Brève introduction au domaine du Clustering
(Classification Automatique ou Analyse Typologique)

Julien Ah-Pine (julien.ah-pine@eric.univ-lyon2.fr)

Laboratoire ERIC

1/2 Journée Clustering - Lyon - 17/02/14
Sommaire

1. Qu’est ce que le clustering et à quoi ça peut servir ?
2. Différentes façons de classifier les éléments
3. Formalisation du problème du clustering
4. Quelques mesures de proximité
5. Quelques algorithmes classiques de clustering
   - Classification hiérarchique
   - Partitionnement “dur”
   - Partitionnement “flou”
6. Evaluation du clustering par Prof. Lallich
Qu’est ce que le clustering et à quoi ça peut servir ?

Différentes façons de classifier les éléments

Formalisation du problème du clustering

Quelques mesures de proximité

Quelques algorithmes classiques de clustering

Evaluation du clustering par Prof. Lallich
Data Clustering: what is it about?

- **Data clustering** (or just clustering), also called **cluster analysis**, **segmentation analysis**, **taxonomy analysis**, or **unsupervised classification**: set of methods that aim at creating groups of objects, or clusters, such that objects in a cluster are very similar and objects in different clusters are distinct.
Data Clustering : what is it about?

- **Data clustering** (or just clustering), also called **cluster analysis**, **segmentation analysis**, **taxonomy analysis**, or **unsupervised classification**: set of methods that aim at creating groups of objects, or clusters, such that objects in a cluster are very similar and objects in different clusters are distinct.

- Different from **supervised classification**: in a supervised context, objects are already assigned to predefined categories or labels and the goal is to learn from a training (labeled) dataset, a decision function that correctly categorize new objects.

→ In **data clustering** the goal is to automatically discover a classification of the objects.
Data Clustering: what is it about?

- **Data clustering** (or just clustering), also called **cluster analysis**, **segmentation analysis**, **taxonomy analysis**, or **unsupervised classification**: set of methods that aim at creating groups of objects, or clusters, such that objects in a cluster are very similar and objects in different clusters are distinct.

- Different from **supervised classification**: in a supervised context, objects are already assigned to predefined categories or labels and the goal is to learn from a training (labeled) dataset, a decision function that correctly categorize new objects.

  - In **data clustering** the goal is to automatically discover a classification of the objects.

- In the following, the term **classification** will refer to the concept of organizing similar objects into groups.
Classification in human activities and sciences

- Classification is a basic human activity:
  - Early men and food: eatable or poisonous?
  - Natural language: each noun is a label used to describe a class of things which have features in common...
Classification in human activities and sciences

- Classification is a basic human activity:
  - Early men and food: eatable or poisonous?
  - Natural language: each noun is a label used to describe a class of things which have features in common.

- Classification is also fundamental to most branches of science:
  - In biology: classification of organisms (taxonomies)
  - In chemistry: classification of chemical elements
  - In astronomy: classification of stars and galaxies.
Classification in human activities and sciences

- Classification is a basic human activity:
  - Early men and food: eatable or poisonous?
  - Natural language: each noun is a label used to describe a class of things which have features in common.

- Classification is also fundamental to most branches of science:
  - In biology: classification of organisms (taxonomies)
  - In chemistry: classification of chemical elements
  - In astronomy: classification of stars and galaxies

- A classification scheme may simply represent a convenient method for organizing a set of objects so that it can be understood more easily and the information it conveys retrieved more efficiently.
Data clustering: why is it also useful in computer science?

- Databases are larger and larger and more and more areas of science due to the development of IT.
Data clustering: why is it also useful in computer science?

- Databases are larger and larger and more and more areas of science due to the development of IT.
- For large datasets, designing a classification scheme by hand is unfeasible.
Data clustering: why is it also useful in computer science?

- Databases are larger and larger and more and more areas of science due to the development of IT.
  - For large datasets, designing a classification scheme by hand is unfeasible.

- **Data clustering** is the process which aims at automatically discovering a classification scheme in order to organize the objects of a large database.
Data clustering : why is it also useful in computer science?

- Databases are larger and larger and more and more areas of science due to the development of IT.

  → For large datasets, designing a classification scheme by hand is unfeasible.

- **Data clustering** is the process which aims at automatically discovering a classification scheme in order to organize the objects of a large database.

- The exploration of such databases using data clustering and other multivariate analysis techniques is now often called **data mining**.
Some applications of data clustering

- In market analysis: segment the market (the customers) and determine target markets.
Some applications of data clustering

- In market analysis: segment the market (the customers) and determine target markets.

- In information retrieval: cluster the results provided by a search engine so that the user can browse the retrieved items in a more efficient way (have you ever tried yippy.com (formerly clusty.com)?)
Some applications of data clustering

- In market analysis: segment the market (the customers) and determine target markets.

