LABORATOIRE HUBERT CURIEN
UMR • CNRS • 5516 • SAINT-ETIENNE

UNIVERSITÉ DE LYON

From Statistics to Data Mining

Master 1
COlour in Science and Industry (COSI)
Cyber-Physical Social System (CPS2)
Saint-Étienne, France

Fabrice MUHLENBACH
https://perso.univ-st-etienne.fr/muhlfabr/
e-mail: fabrice.muhlenbach@univ-st-etienne.fr

Clustering — Basic concepts

- Introduction

➢ in clustering or unsupervised learning no training data, with class labeling, are available

➢ **Goal**:

- group the data into a number of sensible clusters (groups)
- this unravels similarities and differences among the available data

➢ **Applications**:

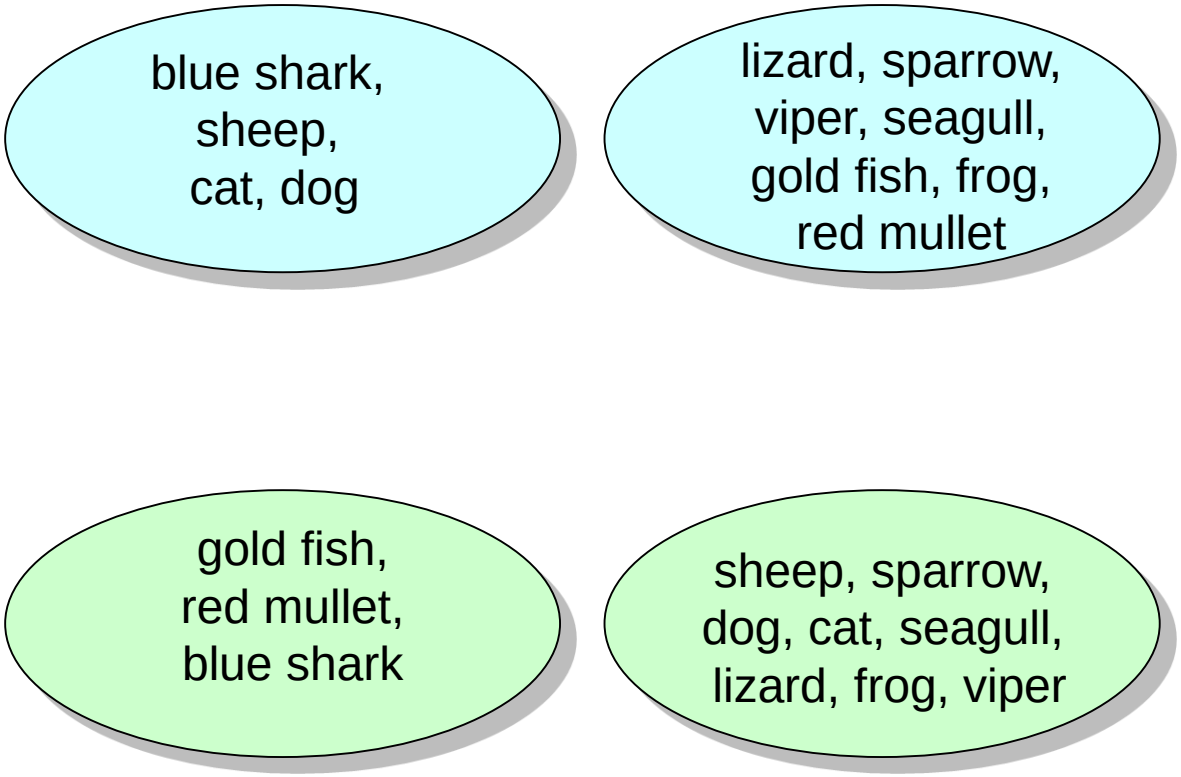
- engineering
- bioinformatics
- social sciences
- medicine
- data and web mining

→

- to perform clustering of a data set, a clustering criterion must first be adopted
- different clustering criteria lead, in general, to different clusters

Clustering — Basic concepts

- Example



1. Two clusters
2. Clustering criterion: How animals give birth to their progeny

1. Two clusters
2. Clustering criterion: Existence of lungs

Clustering — Basic concepts

- Supervised vs. Unsupervised Learning

➢ **Supervised learning (classification)**

- ✓ Supervision: The training data (observations, measurements, etc.) are accompanied by labels indicating the class of the observations
- ✓ New data is classified based on the training set

➢ **Unsupervised learning (clustering)**

- ✓ The class labels of training data is unknown
- ✓ Given a set of measurements, observations, etc. with the aim of establishing the existence of classes or clusters in the data

Clustering — Basic concepts

- Clustering task stages

1) Feature selection:

Features must be properly selected so as to encode as much information as possible concerning the task of interest. *Parsimony* and minimum information redundancy among the features is a major goal.

Preprocessing of features may be necessary prior to their utilization in subsequent stages.

- Prevent the curse of dimensionality
- Select the most important features
- Methods: outlier removal, data normalization, dealing with missing data, selection of the most independent features

Clustering — Basic concepts

- Clustering task stages

2) Proximity measure:
This is a measure that quantifies how “similar” or “dissimilar” two feature vectors are.
It is natural to ensure that all selected features contribute equally to the computation of the proximity measure and there are no features that dominate others.
3) Clustering criterion:
This depends on the interpretation the expert gives to the term “sensible,” based on the type of clusters that are expected to underlie the data set.
The clustering criterion may be expressed via a cost function or some other types of rules.

Clustering — Basic concepts

- Clustering task stages

4) Clustering algorithms:
Having adopted a proximity measure and a clustering criterion, this step refers to the choice of a specific algorithmic scheme that unravels the clustering structure of the data set.

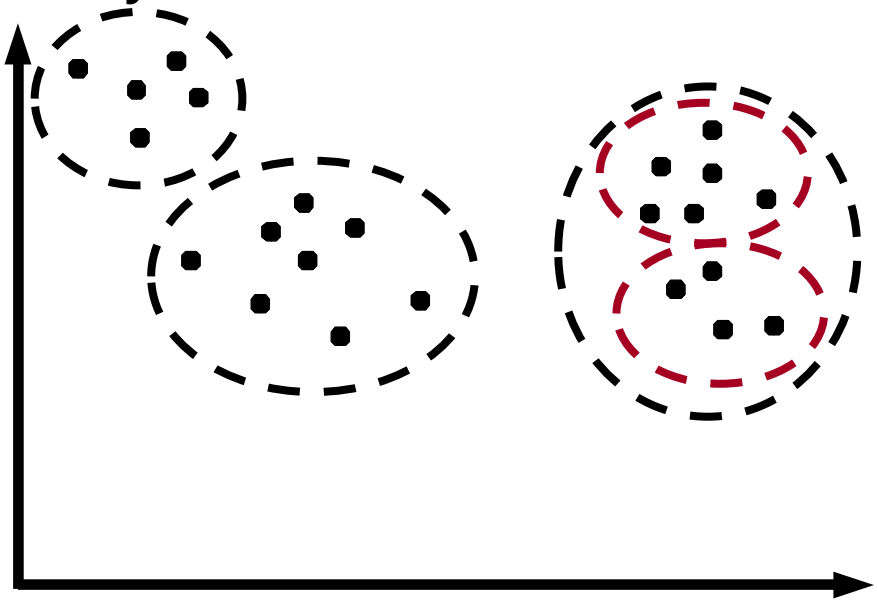
5) Validation of the results:
Once the results of the clustering algorithm have been obtained, we have to verify their correctness → use of appropriate tests.

6) Interpretation of the result:
In many cases, the expert in the application field must integrate the results of clustering with other experimental evidence and analysis in order to draw the right conclusions.

Clustering — Basic concepts

- Clustering task stages

6) Interpretation of the result (continued):
Depending on the similarity measure, the clustering criterion and the clustering algorithm different clusters may result.
Subjectivity is a reality to live with from now on



Clustering — Basic concepts

• Quality of a Clustering

- A good clustering method will produce high quality clusters with:
  - high intra-class similarity
  - low inter-class similarity
- The quality of a clustering result depends on both the similarity measure used by the method and its implementation.
- The quality of a clustering method is also measured by its ability to discover some or all of the hidden patterns.
- Once the results of the clustering algorithm have been obtained, we have to verify their correctness.
- This is usually carried out using appropriate tests.

Clustering — Basic concepts

- Applications of Cluster Analysis

➢ **Data reduction:**

Many times, the amount of the available data, N, is very large and, as a consequence, its processing becomes very demanding. Cluster analysis can be used in order to group the data into a number of "sensible" clusters, m ($<<N$), and to process each cluster as a single entity (= data compression).

➢ **Hypothesis generation:**

In this case we apply cluster analysis to a data set in order to infer some hypotheses concerning the nature of the data. Thus, clustering is used here as a vehicle to suggest hypotheses (→ must then be verified using other data sets).

Clustering — Basic concepts

- Applications of Cluster Analysis

➢ **Hypothesis testing (continued):**

In this context, cluster analysis is used for the verification of the validity of a specific hypothesis.

➢ **Prediction based on groups:**

In this case we apply cluster analysis to the available data set, and the resulting clusters are characterized based on the characteristics of the patterns by which they are formed.
In the sequel, if we are given an unknown pattern, we can determine the cluster to which it is more likely to belong and we characterize it based on the characterization of the respective cluster.

Clustering — Basic concepts

- Clustering definitions

➢ **Hard clustering:**

Each point belongs to a single cluster

- Let $X = \{\underline{x}_1, \underline{x}_2, ..., \underline{x}_N\}$
- An m-clustering R of X, is defined as the partition of X into m sets (clusters), $C_1, C_2, ..., C_m$, so that
  - $C_i \neq \emptyset, i = 1,2,...,m$
  - $\bigcup_{i=1}^{m} C_i = X$
  - $C_i \cap C_j = \emptyset, \ i \neq j, \ i, j = 1,2,...,m$

Data in C_i are more similar to each other and less similar to the data in the rest of the clusters → Quantifying the terms similar-dissimilar depends on the types of clusters that are expected.

Clustering — Basic concepts

- Clustering definitions

➢ **Fuzzy clustering:**

Each point belongs to all clusters up to some degree → a fuzzy clustering of X into m clusters is characterized by m functions

- $u_j : \underline{x} \rightarrow [0,1], \quad j = 1,2,\ldots,m$
- $\sum_{j=1}^{m} u_j(\underline{x}_i) = 1, \quad i = 1,2,\ldots,N$
- $0 < \sum_{i=1}^{N} u_j(\underline{x}_i) < N, \quad j = 1,2,\ldots,m$

These are known as “membership functions.”

Thus, each $\underline{x}_i$ belongs to any cluster “up to some degree”, depending on the value of $u_j(\underline{x}_i), \quad j = 1,2,\ldots,m$

If $u_j(\underline{x}_i)$ close to 1 then high grade of membership of $\underline{x}_i$ to cluster j and low grade of membership for $u_j(\underline{x}_i)$ close to 0.

