neighbors

- **algorithm**
  - for each point:

    cluster points with *p*

    neighbors at < ε distance



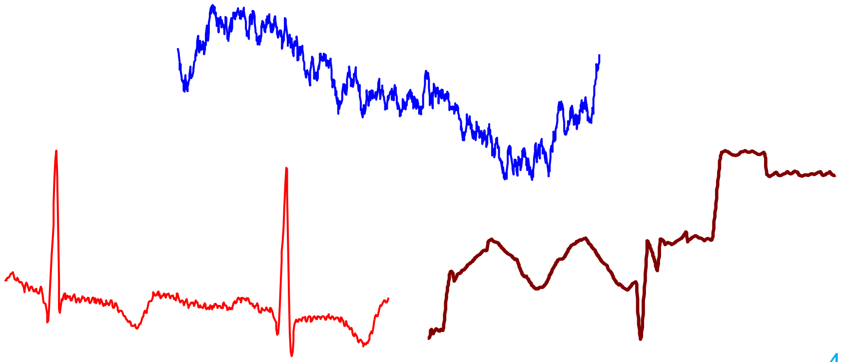
Outline

- Introduction to clustering
- Multivariate similarity metrics
- Approaches
  - hierarchical
  - density-based
- **From multivariate to complex data structures**
- Evaluation
  - intrinsic metrics
  - extrinsic metrics

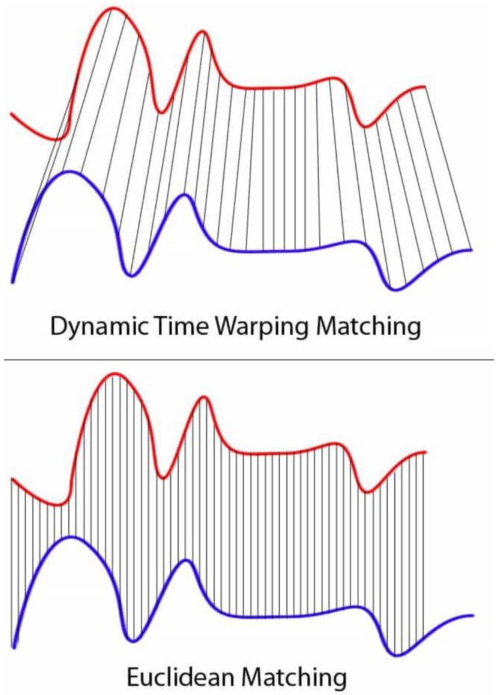
Time series data

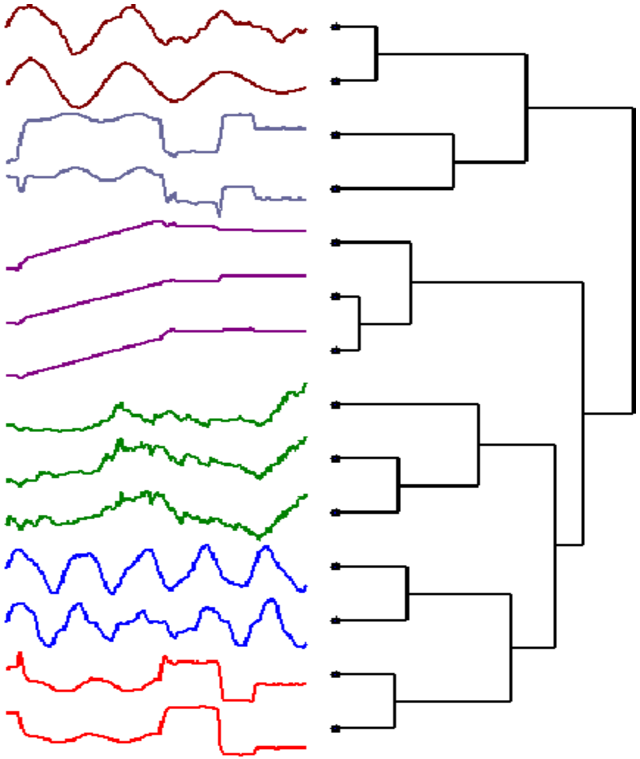
- **Time series**: sequence of values or symbols along time $\mathbf{s} =< \mathbf{x}_1, .., \mathbf{x}_T >$
  - *univariate* or *multivariate*, $\mathbf{x}_j \in \mathbb{R}^m$ (or $\mathbf{x}_j \in \{Y_1..Y_m\}$), where m is the multivariate order
- **Time series data**: $\{\mathbf{s}_1, .., \mathbf{s}_n\}$ where $\mathbf{s}_i$ is a time series
- Time series are *ubiquotous*:
  monitoring biological, individual, organizational, geophysical, digital, mechanical, societal systems
- Movement, image and video as time series, text data as time series
- People measure things...
  - *their blood pressure*
  - *the annual rainfall in New Zealand*
  - *the value of their Yahoo stock*
  - *the number of web hits per second*