Clustering — Basic concepts

- Major Clustering Approaches

- **Partitioning algorithms:**
  Construct various partitions and then evaluate them
- **Hierarchy algorithms:**
  Create a hierarchical decomposition of the set of data (or objects) using some criterion
- **Density-based:**
  Based on connectivity and density functions
- **Grid-based:**
  Based on a multiple-level granularity structure
- **Model-based:**
  A model is hypothesized for each of the clusters and the idea is to find the best fit of that model to each other

Clustering — Basic concepts

- Types of features

➢ With respect to their domain:
  - Continuous (the domain is a continuous subset of $\mathbb{R}$).
  - Discrete (the domain is a finite discrete set):
    Binary or dichotomous → two possible values

➢ With respect to the relative significance of the values they take
  - Nominal (the values code states, e.g., male/female).
  - Ordinal (the values are meaningfully ordered, e.g., poor, good, very good, excellent).
  - Interval-scaled (the difference of two values is meaningful but their ratio is meaningless, e.g., temperature).
  - Ratio-scaled (the ratio of two values is meaningful, e.g., weight).

Clustering — Basic concepts

- Proximity Measures

➢ **Clustering – First step:**
  - ❑ Feature selection → the features must be properly selected so as to encode as much information as possible concerning the task of interest.
  - ❑ Other data preprocessing tasks → data transformation: normalization (min-max, z-score)

➢ **Clustering – Second step:** Choice of a proximity measure (a similarity or a distance measure) to quantifies how “similar” or “dissimilar” two feature vectors are.

➢ We can calculate proximity (or distance) measures between: vectors, sets, or a vector and a set

Clustering — Basic concepts

- Proximity Measures between Vectors

Dissimilarity measure (between vectors of X) is a function $d: \Omega \times \Omega \in \mathbb{R}^2$ with the following properties:

- $\exists d_0 \in \mathbb{R}: -\infty < d_0 \leq d(\omega_1, \omega_2) < +\infty, \forall \omega_1, \omega_2 \in \mathbb{R}$
- $d(\omega_1, \omega_1) = d_0, \forall \omega_1 \in \Omega$
- $d(\omega_1, \omega_2) = d(\omega_2, \omega_1), \forall \omega_1, \omega_2 \in \Omega$

If in addition:

- $d(\omega_1, \omega_2) = 0$ iif $\omega_1 = \omega_2$
- $d(\omega_1, \omega_3) \leq d(\omega_1, \omega_2) + d(\omega_2, \omega_3), \quad \forall \omega_1, \omega_2, \omega_3 \in \Omega$
  (triangular inequality)

→ d is called a *metric* dissimilarity measure

Clustering — Basic concepts

- Proximity Measures between Vectors

Similarity measure (between vectors of X) is a function

$$s: \; X \times X \longrightarrow \mathbb{R}$$

with the following properties:

$$\exists s_0 \in \mathbb{R} \;\; -\infty < s(\underline{x}, \underline{y}) \leq s_0 < +\infty, \;\; \forall \underline{x}, \underline{y} \in X$$

$$s(\underline{x}, \underline{x}) = s_0, \;\; \forall \underline{x} \in X$$

$$s(\underline{x}, \underline{y}) = s(\underline{y}, \underline{x}), \;\; \forall \underline{x}, \underline{y} \in X$$

If in addition $s(\underline{x}, \underline{y}) = s_0 \;\; if \;\; and \;\; only \;\; if \;\; \underline{x} = \underline{y}$

$$s(\underline{x}, \underline{y}) s(\underline{y}, \underline{z}) \leq [s(\underline{x}, \underline{y}) + s(\underline{y}, \underline{z})] s(\underline{x}, \underline{z}), \;\; \forall \underline{x}, \underline{y}, \underline{z} \in X$$

s is called a *metric* similarity measure.

Clustering — Basic concepts

- Proximity Measures between Sets

Let a set of vector $D_i \subset X$, $i=1,...,k$ and $U=\{D_1,...,D_k\}$

A proximity measure $\wp$ on U is a function

$$\wp : U \times U \longrightarrow \mathbb{R}$$

A dissimilarity measure has to satisfy the relations of dissimilarity measure between vectors, where D_i are used in place of $\underline{x}$, $\underline{y}$
(similarly for similarity measures).

Clustering — Basic concepts

- Proximity Measures between Vectors

Real-valued vectors: Dissimilarity measures (DM)

Weighted l_p metric DM

$$d_p(\underline{x},\underline{y}) = \left(\sum_{i=1}^{l} w_i \mid x_i - y_i \mid^p\right)^{1/p}$$

Interesting instances are obtained for

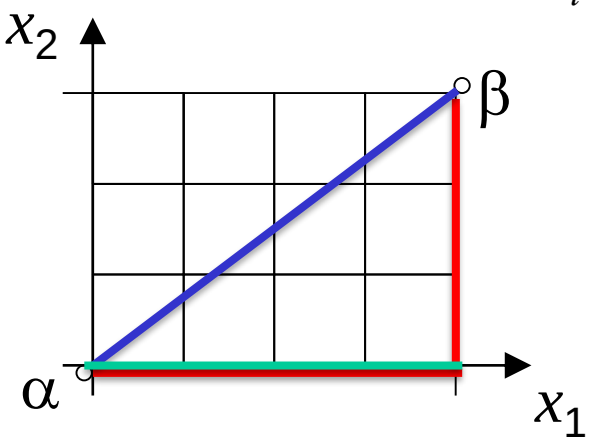
- p = 1 (weighted Manhattan norm)
- p = 2 (weighted Euclidean norm)
- $p = \infty$ ($d_\infty(x,y) = max_{1 \le i \le l}\, w_i \mid x_i - y_i \mid$)

Clustering — Basic concepts

- Proximity Measures between Vectors

Examples of dissimilarity measures (without weights: $w_i = 1$):

$$d_p(\underline{x}, \underline{y}) = \left(\sum_{i=1}^{l} w_i \mid x_i - y_i \mid^p\right)^{1/p}$$



- With $p = 1$ (Manhattan norm), $d(\alpha,\beta) = 7$
- With $p = 2$ (Euclidean norm), $d(\alpha,\beta) = 5$

- With $p = \infty$ ($d_\infty(x,y) = max_{1 \le i \le l} \mid x_i - y_i \mid$), $d(\alpha,\beta) = 4$

Clustering — Basic concepts

- Proximity Measures between Vectors

Other measures: *Mahalanobis distance*

- This measure is a kind of weighted Euclidean distance
- It produces distance contours of the same shape as a data distribution
- It is often more appropriate than Euclidean distance
- It is based on correlations between variables by which different patterns can be identified and analyzed

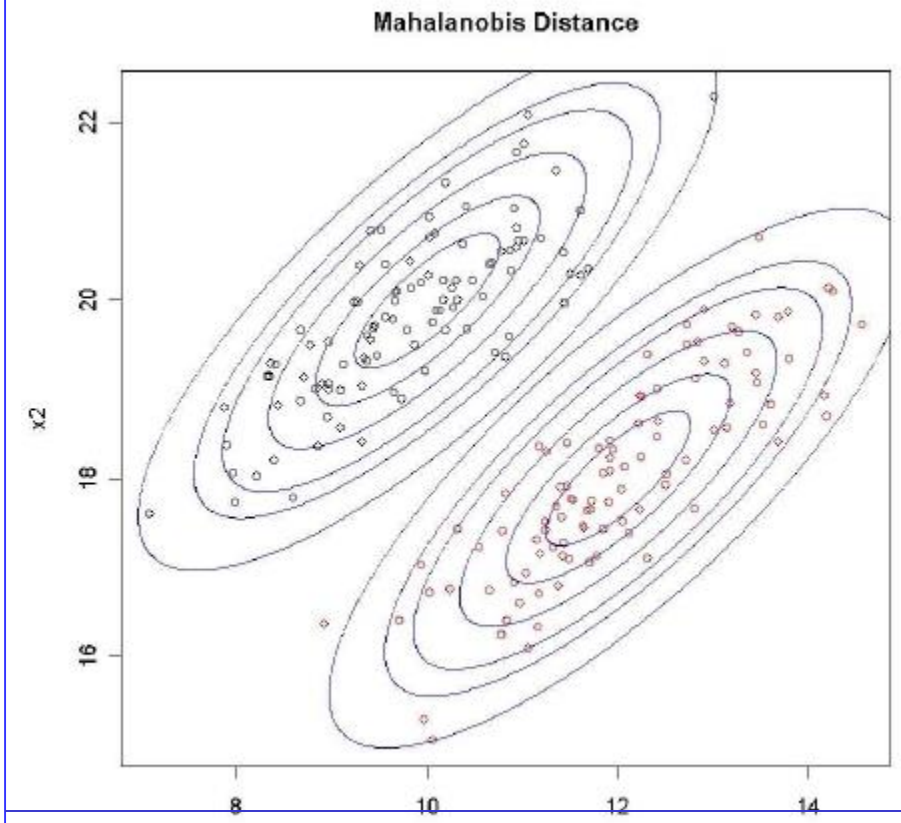
$$d_M(\underline{x}, \underline{y}) = \sqrt{(\underline{x} - \underline{y})^T S^{-1} (\underline{x} - \underline{y})}$$

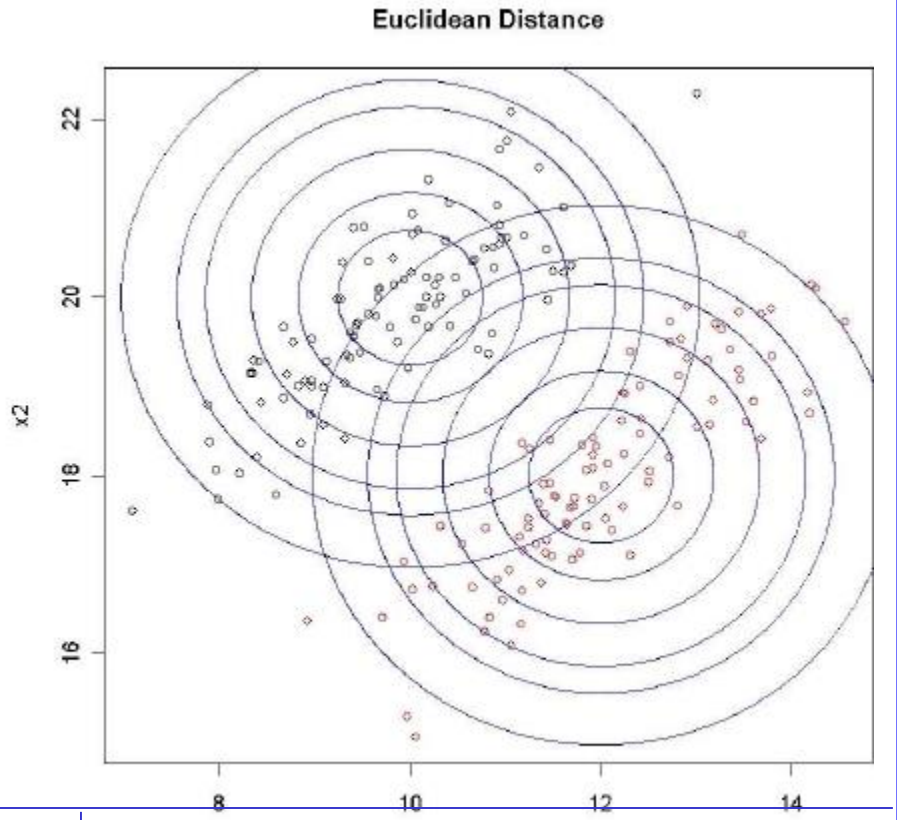
where S is the covariance matrix

Clustering — Basic concepts

- Proximity Measures between Vectors

Other measures: *Mahalanobis distance*





Clustering — Basic concepts

- Proximity Measures between Vectors

Real-valued vectors: Similarity measures

➢ Inner product

$$s_{inner}(\underline{x},\underline{y}) = \underline{x}^T \underline{y} = \sum_{i=1}^{l} x_i y_i$$

➢ Tanimoto measure

$$s_T(\underline{x},\underline{y}) = \frac{\underline{x}^T \underline{y}}{\|\underline{x}\|^2 + \|\underline{y}\|^2 - \underline{x}^T \underline{y}}$$

$$s_T(\underline{x},\underline{y}) = 1 - \frac{d_2(\underline{x},\underline{y})}{\|\underline{x}\| + \|\underline{y}\|}$$

Clustering — Basic concepts

- Proximity Measures between Vectors

Discrete-valued vectors: Distance and Similarity measures

Let $F = \{0, 1, \ldots, k-1\}$ be a set of symbols and $X = \{x_1, \ldots, x_N\} \subset F^l$
Let $A(x,y) = [a_{ij}]$, $i, j = 0, 1, \ldots, k-1$, where a_{ij} is the number of places where $\underline{x}$ has the i-th symbol and $\underline{y}$ has the j-th symbol.

The Hamming distance (number of places where $\underline{x}$ and $\underline{y}$ differ)

$$d_H(\underline{x}, \underline{y}) = \sum_{i=0}^{k-1} \sum_{\substack{j=0 \\ j \neq i}}^{k-1} a_{ij}$$

The $\underline{l_1}$ distance $d_1(\underline{x}, \underline{y}) = \sum_{i=1}^{l} | x_i - y_i |$

Clustering — Basic concepts

- Proximity Measures between Vectors

Binary value vectors: Similarity and Distance measures

A contingency table for binary data:

		y		
		1	0	*sum*
x	1	a	b	$a+b$
	0	c	d	$c+d$
	sum	$a+c$	$b+d$	p

- Simple matching coefficient (invariant, if the binary variable is symmetric): $d_{SMC}(\underline{x},\underline{y}) = \frac{b+c}{p}$ $\quad s_{SMC}(\underline{x},\underline{y}) = \frac{a+d}{p}$
- Jaccard coefficient (non-invariant if the binary variable is asymmetric): $d_J(\underline{x},\underline{y}) = \frac{b+c}{a+b+c}$ $\quad s_J(\underline{x},\underline{y}) = \frac{a}{a+b+c}$

Clustering — Basic concepts

- Proximity Measures between a Vector and a Set

➢ Let $X=\{\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_N\}$ and $C \subset X$, $\underline{x} \in X$

➢ All points of C contribute to the definition of $\wp(\underline{x}, C)$

❑ Max proximity function

$$\wp^{ps}_{\max}(\underline{x}, C) = \max_{\underline{y} \in C} \wp(\underline{x}, \underline{y})$$

❑ Min proximity function

$$\wp^{ps}_{\min}(\underline{x}, C) = \min_{\underline{y} \in C} \wp(\underline{x}, \underline{y})$$

❑ Average proximity function

$$\wp^{ps}_{avg}(\underline{x}, C) = \frac{1}{n_C} \sum_{\underline{y} \in C} \wp(\underline{x}, \underline{y})$$

Clustering — Basic concepts

- Proximity Measures between a Vector and a Set

- A representative of C, r_C, contributes to the definition of $\wp(\underline{x},C)$
- In this case: $\wp(\underline{x},C)=\wp(\underline{x},\underline{r}_C)$
- Typical representatives are:
  - The mean vector: $\underline{m}_p = \left(1/n_C\right)\sum_{\underline{y}\in C}\underline{y}$ (n_C is the cardinality of C)
  - The mean center: $\underline{m}_C \in C: \sum_{\underline{y}\in C} d(\underline{m}_C,\underline{y}) \le \sum_{\underline{y}\in C} d(\underline{z},\underline{y}),\ \forall \underline{z}\in C$
  - The median center: $\underline{m}_{med} \in C: \ med(d(\underline{m}_{med},\underline{y}) \mid \underline{y}\in C) \le med(d(\underline{z},\underline{y}) \mid \underline{y}\in C),\ \forall \underline{z}\in C$
- Note: Other representatives (e.g., hyperplanes, hyperspheres) are useful in certain applications (e.g., object identification using clustering techniques).

Clustering — Basic concepts

- Proximity Measures between Sets

➢ Let $X=\{\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_N\}$, $D_i, D_j \subset X$ and $n_i=|D_i|$, $n_j=|D_j|$

➢ All points of each set contribute to $\wp(D_i,D_j)$

❑ Max proximity function (measure but not metric, only if $\wp$ is a similarity measure)

$$\wp^{ss}_{\max}(D_i, D_j) = \max_{\underline{x}\in D_i, \underline{y}\in D_j} \wp(\underline{x}, \underline{y})$$

❑ Min proximity function (measure but not metric, only if $\wp$ is a dissimilarity measure)

$$\wp^{ss}_{\min}(D_i, D_j) = \min_{\underline{x}\in D_i, \underline{y}\in D_j} \wp(\underline{x}, \underline{y})$$

❑ Average proximity function (not a measure, even if $\wp$ is a measure) $\wp^{ss}_{avg}(D_i, D_j) = \left(1/n_i n_j\right) \sum_{x\in D_i} \sum_{x\in D_j} \wp(\underline{x}, \underline{y})$

Clustering — Basic concepts

- Proximity Measures between Sets

➢ Let Each set D_i is represented by its representative vector $\underline{m}_i$

❑ Mean proximity function (it is a measure provided that $\wp$ is a measure): $\wp^{ss}_{mean}(D_i, D_j) = \wp(\underline{m}_i, \underline{m}_j)$

❑ $\wp^{ss}_{e}(D_i, D_j) = \sqrt{\frac{n_i n_j}{n_i + n_j}} \wp(\underline{m}_i, \underline{m}_j)$

➢ Some remarks

❑ different choices of proximity functions between sets
→ totally different clustering results

❑ different proximity measures between vectors in the same proximity function between sets
→ different clustering results

Clustering — Algorithms

- Introduction

Number of possible clusterings

Let $X=\{\underline{x}_1,\underline{x}_2,\ldots,\underline{x}_N\}$.

Question:

→ In how many ways the N points can be assigned into m groups?

Answer:

→ $$S(N,m)=\frac{1}{m!}\sum_{i=0}^{m}(-1)^{m-1}\binom{m}{i}i^N$$

Examples:

$S(15,3)$ = 2,375,101

$S(20,4)$=45,232,115,901

$S(100,5)=10^{68}$!

Clustering — Algorithms

- Basic questions

A way out:

Consider only a small fraction of clusterings of X and select a “sensible” clustering among them

Question 1: Which fraction of clusterings is considered?

Question 2: What “sensible” means?

The answer depends on the specific clustering algorithm and the specific criteria to be adopted.

Clustering — Algorithms

- Major categories

- **Sequential:** A single clustering is produced. One or few sequential passes on the data.
- **Hierarchical:** A sequence of (nested) clusterings is produced.
  - ✓ Agglomerative: Matrix theory, Graph theory
  - ✓ Divisive
  - ✓ Combinations of the above (e.g., Chameleon)
- **Cost function optimization:** For most of the cases a single clustering is obtained.
  - ✓ Hard clustering: k-means, k-medoids algorithms, etc.
  - ✓ Fuzzy clustering
  - ✓ Possibilistic clustering (based on the possibility of a point to belong to a cluster)

Clustering — Algorithms

- Major categories

➢ **Other schemes**

- Algorithms based on graph theory (e.g., Minimum Spanning Tree, regions of influence, directed trees).
- Competitive learning algorithms (basic competitive learning scheme, Kohonen self organizing maps).
- Subspace clustering algorithms.
- Binary morphology clustering algorithms.

Sequential Clustering Algorithms

- Sequential

➢ **Basic idea:** A single clustering is produced. One or few sequential passes on the data.

➢ The common traits shared by these algorithms are:

✓ One or very few passes on the data are required.

✓ The number of clusters is not known a-priori, except (possibly) an upper bound, q.

✓ The clusters are defined with the aid of:

❑ An appropriately defined distance $d(\underline{x},C)$ of a point from a cluster.

❑ A threshold Θ associated with the distance.

Sequential Clustering Algorithms

• Basic Sequential Algorithm Scheme (BSAS)

- $m = 1$ \{number of clusters\}\
- $C_m = \{\underline{x}_1\}$
- For $i = 2$ to N
  - Find C_k: $d(\underline{x}_i, C_k) = min_{1 \le j \le m} d(\underline{x}_i, C_j)$
  - If $(d(\underline{x}_i, C_k) > \Theta)$ *AND* $(m < q)$ then
    - $m = m + 1$
    - $C_m = \{\underline{x}_i\}$
  - Else
    - $C_k = C_k \cup \{\underline{x}_i\}$
    - Where necessary, update representatives (*)
  - End {if}
- End {for}

(*) When the mean vector $\underline{m}_C$ is used as representative of the cluster C with n_c elements, the updating in the light of a new vector $\underline{x}$ becomes

$$\underline{m}_C^{new} = (n_C \, \underline{m}_C + \underline{x}) / (n_C + 1)$$

Sequential Clustering Algorithms

- Basic Sequential Algorithm Scheme (BSAS)

➢ **Remarks**:

- The order of presentation of the data in the algorithm plays important role in the clustering results. Different orders of presentation may lead to totally different clustering results, in terms of the number of clusters as well as the clusters themselves.
- In BSAS the decision for a vector ***x*** is reached prior to the final cluster formation.
- BSAS perform a single pass on the data. Its complexity is $O(N)$
- If clusters are represented by point representatives, compact clusters are favored.

Sequential Clustering Algorithms

- Other sequential clustering algorithms

➢ **MBSAS**, a modification of BSAS:

- In MBAS, a decision for a vector ***x*** during the pattern classification phase is reached taking into account all clusters.
- A cluster determination phase (first pass on the data), which is the same as BSAS with the exception that no vector is assigned to an already formed cluster. At the end of this phase, each cluster consists of a single element
- A pattern classification phase (second pass on the data), where each one of the unassigned vector is assigned to its closest cluster

➢ **Maxmin algorithm**

➢ **Two-threshold sequential scheme**

Hierarchical Clustering Algorithms

- Introduction

➢ Hierarchical clustering algorithms produce a hierarchy of (hard) clusterings instead of a single clustering.