  ... and things change over time



Time series clustering





Text document clustering

- Group related documents based on their content
  - the similarity between every string pair is calculated as a basis for determining the clusters
  - considering term vector spaces… cosine

thousands of terms

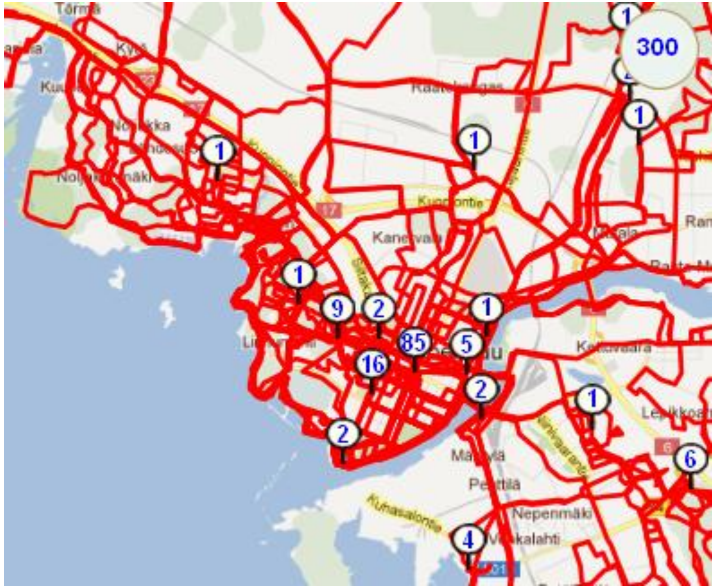
documents		T_1	T_2	………	T_m		class
	D_1	12	0	………	6		sports
	D_2	3	10	………	28		travel
	⋮					⋮	⋮
	D_n	0	11	………	16		jobs

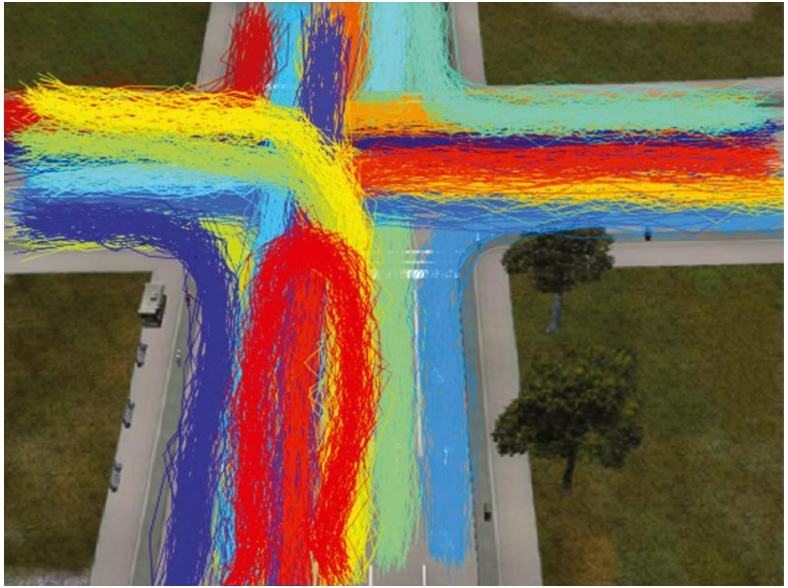
- A similarity measure is required to calculate the similarity between two strings