➢ Applications in:
- social sciences
- biological taxonomy
- modern biology
- medicine
- archaeology
- computer science and engineering...

Hierarchical Clustering Algorithms

- Principle

Let $X = \{x_1,\ldots,x_N\}$, $x_i=[x_{i1},\ldots,x_{il}]^T$. Recall that in hard clustering each vector belongs exclusively to a single cluster.
An m-(hard) clustering of X, $\mathcal{R}$, is a partition of X into m sets (clusters) $C_1,\ldots,C_m$, so that: $C_i \neq \emptyset, i = 1,2,...,m$

$$\bigcup_{i=1}^{m} C_i = X$$

$$C_i \cap C_j = \emptyset, \quad i \neq j, \quad i, j = 1,2,\ldots, m$$

By the definition: $\mathcal{R} = \{C_j, j=1,...m\}$

- **Definition:** A clustering $\mathcal{R}_1$ containing k clusters is said to be nested in the clustering $\mathcal{R}_2$ containing r ($<k$) clusters, if each cluster in $\mathcal{R}_1$ is a subset of a cluster in $\mathcal{R}_2$.
- We write $\mathcal{R}_1 \angle \mathcal{R}_2$

Hierarchical Clustering Algorithms

- Principle

➢ Example: Let $\mathcal{R}_1=\{\{\underline{x}_1,\underline{x}_3\},\{\underline{x}_4\},\{\underline{x}_2,\underline{x}_5\}\}$, $\mathcal{R}_2=\{\{\underline{x}_1,\underline{x}_3,\underline{x}_4\},\{\underline{x}_2,\underline{x}_5\}\}$,
$\mathcal{R}_3=\{\{\underline{x}_1,\underline{x}_4\},\{\underline{x}_3\},\{\underline{x}_2,\underline{x}_5\}\}$, $\mathcal{R}_4=\{\{\underline{x}_1,\underline{x}_2,\underline{x}_4\},\{\underline{x}_3,\underline{x}_5\}\}$.
It is $\mathcal{R}_1 \angle \mathcal{R}_2$, but not $\mathcal{R}_1 \angle \mathcal{R}_3$, $\mathcal{R}_1 \angle \mathcal{R}_4$, $\mathcal{R}_1 \angle \mathcal{R}_1$.

➢ Remarks:

- Hierarchical clustering algorithms produce a hierarchy of nested clusterings.
- They involve N steps at the most
- At each step t, the clustering $\mathcal{R}_t$ is produced by $\mathcal{R}_{t\text{-}1}$.

➢ Main categories:

- Agglomerative clustering algorithms: Here $\mathcal{R}_0=\{\{\underline{x}_1\},\ldots,\{\underline{x}_N\}\}$, $\mathcal{R}_{N\text{-}1}=\{\{\underline{x}_1,\ldots,\underline{x}_N\}\}$ and $\mathcal{R}_0 \angle \ldots \angle \mathcal{R}_{N\text{-}1}$.
- Divisive clustering algorithms: Here $\mathcal{R}_0=\{\{\underline{x}_1,\ldots,\underline{x}_N\}\}$, $\mathcal{R}_{N\text{-}1}=\{\{\underline{x}_1\},\ldots,\{\underline{x}_N\}\}$ and $\mathcal{R}_{N\text{-}1} \angle \ldots \angle \mathcal{R}_0$.

Hierarchical Clustering Algorithms

- Agglomerative Algorithms (based on matrix theory)

- Let $g(C_i,C_j)$ a proximity function between two clusters of X.
- Generalized Agglomerative Scheme (GAS)
  - Initialization
    - Choose $\mathcal{R}_0=\{\{\underline{x}_1\},\ldots,\{\underline{x}_N\}\}$
    - $t=0$
  - Repeat
    - $t = t + 1$
    - Choose (C_i,C_j) in $\mathcal{R}_{t-1}$ such that
$$g(C_i,C_j)=\begin{cases}\min_{r,s} g(C_r,C_s), & \text{if } g \text{ is a disim. function}\\ \max_{r,s} g(C_r,C_s), & \text{if } g \text{ is a sim. function}\end{cases}$$
    - Define $C_q=C_i\cup C_j$ and produce $\mathcal{R}_t=(\mathcal{R}_{t-1}-\{C_i,C_j\})\cup\{C_q\}$
  - Until all vectors lie in a single cluster.

Hierarchical Clustering Algorithms

- Agglomerative Algorithms

➢ **Remarks**:

- If two vectors come together into a single cluster at level t of the hierarchy, they will remain in the same cluster for all subsequent clusterings.
- As a consequence, there is no way to recover a “poor” clustering that may have occurred in an earlier level of hierarchy.
- Number of operations: O(N^3)

Hierarchical Clustering Algorithms

- Agglomerative Algorithms

➢ **Definitions of some useful quantities**

Let $X=\{\underline{x}_1,\underline{x}_2,\ldots,\underline{x}_N\}$, with $\underline{x}_i=[x_{i1},x_{i2},\ldots,x_{il}]^T$.

– Pattern matrix ($D(X)$):
An Nxl matrix whose i-th row is $\underline{x}_i$ (transposed).

– Proximity (similarity or dissimilarity) matrix ($P(X)$):
An NxN matrix whose (i,j) element equals the proximity $\wp(\underline{x}_i,\underline{x}_j)$ (similarity $s(\underline{x}_i,\underline{x}_j)$, dissimilarity $d(\underline{x}_i,\underline{x}_j)$).

Hierarchical Clustering Algorithms

- Agglomerative Algorithms

Example

Let $X = \{\underline{x}_1, \underline{x}_2, \underline{x}_3, \underline{x}_4, \underline{x}_5\}$,

with $\underline{x}_1 = [1, 1]^T$, $\underline{x}_2 = [2, 1]^T$, $\underline{x}_3 = [5, 4]^T$, $\underline{x}_4 = [6, 5]^T$, $\underline{x}_5 = [6.5, 6]^T$.

$$D(X) = \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 5 & 4 \\ 6 & 5 \\ 6.5 & 6 \end{bmatrix}$$

Euclidean distance

$$P(X) = \begin{bmatrix} 0 & 1 & 5 & 6.4 & 7.4 \\ 1 & 0 & 4.2 & 5.7 & 6.7 \\ 5 & 4.2 & 0 & 1.4 & 2.5 \\ 6.4 & 5.7 & 1.4 & 0 & 1.1 \\ 7.4 & 6.7 & 2.5 & 1.1 & 0 \end{bmatrix}$$

Tanimoto distance

$$P'(X) = \begin{bmatrix} 1 & 0.75 & 0.26 & 0.21 & 0.18 \\ 0.75 & 1 & 0.44 & 0.35 & 0.20 \\ 0.26 & 0.44 & 1 & 0.96 & 0.90 \\ 0.21 & 0.35 & 0.96 & 1 & 0.98 \\ 0.18 & 0.20 & 0.90 & 0.98 & 1 \end{bmatrix}$$

Hierarchical Clustering Algorithms

- Agglomerative Algorithms

- Threshold dendrogram (or dendrorgram) → it is an effective way of representing the sequence of clusterings which are produced by an agglomerative algorithm.

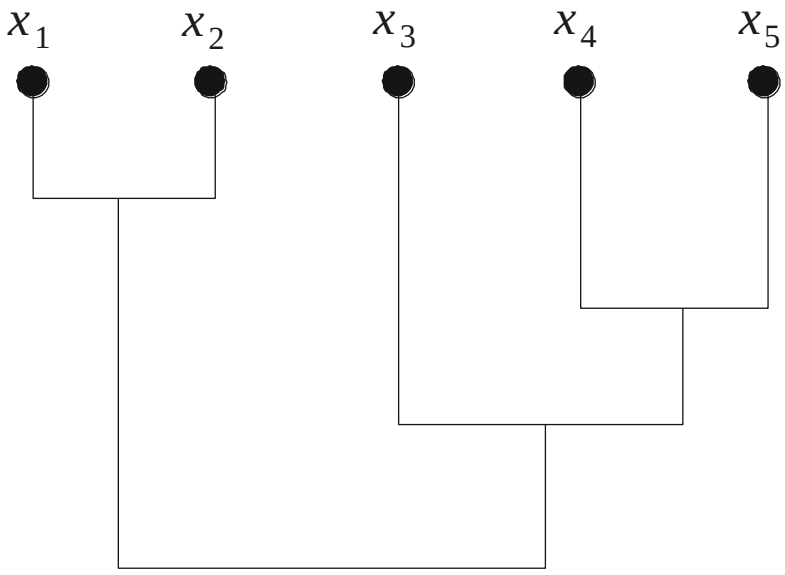
$\{\{x_1\},\{x_2\},\{x_3\},\{x_4\},\{x_5\}\}$

$\{\{x_1,x_2\},\{x_3\},\{x_4\},\{x_5\}\}$

$\{\{x_1,x_2\},\{x_3\},\{x_4,x_5\}\}$

$\{\{x_1,x_2\},\{x_3,x_4,x_5\}\}$

$\{\{x_1,x_2,x_3,x_4,x_5\}\}$



Hierarchical Clustering Algorithms

- Agglomerative Algorithms

➢ Distance functions:

○ A number of distance functions comply with the following update equation:

$$d(C_q,C_s)=a_i d(C_i,C_s)+a_j(d(C_j,C_s)+bd(C_i,C_j)+c|d(C_i,C_s)-d(C_j,C_s)|$$

○ Algorithms that follow the above equation are:

❑ **Single link (SL) algorithm:** (a_i=1/2, a_j=1/2, b=0, c=-1/2).

In this case: $d(C_q,C_s)=min\{d(C_i,C_s), d(C_j,C_s)\}$

$$d(C_q,C_s)=f(d(C_i,C_s),d(C_j,C_s),d(C_i,C_j))$$

❑ **Weighted Pair Group Method Average (WPGMA)** (a_i=1/2, a_j=1/2, b=0, c=0).

In this case: $d(C_q,C_s)=(d(C_i,C_s) + d(C_j,C_s))/2$

Hierarchical Clustering Algorithms

- Agglomerative Algorithms

$d(C_q,C_s)=a_i d(C_i,C_s)+a_j(d(C_j,C_s)+bd(C_i,C_j)+c|d(C_i,C_s)-d(C_j,C_s)|$

- Algorithms that follow the above equation are:

➢ **Unweighted Pair Group Method Average (UPGMA)**

($a_i=n_i/(n_i+n_j)$, $a_j=n_j/(n_i+n_j)$, $b=0$, $c=0$, where n_i is the cardinality of C_i).