approximate string matching

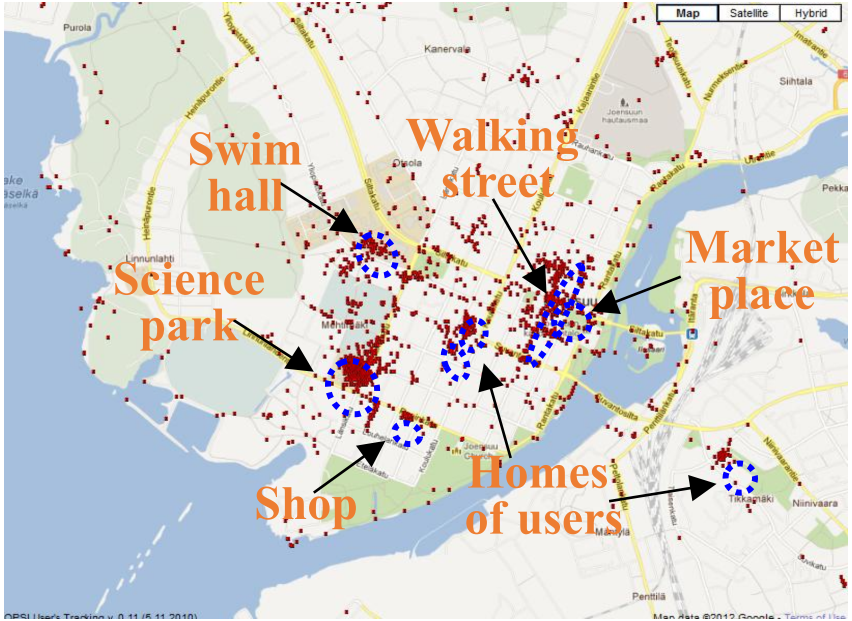
semantic similarity
stem, feature extraction and lexical analysis

GPS trajectory clustering





Spatial clustering



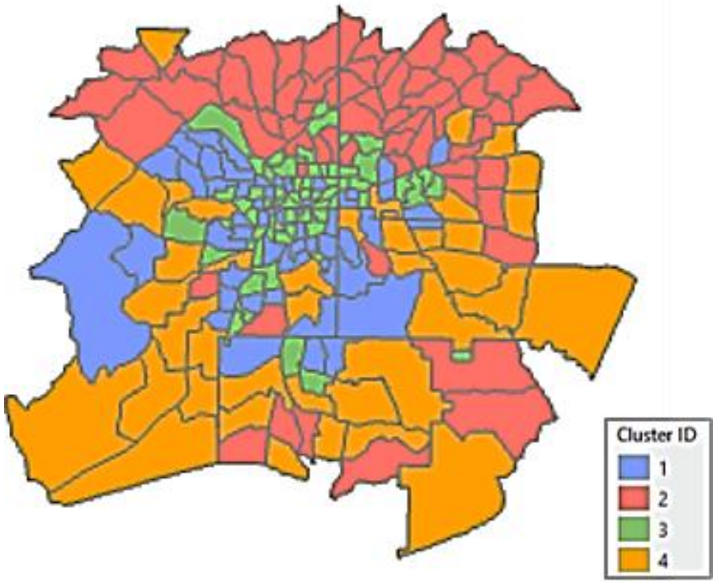
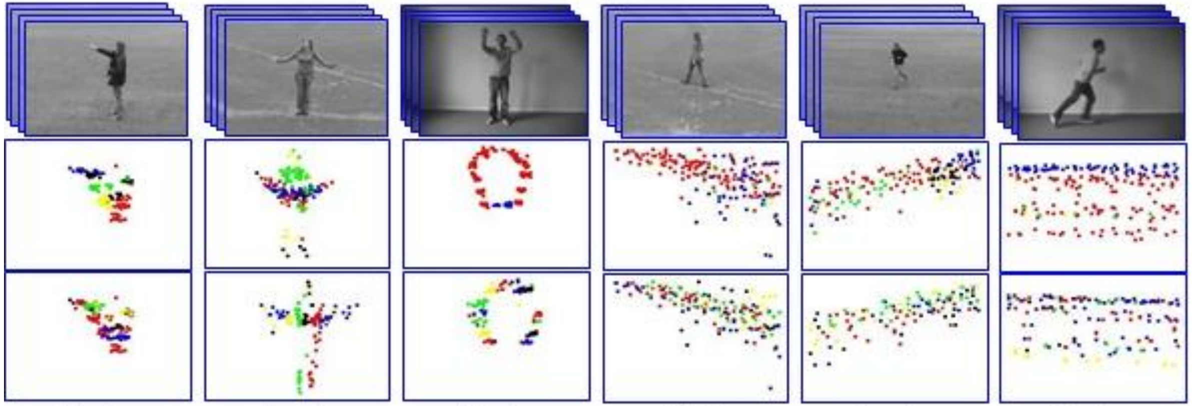


Image and video clustering

Image: Gansbeke et al.

Image: Liu and Shah



TÉCNICO+
FORMAÇÃO AVANÇADA

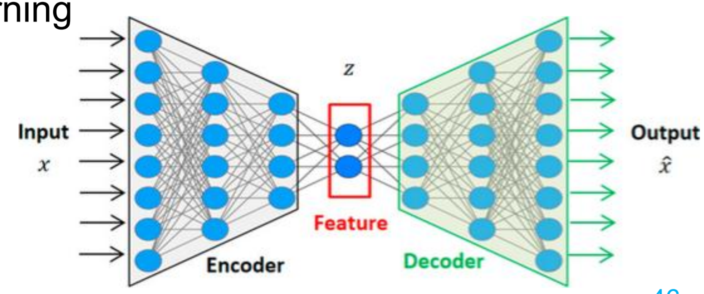
Representation learning (*next class*)

These and many other **complex data structures** may be encountered

- **video**, **events**, **tensors**, **heterogeneous** data structures…

Two major solutions

- **dedicated distances** or clustering **approaches**
- obtain (numeric) **representations** of these complex observations by **extracting features**
    - features can be extracted using **simple statistics**
        - e.g. extract centrality/variability/slope/max/min statistics on time series using sliding windows
    - ***embeddings*** can be extracted using representation learning
        - example: **auto-encoder neural networks** can be applied to deal with arbitrary complex inputs



Outline

- Introduction to clustering
- Multivariate similarity metrics
- Approaches
  - hierarchical
  - density-based
- From multivariate to complex data structures
- **Evaluation**
  - intrinsic metrics
  - extrinsic metrics

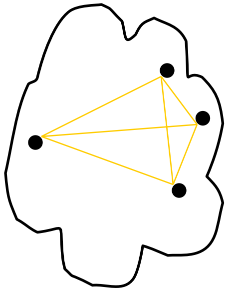
Evaluation: clustering quality

- 3 kinds of measures: external, internal and relative indexes
- **External** (supervised): extent to which cluster labels match true labels
  - requires prior or expert knowledge
- **Internal** (unsupervised): goodness without external information
  - how well they are separated (e.g. silhouette)
  - should be independent from algorithm-specific functions (unbiased)
- **Relative**: compare different cluster analyses (different parameters/algorithms)

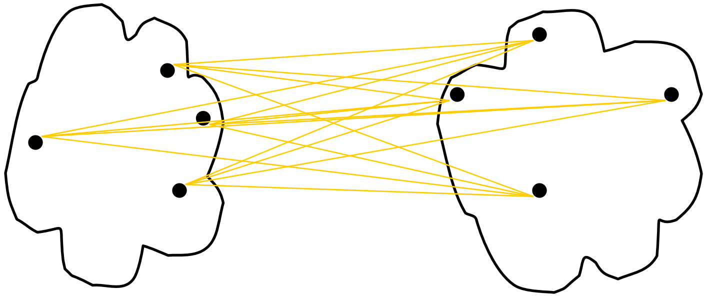
Internal measures: cohesion and separation

Proximity graph-based approach to measure cohesion and separation

- **Cohesion** is the sum of the weight of all links within a cluster
- **Separation** is the sum of the weights between nodes in the cluster and nodes outside the cluster



cohesion



separation

Internal measures: cohesion and separation

- **Cohesion** (e.g. *sum of squared errors* or sum of square within):

  how closely related are points in a cluster

$$SSE = SSW = \sum_{k=1}^{K} \sum_{x_i \in C_k} d(x_i, c_k)^2$$

- **Separation** (e.g. *sum of squares between* clusters)

  how distinct or well-separated a cluster is from other clusters

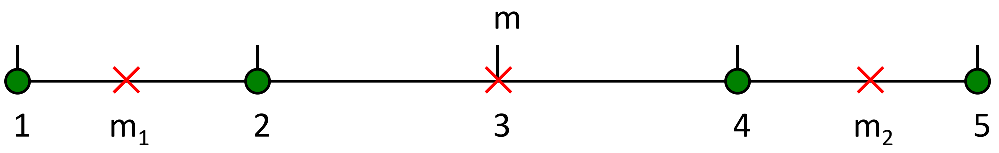
$$SSB = BSS = \sum_{k} |c_k| d(c_k, \bar{x})^2$$