In this case: $d(C_q,C_s)=(n_i\, d(C_i,C_s) + n_j\, d(C_j,C_s))/(n_i+n_j)$

➢ **Unweighted Pair Group Method Centroid (UPGMC)**

($a_i=n_i/(n_i+n_j)$, $a_j=n_j/(n_i+n_j)$, $b=-n_i\, n_j/(n_i+n_j)^2$, $c=0$).

In this case:

$$d_{qs} = \frac{n_i}{n_i+n_j} d_{is} + \frac{n_j}{n_i+n_j} d_{js} - \frac{n_i n_j}{(n_i+n_j)^2} d_{ij}$$

For the UPGMC, it is true that $dqs=\|m_q-m_s\|^2$, where m_q is the mean of C_q

Hierarchical Clustering Algorithms

- Agglomerative Algorithms

$d(C_q,C_s)=a_i d(C_i,C_s)+a_j(d(C_j,C_s)+bd(C_i,C_j)+c|d(C_i,C_s)-d(C_j,C_s)|$

- Algorithms that follow the above equation are:

➢ **Weighted Pair Group Method Centroid (WPGMC)**

(a_i=1/2, a_j=1/2, b=-1/4, c=0). In this case $d_{qs}=(d_{is}+d_{js})/2-d_{ij}/4$

For WPGMC there are cases where $d_{qs} \leq max\{d_{is}, d_{js}\}$ (crossover)

➢ **Ward or minimum variance algorithm**

Here the distance d'_{ij} between C_i and C_j is defined as

$$d'_{ij}=(n_i\, n_j/(n_i+n_j))\, \|\underline{m}_i-\underline{m}_j\|^2$$

Remark: Ward's algorithm forms $\mathcal{R}_{t+1}$ by merging the two clusters that lead to the smallest possible increase of the total variance, i.e., $E_t=\sum_{r=1}^{N-t}\sum_{\underline{x}\in C_r}\|\underline{x}-\underline{m}_r\|^2$

Hierarchical Clustering Algorithms

- Agglomerative Algorithms

➢ **Complexity issues:**

- Generalized Agglomerative Scheme requires, in general, O(N^3) operations.
- More efficient implementations require O($N^2 \log N$) computational time.
- For a class of widely used algorithms, implementations that require O(N^2) computational time and O(N^2) or O(N) storage have also been proposed.
- Parallel implementations on SIMD machines (Single instruction, multiple data) have also been considered.

Hierarchical Clustering Algorithms

- Agglomerative Algorithms (based on graph theory)

- Some basic definitions from graph theory:

➢ A **graph**, G, is defined as an ordered pair $G=(V,E)$, where $V=\{v_i, i=1,...,N\}$ is a set of **vertices** and E is a set of **edges** connecting some pairs of vertices. An edge connecting v_i and v_j is denoted by e_{ij} or (v_i,v_j).

➢ A graph is called **undirected** graph if there is no direction assigned to any of its edges. Otherwise, we deal with **directed** graphs.

➢ A graph is called **unweighted** graph if there is no cost associated with any of its edges. Otherwise, we deal with **weighted** graphs.

➢ A **path** in G between vertices v_{i_1} and v_{i_n} is a sequence of vertices and edges of the form $v_{i_1} e_{i_1 i_2} v_{i_2} ... v_{i_{n-1}} e_{i_{n-1} i_n} v_{i_n}$.

➢ A **loop** in G is a path where v_{i_1} and v_{i_n} coincide.

Hierarchical Clustering Algorithms

- Agglomerative Algorithms (based on graph theory)

- Some basic definitions from graph theory:

➢ A **subgraph** $G'=(V',E')$ of G is a graph with $V' \subseteq V$ and $E' \subseteq E_1$, where E_1 is a subset of E containing vertices that connect vertices of V'. Every graph is a subgraph to itself.

➢ A **connected** subgraph $G'=(V',E')$ is a subgraph where there exists at least one path connecting any pair of vertices in V'.

➢ A **complete subgraph** $G'=(V',E)$ is a subgraph where for any pair of vertices in V' there exists an edge in E'' connecting them.

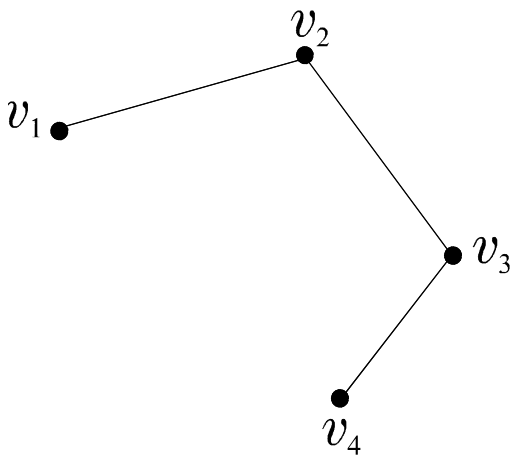
➢ A **maximally connected** subgraph of G is a connected subgraph G' of G that contains as many vertices of G as possible.

➢ A **maximally complete** subgraph of G is a complete subgraph G' of G that contains as many vertices of G as possible.

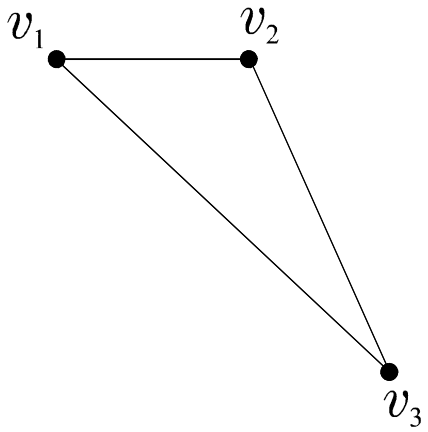
Hierarchical Clustering Algorithms

- Agglomerative Algorithms (based on graph theory)

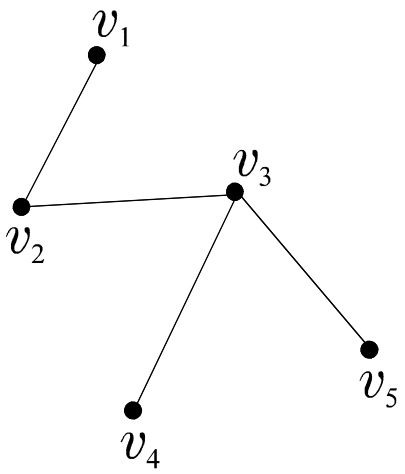
➢ Examples



Path



Loop

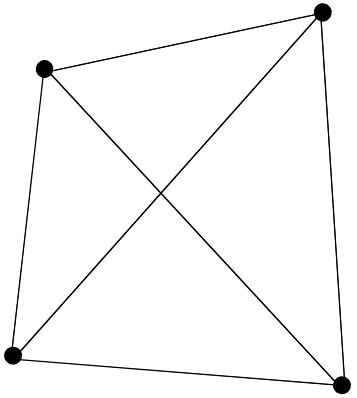


Connected graph

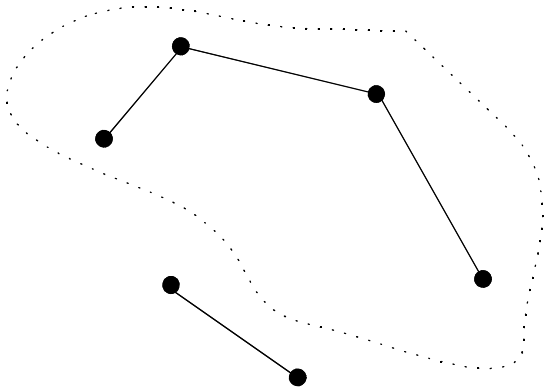
Hierarchical Clustering Algorithms

- Agglomerative Algorithms (based on graph theory)

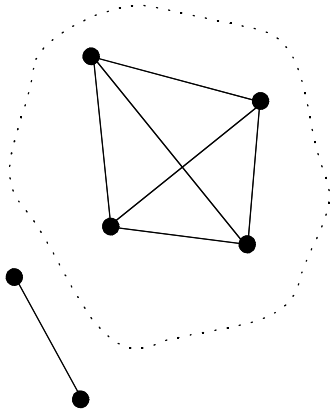
➢ Examples



Complete graph



Maximally connected subgraph



Maximally complete subgraph

Hierarchical Clustering Algorithms

- Agglomerative Algorithms (based on graph theory)

- Useful tools for the algorithms based on graph theory are the threshold graph and the proximity graph.
- **A threshold graph** $G(a)$ is an undirected, unweighted graph with N nodes, each one corresponding to a vector of X.
  - ✓ No self-loops or multiple edges between any two vertices are encountered.
  - ✓ The set of edges of $G(a)$ contains those edges (v_i, v_j) for which the distance $d(x_i,x_j)$ between the vectors corresponding to v_i and v_j is less than a.
- **A proximity graph** $G_p(a)$ is a threshold graph $G(a)$, all of whose edges (v_i, v_j) are weighted with the proximity measure $d(x_i,x_j)$.

Hierarchical Clustering Algorithms

- Agglomerative Algorithms –MST

➢ **Definitions:**

❑ **Spanning Tree**: It is a connected graph (containing all the vertices of the graph), with no loops

❑ **Weight of a Spanning Tree:** The sum of the weights of its edges (provided that they have been assigned with a weight).

❑ **Minimum Spanning Tree (MST):** A spanning tree with the smallest weight among the spanning trees connecting all the vertices of the graph.

➢ **Remarks:**

❑ The MST has N-1 edges.

❑ When all the weights are different from each other, the MST is unique. Otherwise, it may not be unique.

Hierarchical Clustering Algorithms

- Divisive Algorithms

- Let $g(C_i,C_j)$ be a dissimilarity function between two clusters.

- Let C_{tj} denote the j-th cluster of the t-th clustering $\mathcal{R}_t$, $t = 0,\ldots, N\text{-}1$, $j = 1,\ldots, t+1$

Hierarchical Clustering Algorithms

- Divisive Algorithms

➢ Generalized Divisive Scheme (GDS)

- Initialization
  - Choose $\mathcal{R}_0=\{X\}$ as the initial clustering
  - $t = 0$
- Repeat
  - $t = t + 1$
  - For $i = 1$ to t
    - Among all possible pairs of clusters (C_r,C_s) that form a partition of $C_{t-1,i}$, find the pair $(C^1_{t-1,i}, C^2_{t-1,i})$ that gives the maximum value for g.
  - End for
  - From the t pairs defined in the previous step, choose the one that maximizes g. Suppose that this is $(C^1_{t-1,j}, C^2_{t-1,j})$.
  - The new clustering is: $\mathcal{R}_t=(\mathcal{R}_{t-1} - \{C_{t-1,j}\})\cup\{C^1_{t-1,j}, C^2_{t-1,j}\}$
  - Relabel the clusters of $\mathcal{R}_t$.
- Until each vector lies in a single cluster.

Hierarchical Clustering Algorithms

- Divisive Algorithms

➢ **Remarks:**

- Different choices of g (the dissimilarity function between two clusters) rise to different algorithms.
- The GDS is computationally very demanding even for small N.
- Algorithms that rule out many partitions as not “reasonable”, under a prespecified criterion, have also been proposed.
- Algorithms where the splitting of the clusters is based on all features of the feature vectors are called polythetic algorithms. Otherwise, if the splitting is based on a single feature at each step, the algorithms are called monothetic algorithms.

Hierarchical Clustering Algorithms

- Divisive Algorithms

➢ **Choice of the Best Number of Clusters**

- A major issue associated with hierarchical algorithms is: “How the clustering that best fits the data is extracted from a hierarchy of clusterings?”

➢ **Some approaches:**

- Search in the proximity dendrogram for clusters that have a large lifetime (the difference between the proximity level at which a cluster is created and the proximity level at which it is absorbed into a larger cluster (however, this method involves human subjectivity)).

Hierarchical Clustering Algorithms

- Divisive Algorithms

➢ **Choice of the Best Number of Clusters**

- Define a function $h(C)$ that measures the dissimilarity between the vectors of the same cluster C. Let ϑ be an appropriate threshold for h(C). Then $\Re_t$ is the final clustering if there exists a cluster C in $\Re_t$+1 with dissimilarity between its vectors ($h(C)$) greater than ϑ (extrinsic method). The final clustering $\Re_t$ must satisfy the following condition:

$$d^{ss}{}_{min}(C_i,C_j) > max\{h(C_i),h(C_j)\}, \quad \forall Ci,Cj \in \Re_t$$

- Another idea: Muhlenbach F. & Lallich S.: “A New Clustering Algorithm Based on Regions of Influence with Self-Detection of the Best Number of Clusters.” ICDM 2009: 884-889

Hierarchical Clustering Algorithms

- Divisive Algorithms

➢ **Conclusion**

- Major weakness of agglomerative clustering methods:
  - ❑ do not scale well: time complexity of at least O(n^2), where n is the number of total objects
  - ❑ can never undo what was done previously
- Integration of hierarchical with distance-based clustering:
  - ❑ **BIRCH** (1996): uses CF-tree and incrementally adjusts the quality of sub-clusters
  - ❑ **CURE** (1998): selects well-scattered points from the cluster and then shrinks them towards the center of the cluster by a specified fraction
  - ❑ **CHAMELEON** (1999): uses dynamic modeling

Partitioning Algorithms

- Introduction

- **Basic concept:** Partitioning algorithms are clustering algorithms working by function optimization
- **Partitioning method:** Construct a partition of a database D of n objects into a set of k clusters
- Given a k, find a partition of k clusters that optimizes the chosen partitioning criterion
  - Global optimal: exhaustively enumerate all partitions
  - Heuristic methods: k-means and k-medoids algorithms
  - k-means (MacQueen'67): each cluster is represented by the center of the cluster
  - k-medoids or PAM (Kaufman & Rousseeuw'87): each cluster is represented by one of the cluster objects

Partitioning Algorithms

- Clustering Algorithms via Function Optimization

- In this context the clusters are assumed to be described by a parametric specific model whose parameters are unknown (all parameters are included in a vector denoted by θ).
- **Examples:**
  - Compact clusters. Each cluster C_i is represented by a point $\underline{m}_i$ in the l-dimensional space. Thus $\underline{\theta}=[\underline{m}_1{}^T, \underline{m}_2{}^T, \ldots, \underline{m}_m{}^T]^T$.
  - Ring-shaped clusters. Each cluster C_i is modeled by a hypersphere $C(\underline{c}_i,r_i)$, where $\underline{c}_i$ and r_i are its center and its radius, respectively. Thus $\underline{\theta}=[\underline{c}_1{}^T, r_1, \underline{c}_2{}^T, r_2, \ldots, \underline{c}_m{}^T, r_m]^T$.
- A cost $J(\theta)$ is defined as a function of the data vectors in X and θ. Optimization of $J(\theta)$ with respect to θ results in θ that characterizes optimally the clusters underlying X.

Partitioning Algorithms

- K-Means

- Given k, the k-means algorithm is implemented in four steps:

1. Partition objects into k non-empty subsets

2. Compute seed points as the centroids of the clusters of the current partition (the centroid is the center, *i.e.*, mean point, of the cluster)

3. Assign each object to the cluster with the nearest seed point

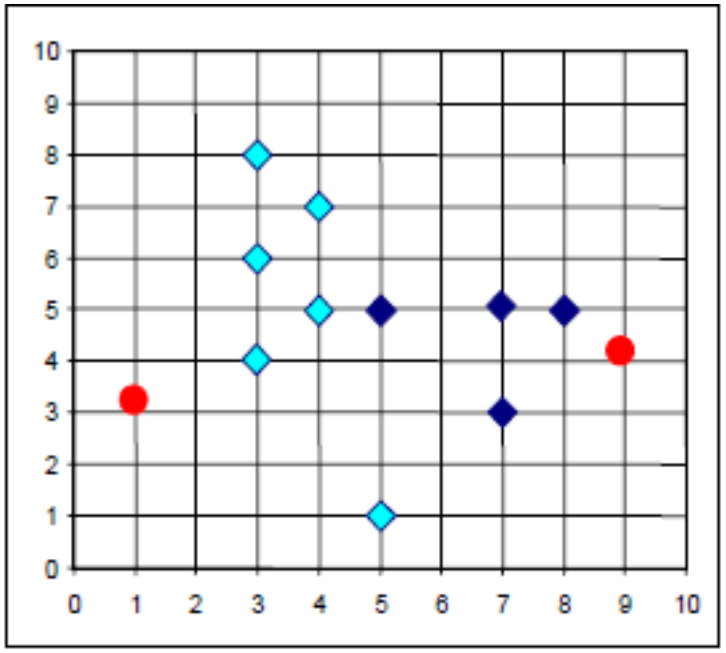
4. Go back to Step 2, stop when no more new assignment

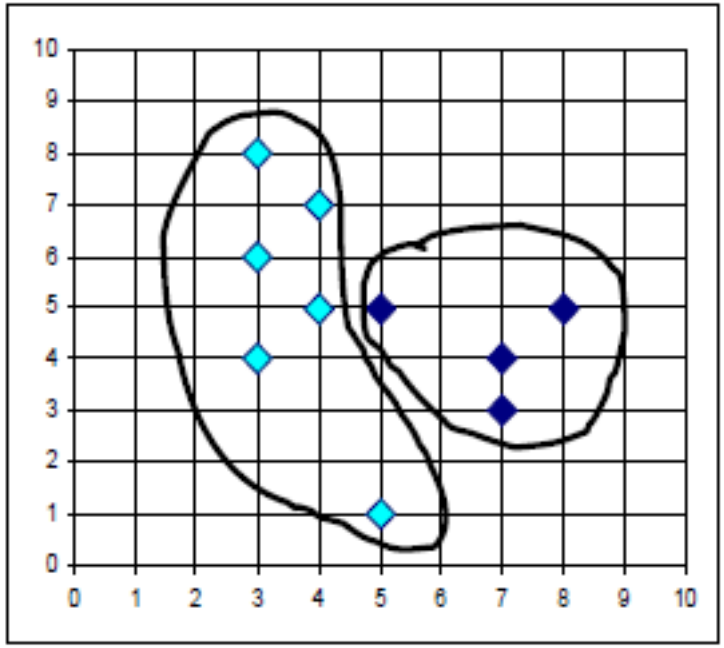
Partitioning Algorithms

- K-Means

$k = 2$

Arbitrarily choose k object as initial cluster center



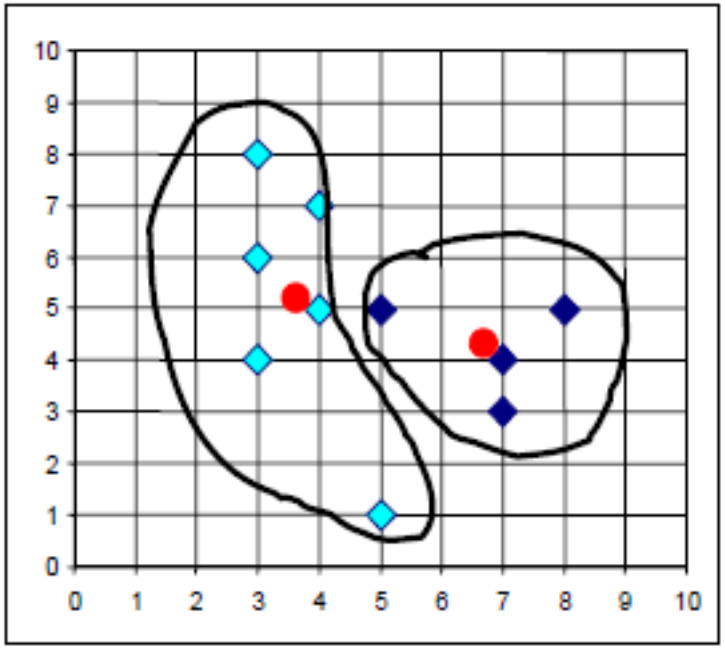


Assign each objects to most similar center

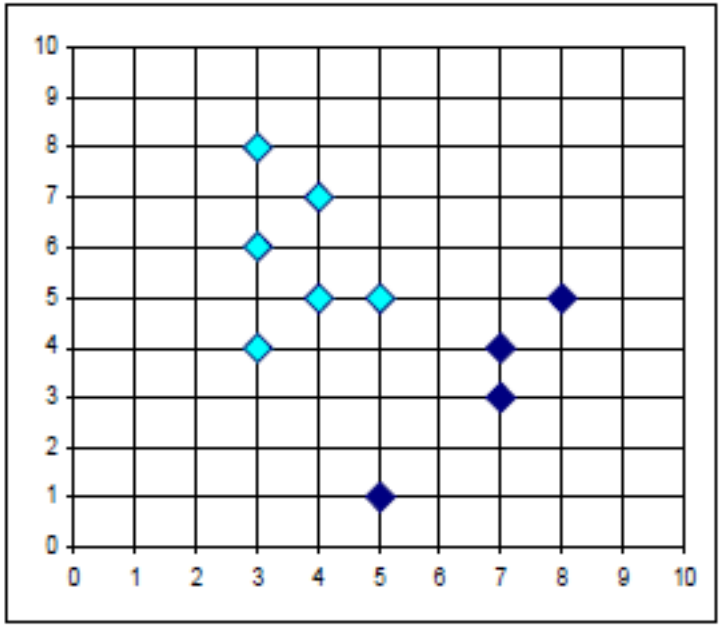
Partitioning Algorithms

- K-Means

Update the cluster means



Reassign each objects
to most similar center



Reupdate the
cluster means…

Partitioning Algorithms

- K-Means

➢ **Strength:**
Relatively efficient: O(ktn), where n is # objects, k is # clusters, and t is # iterations. Normally, k, t << n.