- **Total error** (e.g. sum of squares): within and between errors $TSS = SSB + SSE$

$$TSS = \sum_{i}^{n} d(x_i, \bar{x})^2$$

Internal measures: cohesion and separation

SSB + SSE = constant



K=1 cluster:

$$SSE = (1-3)^2 + (2-3)^2 + (4-3)^2 + (5-3)^2 = 10$$
$$SSB = 4 \times (3-3)^2 = 0$$
$$Total = 10 + 0 = 10$$

K=2 clusters:

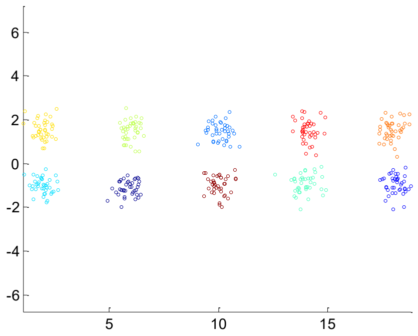
$$SSE = (1-1.5)^2 + (2-1.5)^2 + (4-4.5)^2 + (5-4.5)^2 = 1$$
$$SSB = 2 \times (3-1.5)^2 + 2 \times (4.5-3)^2 = 9$$
$$Total = 1 + 9 = 10$$

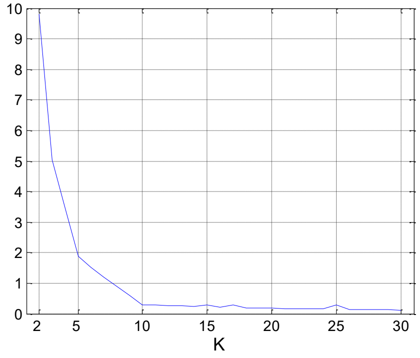
Internal measures: cohesion

- For each observation, the error is the distance to the nearest cluster
- Square these errors (to penalize larger distances) and sum these errors

$$SSE = \sum_{k=1}^{K} \sum_{x_i \in C_k} d(x_i, c_k)^2$$

- Good to compare two clustering solutions or two clusters
- Can also be used to estimate the number of clusters

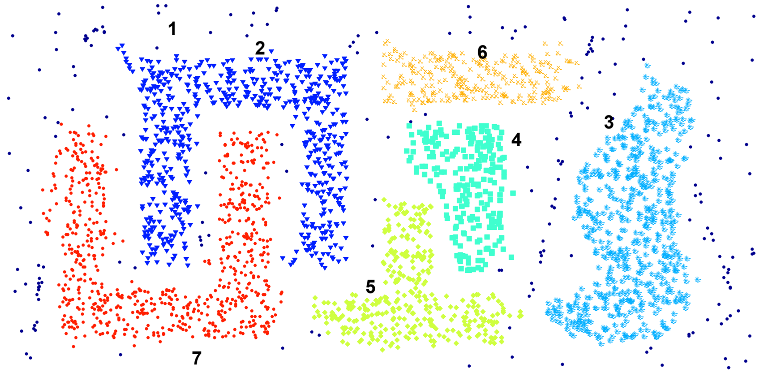


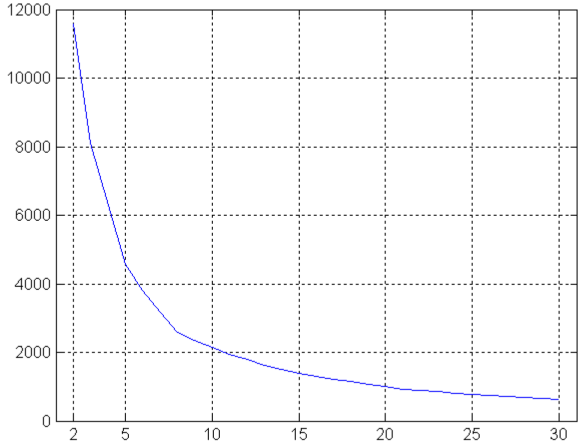


Internal measures: cohesion

Challenge on finding optimal #clusters:

- an easy way to reduce SSE is to increase the #clusters
- solution: elbow method (next class)



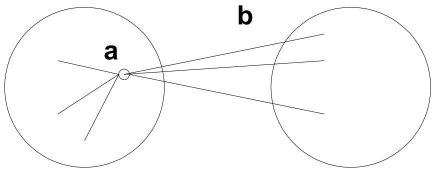


Internal measures: silhouette coefficient

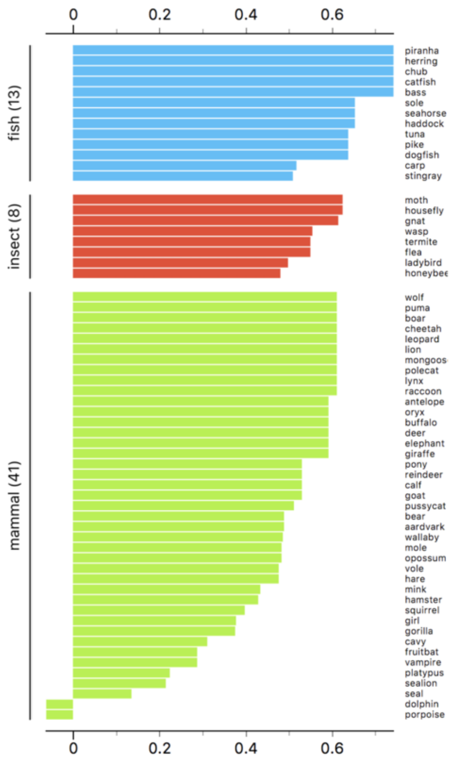
- Combine ideas of both *cohesion* and *separation*
- Calculated for a specific object $\mathbf{x}_i$
  - a = average distance of $\mathbf{x}_i$ to the points in its cluster
  - b = min (average distance of $\mathbf{x}_i$ to points in another cluster)
  - the silhouette coefficient for a point is then given by

    $s = 1 - a/b$ if $a < b$, (or $s = b/a - 1$ if $a \geq b$, not the usual case)

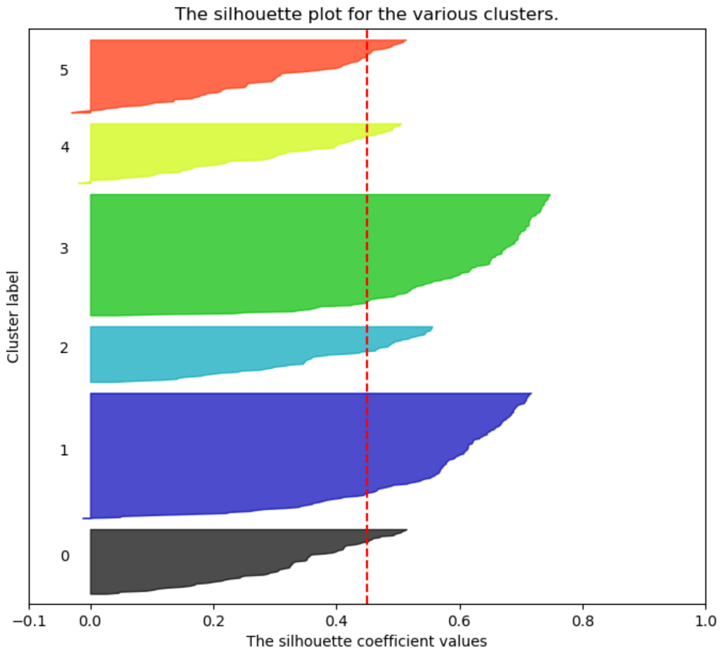
    between -1 and 1 (the closer to 1 the better)