➢ **Comment:** Often terminates at a local optimum.
The global optimum may be found using other techniques

➢ **Weakness:**

❑ Applicable only when mean is defined, then what about categorical data or proximity measures without mean (e.g., orthodromic distance)?

❑ Need to specify k, the number of clusters, in advance

❑ Unable to handle noisy data and outliers

❑ Not suitable to discover clusters with non-convex shapes

Partitioning Algorithms

- K-Medoids and other K-Means Variants

- Find representative objects, called *medoids*, in clusters
- PAM (Partitioning Around Medoids, 1987)
  - starts from an initial set of medoids and iteratively replaces one of the medoids by one of the non-medoids if it improves the total distance of the resulting clustering
  - PAM works effectively for small data sets, but does not scale well for large data sets
- CLARA (Kaufmann & Rousseeuw, 1990)
- CLARANS (Ng & Han, 1994): Randomized sampling
- Focusing + spatial data structure (Ester et al., 1995)

Clustering Algorithms –Miscellaneous

- Conclusion

➢ Other strategies:

- Graph theory based clustering algorithms → other graphs than the MST, e.g., algorithms based on region of influence
- Density-based clustering algorithms → discover clusters of arbitrary shape, handle noise, e.g., DBSCAN, CLIQUE...
- Competitive learning algorithms → neural network clustering, e.g., the Self-Organizing Map (SOM)
- Clustering algorithms for high dimensional data sets → reduction of the dimensionality (by feature selection or PCA) to avoid the “curse of dimensionality”, or approaches based on the clustering on a subspace of the dataset

Clustering Validation

- The right questions to ask

- What is “good clustering”?
- How many clusters?
- How to find the clusters?

- A good clustering method will produce high quality clusters:
  - high intra-class similarity
  - low inter-class similarity

- The quality of a clustering result depends on both the similarity measure used by the method and its implementation.
- The quality of a clustering method is also measured by its ability to discover some or all of the hidden patterns.

Clustering Validation

- Clustering Quality Measures

- Dissimilarity/Similarity metric: Similarity is expressed in terms of a distance function, which is typically metric: $d(i, j)$
- There is a separate "quality" function that measures the "goodness" of a cluster.
- The definitions of distance functions are usually very different for interval-scaled, Boolean, categorical, ordinal and ratio variables.
- Weights should be associated with different variables based on applications and data semantics.
- It is hard to define "similar enough" or "good enough"
  → the answer is typically highly subjective.

Clustering Validation

- Clustering Quality Measures

- For cluster analysis, the question is how to evaluate the "goodness" of the resulting clusters?

- But "clusters are in the eye of the beholder"!

- Then why do we want to evaluate them?
  - To avoid finding patterns in noise
  - To compare clustering algorithms
  - To compare two sets of clusters
  - To compare two clusters

Clustering Validation

- Different Aspects of Cluster Validation

1. Determining the clustering tendency of a set of data, i.e., distinguishing whether non-random structure actually exists in the data (one cluster).
2. Comparing the results of a cluster analysis to externally known results, e.g., to externally given class labels.
3. Evaluating how well the results of a cluster analysis fit the data *without* reference to external information → Use only the data
4. Comparing the results of two different sets of cluster analyses to determine which is better.
5. Determining the 'correct' number of clusters.

For 2, 3, and 4, we can further distinguish whether we want to evaluate the entire clustering or just individual clusters.

Clustering Validation

• Framework for Cluster Validity

- Need a framework to interpret any measure.
  - For example, if our measure of evaluation has the value, 10, is that good, fair, or poor?
- Statistics provide a framework for cluster validity
  - The more “atypical” a clustering result is, the more likely it represents valid structure in the data
  - Can compare the values of an index that result from random data or clusterings to those of a clustering result (if the value of the index is unlikely, then the cluster results are valid)
  - These approaches are more complicated and harder to understand.
- For comparing the results of two different sets of cluster analyses, a framework is less necessary.
  - However, there is the question of whether the difference between two index values is significant

Clustering Validation

• Measures of Cluster Validity

- Numerical measures that are applied to judge various aspects of cluster validity, are classified into the following three types.
  - **External Index**: Used to measure the extent to which cluster labels match externally supplied class labels (e.g., entropy)
  - I**nternal Index**: Used to measure the goodness of a clustering structure without respect to external information (e.g., Sum of Squared Error)
  - **Relative Index**: Used to compare two different clusterings or clusters (often an external or internal index is used for this function, like SSE or entropy)

- Sometimes these are referred to as criteria instead of indices
  - However, sometimes criterion is the general strategy and index is the numerical measure that implements the criterion.

Clustering Validation

• Clustering tendency

- Most clustering algorithms impose a clustering structure to the data set *X* at hand. However, *X* may not possess a clustering structure. Before we apply any clustering algorithm on X, it must first be verified that X possesses a clustering structure. This is known as **clustering tendency**.
- Clustering tendency is heavily based on hypothesis testing.
- Specifically, it is based on testing the randomness (null) hypothesis (H_0) against the regularity (H_1) hypothesis and the clustering (H_2) hypothesis
  - **Randomness hypothesis** (H_0): "The vectors of *X* are randomly distributed, according to the uniform distribution in the sampling window (the compact convex support set for the underlying distribution of the vectors of the data set *X*) of *X*".
  - **Regularity hypothesis** (H_1): "The vectors of *X* are regularly spaced (that is they are not too close to each other) in the sampling window".
  - **Clustering hypothesis** (H_2): "The vectors of *X* form clusters".

Clustering Validation

- External and internal criteria

- Hypothesis testing is employed.
- The null hypothesis H_0, which is a statement of randomness concerning the structure of *X*, is defined.
- The generation of a reference data population under the random hypothesis takes place.
- An appropriate statistic, *q*, whose values are indicative of the structure of a data set, is defined. The value of *q* that results from our data set *X* is compared against the values obtained for *q* when the elements of the reference (random) population are considered.
- Ways for generating reference populations under the null hypothesis (each one used in different situations):
  - Random position hypothesis.
  - Random graph hypothesis.
  - Random label hypothesis.

Clustering Validation

- Statistics suitable for external criteria

➢ For the comparison of C with an independently drawn partition P of X:
  - ❑ Rand statistic
  - ❑ Jaccard statistic
  - ❑ Fowlkes-Mallows index
  - ❑ Hubert's Γ statistic
  - ❑ Normalized Γ statistic

➢ For assessing the agreement between P and the proximity matrix P:
  - ❑ Γ statistic

Clustering Validation

- Algorithm for external criteria

➢ Algorithm for matching partitions and clusters

MatchPartitionCluster $(P, C, match)$:

1 **foreach** $p \in P$ **do**
2 $match(p) \leftarrow \emptyset$
3 **foreach** $c \in C$ **do**
4 $overlap(p, c) \leftarrow \frac{|p \cap c|}{|p|}$
5 **while** $overlap \neq \emptyset$ **do**
6 $(p_{max}, c_{max}) \leftarrow GetMaxOverlap(overlap)$
7 $match(p_{max}) \leftarrow c_{max}$
8 $overlap \leftarrow overlap - \{overlap(p_{max}, *), overlap(*, c_{max})\}$

Clustering Validation

- Purity-based measures for external criteria

➢ Purity $\frac{|c_i \cap p_j|}{|c_i|}$ $\max_j \rho_{ij}$ $purity_C = \sum_r \frac{|c_i|}{|c|} purity_i$

➢ Precision / Recall / F-Measure
prec(i,j), recall(i,j)

$$F(i,j) = \frac{2 \times prec(i,j) \times rec(i,j)}{prec(i,j) + rec(i,j)}$$

➢ Entropy $e_i = -\sum_q \rho_{ij} \log_2 \rho_{ij}$

$$e_C = \sum_r \frac{|c_i|}{|c|} e_i$$

Clustering Validation

- Matching measures for external criteria

➢ Rand statistics [Rand, 1985]

❑ Given partition (P) and ground truth (G), measure the number of vector pairs that are:

a: in the same class both in P and G.

b: in the same class in P but different classes in G.

c: in different classes in P but in the same class in G.

d: in different classes both in P and G.

		G		
		1	0	*sum*
P	1	a	b	$a+b$
	0	c	d	$c+d$
	sum	$a+c$	$b+d$	m

$$R = \frac{a+d}{a+b+c+d}$$

➢ Jaccard index [Jaccard, 1901] $$J = \frac{a}{a+b+c}$$

Clustering Validation

- Correlation measures for external criteria

➢ Hubert

$$\Gamma = \frac{1}{m}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} X_P(i,j)X_C(i,j)$$

➢ Normalized Tau Statistics:

$$\hat{\Gamma} = \frac{\frac{1}{m}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}(X_P(i,j)-\mu_P)(X_C(i,j)\mu_C)}{\sigma_P\sigma_C}$$

where μ_P and μ_C are the means and σ_P and σ_C are the variances of the matrices X_C and X_P.

Clustering Validation

- Statistics suitable for internal criteria

➢ Validation of hierarchy of clusterings

- ❑ Cophenetic correlation coefficient (CPCC)
- ❑ γ statistic
- ❑ Kudall's τ statistic.

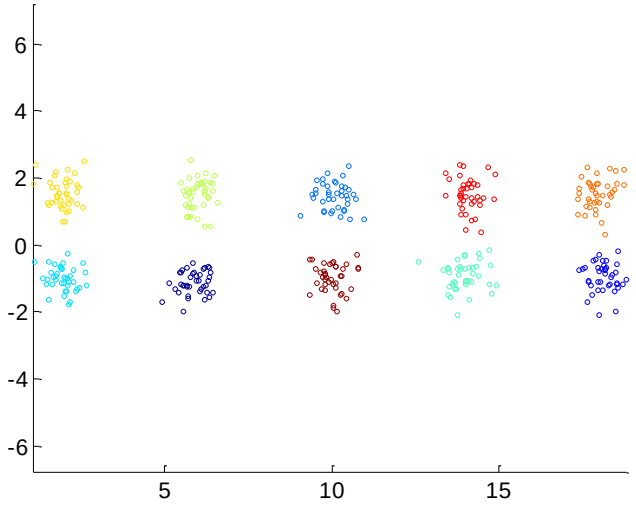
➢ Validation of individual clusterings

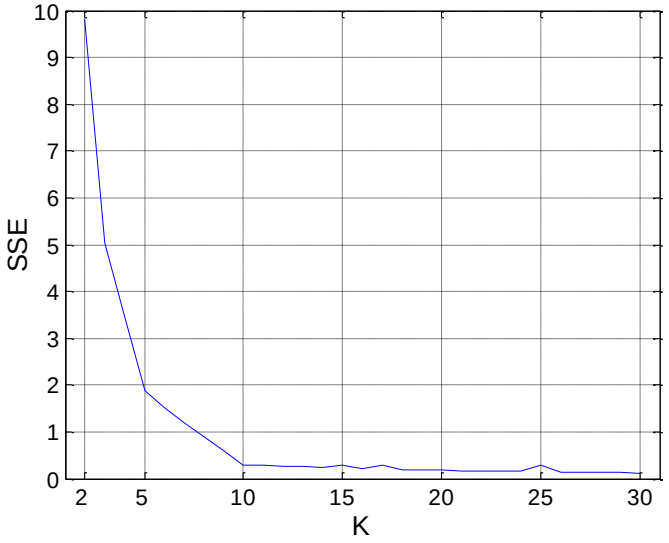
- ❑ Γ statistic
- ❑ Normalized Γ statistic.

Clustering Validation

- Internal validation –Sum of square error

- Clusters in more complicated figures aren't well separated
- Internal Index: Used to measure the goodness of a clustering structure without respect to external information (SSE)
- SSE is good for comparing two clusterings or two clusters (average SSE).
- Can also be used to estimate the number of clusters





Clustering Validation

- Internal validation –Cohesion and separation

- Cluster Cohesion: Measures how closely related are objects in a cluster (e.g., SSE)
- Cluster Separation: Measure how distinct or well-separated a cluster is from other clusters

- **Example:** Squared Error
  - Cohesion is measured by the **within** cluster sum of squares (SSE)

$$\boldsymbol{WSS} = \sum_{\boldsymbol{i}} \sum_{\boldsymbol{x} \in \boldsymbol{C_i}} (\boldsymbol{x} - \boldsymbol{m_i})^2$$

  - Separation is measured by the **between** cluster sum of squares

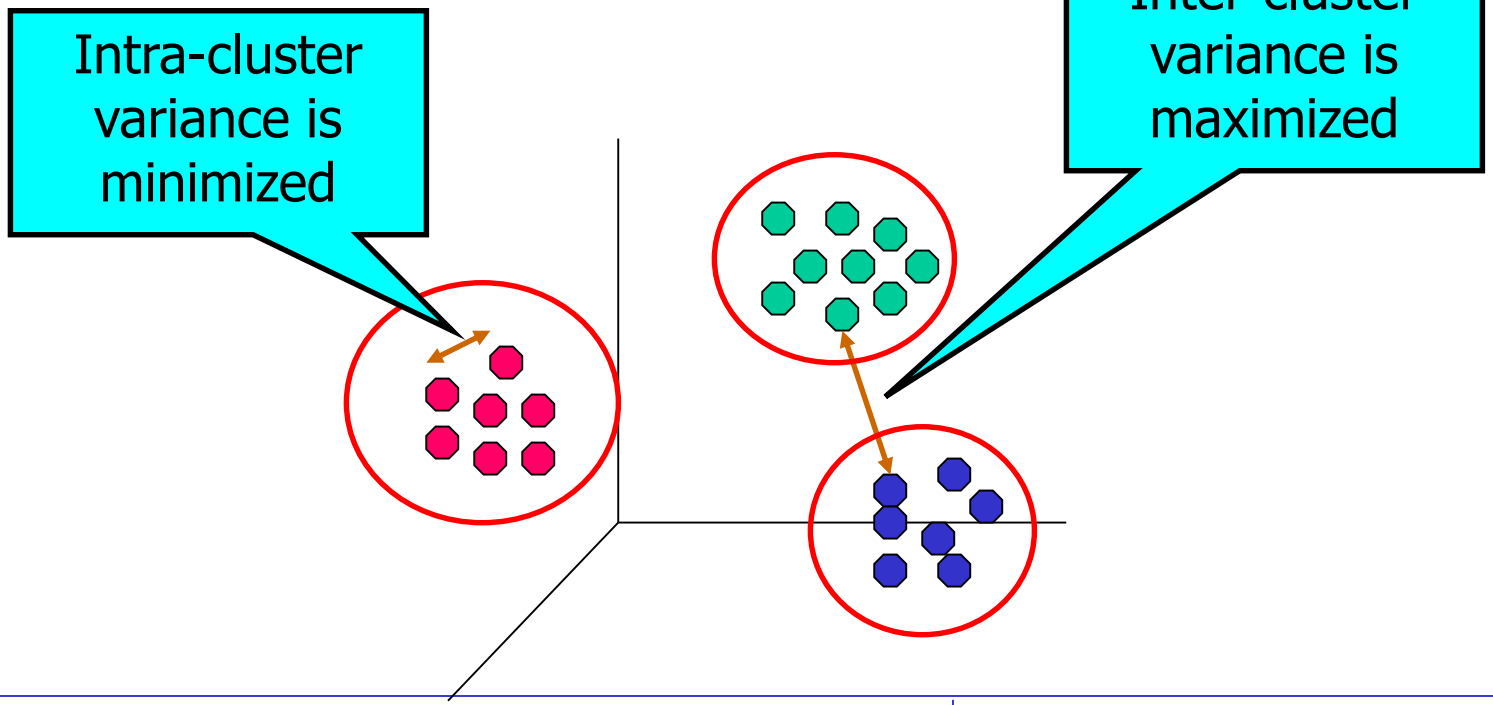
$$BSS = \sum_{i} |C_i| (m - m_i)^2$$

Where $|C_i|$ is the size of cluster i

Clustering Validation

- Internal validation –Cohesion and separation

➢ From MSE to cluster validity
  - ❑ Minimize within cluster variance (MSE)
  - ❑ Maximize between cluster variance



Clustering Validation

- Internal validation –Cohesion and separation

- SSW / m ---- Ball and Hall (1965)
- $m^2|W|$ ---- Marriot (1971)
- $\frac{SSB / m - 1}{SSW / n - m}$ ---- Calinski & Harabasz (1974)
- $\log(SSB/SSW)$ ---- Hartigan (1975)
- $d \log(\sqrt{SSW /(dn^2)}) + \log(m)$ ---- Xu (1997)

d is the dimension of data

n is the size of data, m is the number of clusters

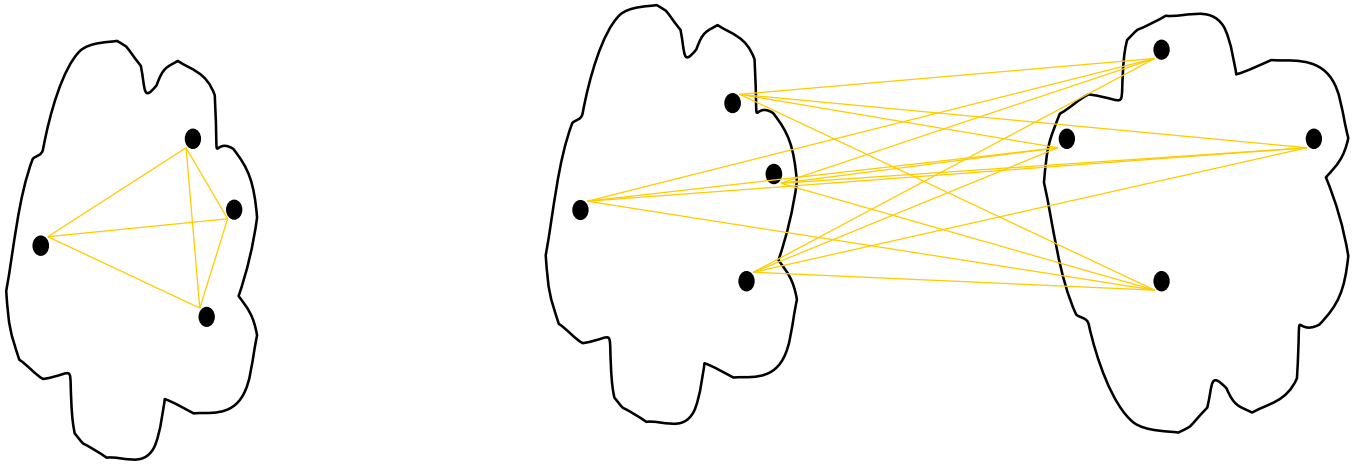
SSW is the sum of squares within the clusters

SSB is the sum of squares between the clusters

Clustering Validation

- Internal validation –Cohesion and separation

➢ A proximity graph based approach can also be used for cohesion and separation.
  - ❑ Cluster cohesion is the sum of the weight of all links within a cluster.
  - ❑ Cluster separation is the sum of the weights between nodes in the cluster and nodes outside the cluster.



cohesion separation

Clustering Validation

- Internal validation –Cohesion and separation

- **Cohesion** $a(x)$: average distance of x to all other vectors in the same cluster.

- **Separation** $b(x)$: average distance of x to the vectors in other clusters. Find the minimum among the clusters.

- **silhouette** $s(x)$: $$s(x) = \frac{b(x) - a(x)}{\max\{a(x), b(x)\}}$$

$s(x)$ = [-1, +1]: -1=bad, 0=indifferent, 1=good

- **Silhouette coefficient** (SC): $$SC = \frac{1}{N}\sum_{i=1}^{N} s(x)$$

Clustering Validation

• Internal validation –Relative criteria

- Let *A* denote the set of parameters of a clustering algorithm.
- Statement of the problem:
  "Among the clusterings produced by a specific clustering algorithm, for different values of the parameters in *A*, choose the one that best fits the data set *X*".
- We consider two cases

1. *A* does not contain the number of clusters *m*.
   The estimation of the best set of parameter values is carried out as follows:
   - Run the algorithm for a wide range of values of its parameters.
   - Plot the number of clusters, *m*, versus the parameters of *A*.
   - Choose the widest range for which *m* remains constant.
   - Adopt the clustering that corresponds to the values of the parameters in *A* that lie in the middle of this range.

Clustering Validation

• Internal validation –Relative criteria

➢ We consider two cases

2. *A* does contain the number of clusters *m*.

The estimation of the best set of parameter values is carried out as follows:

- ❑ Select a suitable performance index *q* (the best clustering).
- ❑ For $m=m_{min}$ to m_{max}
  - ❑ Run the algorithm *r* times using different sets of values for the other parameters of *A* and each time compute *q*.
  - ❑ Choose the clustering that corresponds to the best *q*.
- ❑ End for
- ❑ Plot the best values of *q* for each *m* versus *m*.
- ❑ The presence of a significant knee indicates the number of clusters underlying *X*. Adopt the clustering that corresponds to that knee.
- ❑ The absence of such a knee indicates that *X* possesses no clustering structure.

Clustering Validation

- Final Comment on Cluster Validity

"The validation of clustering structures is the most difficult and frustrating part of cluster analysis.

Without a strong effort in this direction, cluster analysis will remain a black art accessible only to those true believers who have experience and great courage."

Algorithms for Clustering Data, Jain and Dubes

Clustering – Conclusion

- Importance of:
  - ✓ a proximity measure (distance / similarity)
  - ✓ a clustering criterion (SL, CL)
  - ✓ a clustering algorithm
- Clustering algorithms:
  - ✓ sequential algorithms (BSAS → θ?)
  - ✓ hierarchical algorithms (GAS, DAS, MST → dendrogram)
  - ✓ partitioning algorithms (k-Means, k-Medoids → k?)
  - ✓ other clustering algorithms (MST, SOM, DBSCAN, CLIQUE)
- Outlier analysis
- Clustering validation