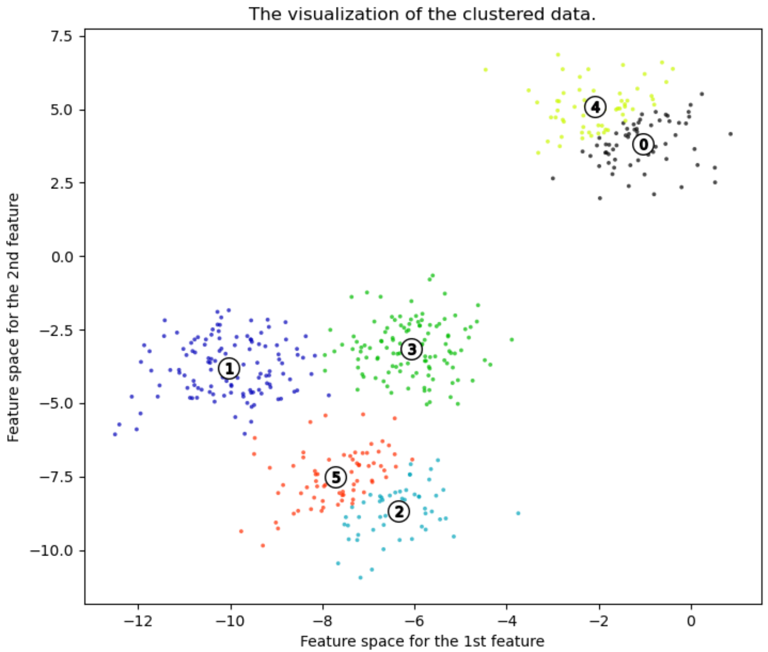


- Silhouette of cluster and clustering solution: average of silhouettes



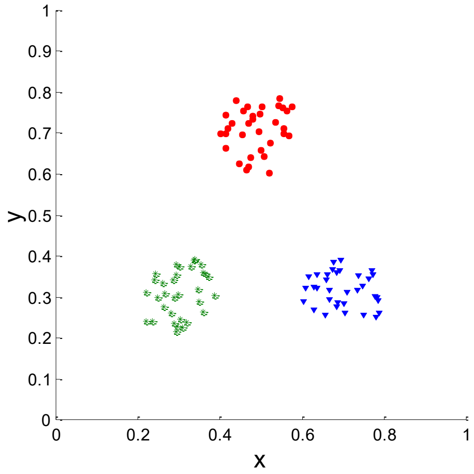
Internal measures: silhouette coefficient

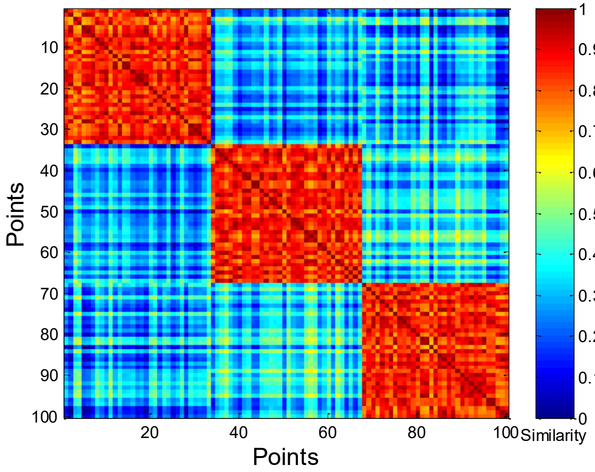




Internal measures: similarity matrix

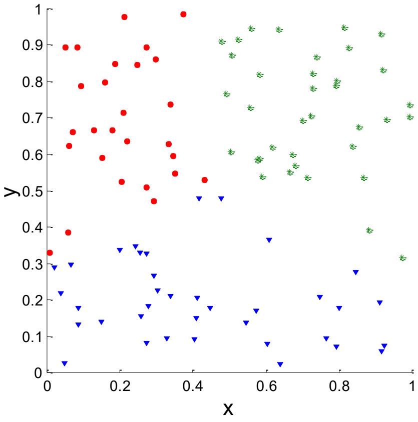
- Order the similarity matrix with respect to cluster labels and inspect visually



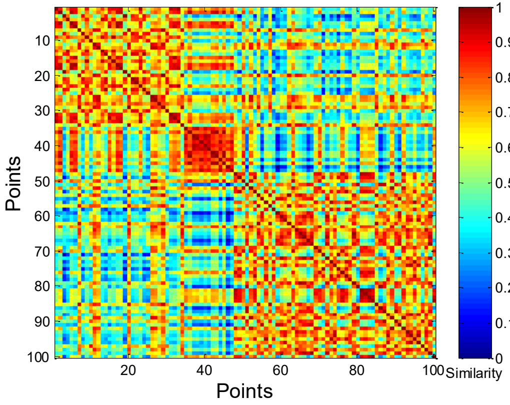


Internal measures: similarity matrix

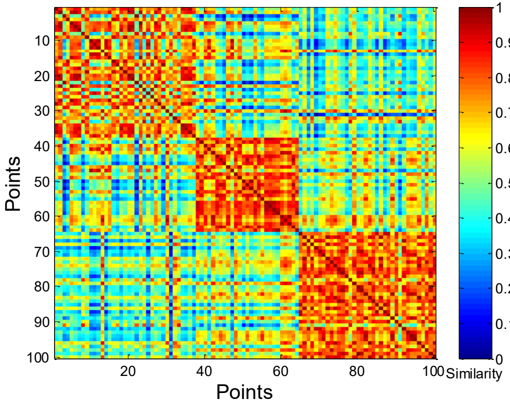
- Clusters in random data are not well-defined



complete link



k-means



Recall: clustering evaluation

- 3 kinds of measures: external, internal and relative indexes
- **External** (supervised): extent to which cluster labels match true labels
  - requires prior or expert knowledge
- **Internal** (unsupervised): goodness without external information
  - how well they are separated (e.g. silhouette)
  - should be independent from algorithm-specific functions (unbiased)
- **Relative**: compare different cluster analyses (different parameters/algorithms)

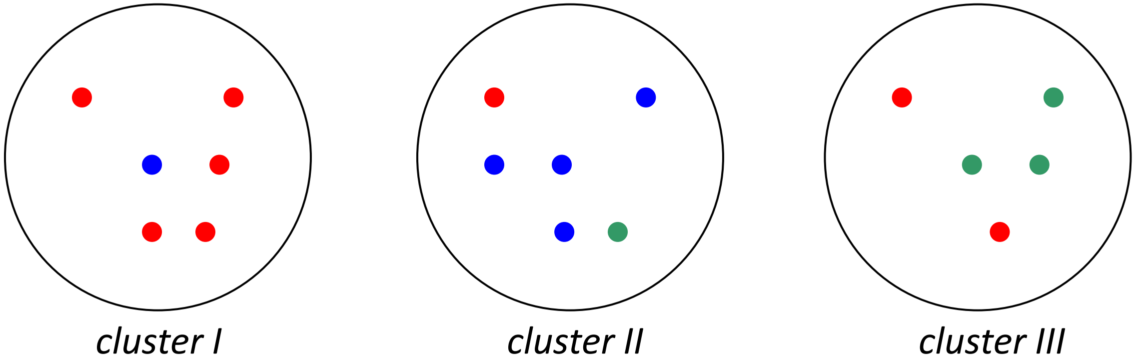
External measures: purity

- $\Omega = \{\omega_1, \omega_2, \ldots, \omega_K\}$ is the set of clusters
  $C = \{c_1, c_2, \ldots, c_J\}$ is the set of classes
- For each cluster ω_k: find class c_j with most objects in ω_k, n_{kj}
- Sum all n_{kj} and divide by total number of points

$$purity(\Omega, C) = \frac{1}{n} \sum_k \max_j |\omega_k \cap c_j|$$

- **Problem**: biased ⇒ *n* clusters maximizes purity
- Alternatives: entropy of classes in clusters

External measures: purity



cluster I: purity = 1/6 (max(5, 1, 0)) = 5/6

cluster II: purity = 1/6 (max(1, 4, 1)) = 4/6

cluster III: purity = 1/5 (max(2, 0, 3)) = 3/5

solution: purity = 1/17 (5+4+3) = 12/17

External measures: rand index

- Counts of object pairs

	same cluster	*different clusters*
same class	true positives (TP)	false negatives (FN)
different classes	false positives (FP)	true negatives (TN)

- **Rand index** $\mathrm{RI} = \frac{TP+TN}{TP+FP+FN+TN}$

- Given a specific cluster (*positive*):
  - precision = TP/(TP+FP)
  - recall = TP/(TP+FN)
  - F-measure = 2×precision×recall / (precision + recall)

External measures: rand index

Rand index?

Number of object pairs	Same cluster	Different clusters
Same class in ground truth	20	24
Different classes in ground truth	20	72

Thank you!

Rui Henriques

rmch@tecnico.ulisboa.pt