- In information retrieval: cluster the results provided by a search engine so that the user can browse the retrieved items in a more efficient way (have you ever tried yippy.com (formerly clusty.com)?)

- In image processing: segment a gray-scale or a color image in order to detect objects and/or to compress the image.
Some applications of data clustering

- In market analysis: segment the market (the customers) and determine target markets.

- In information retrieval: cluster the results provided by a search engine so that the user can browse the retrieved items in a more efficient way (have you ever tried yippy.com (formerly clusty.com)?)

- In image processing: segment a gray-scale or a color image in order to detect objects and/or to compress the image.

- In on-line social network analysis (such as Facebook, LinkedIn, . . .): detect communities given a graph of relationships between people . . .
Rappel du Sommaire

1. Qu’est ce que le clustering et à quoi ça peut servir ?

2. Différentes façons de classifier les éléments

3. Formalisation du problème du clustering

4. Quelques mesures de proximité

5. Quelques algorithmes classiques de clustering

6. Evaluation du clustering par Prof. Lallich
Different types of classification scheme

- We can have different types of classification schemes:
  - A flat partition (set of clusters or segments)
  - A hierarchical tree or taxonomy (a set of nested partitions)
  - Hard or soft (or fuzzy) memberships to clusters
Flat partition

- A flat partition is a crisp or hard clustering or an equivalence relation.
Flat partition

- A flat partition is a crisp or hard clustering or an equivalence relation.

Flat partition:
\{x_1, x_2, x_3\}, \{x_4, x_5\}, \{x_6\}
Flat partition

- A flat partition is a crisp or hard clustering or an equivalence relation.

- Equivalence relation: an edge between $x_i$ and $x_j$ means “$x_i$ is in the same cluster as $x_j$”. The adjacency matrix $A$ is:

$$A = \begin{pmatrix}
  1 & 1 & 1 & 1 & 1 & 1 \\
  1 & 1 & 1 & 1 & 1 & 1 \\
  1 & 1 & 1 & 1 & 1 & 1 \\
  1 & 1 & 1 & 1 & 1 & 1 \\
  1 & 1 & 1 & 1 & 1 & 1 \\
  1 & 1 & 1 & 1 & 1 & 1 \\
\end{pmatrix}$$
Flat partition

- A flat partition is a crisp or hard clustering or an equivalence relation.

- Assignment relation: each item is assigned to one cluster. The assignment matrix $U$ is:

\[
U = \begin{pmatrix}
1 & C_2 & C_3 \\
x_1 & 1 & 1 \\
x_2 & 1 & \\
x_3 & 1 & \\
x_4 & 1 & 1 \\
x_5 & 1 & 1 \\
x_6 & 1 & 1
\end{pmatrix}
\]
A hierarchical clustering is a set of nested flat partitions.
Hierarchical clustering and dendrograms

- A hierarchical clustering is a set of nested flat partitions.
Hierarchical clustering and dendrograms

- A hierarchical clustering is a set of nested flat partitions.
Hierarchical clustering and dendrograms

- A hierarchical clustering is a set of nested flat partitions.
Hierarchical clustering and dendrograms

- A hierarchical clustering is a set of nested flat partitions.
A hierarchical clustering is a set of nested flat partitions.
Fuzzy partition

- In a fuzzy partition objects can belong to different clusters.
Fuzzy partition

- In a fuzzy partition objects can belong to different clusters.

- Fuzzy assignment relation: each item can have a non null membership value with several clusters. The fuzzy assignment matrix $U$ is:

$$
U = \begin{pmatrix}
C_1 & C_2 & C_3 \\
1 & 1 & 0.75 & 0.25 & 0.67 & 0.33 \\
1 & 0.25 & 1 & 1
\end{pmatrix}
$$
Rappel du Sommaire

1. Qu’est ce que le clustering et à quoi ça peut servir ?
2. Différentes façons de classifier les éléments
3. Formalisation du problème du clustering
4. Quelques mesures de proximité
5. Quelques algorithmes classiques de clustering
6. Evaluation du clustering par Prof. Lallich
The partitioning problem

Suppose $F(\mathbb{D}, C)$ is a function that measures the \textbf{quality} of the clustering $C$ given the set of data points $\mathbb{D}$. 
The partitioning problem

- Suppose $F(\mathbb{D}, C)$ is a function that measures the **quality** of the clustering $C$ given the set of data points $\mathbb{D}$.
- Let $C_n$ be the set of all partitions of a set of $n$ data points.
The partitioning problem

- Suppose $F(\mathbb{D}, C)$ is a function that measures the quality of the clustering $C$ given the set of data points $\mathbb{D}$.
- Let $\mathcal{C}_n$ be the set of all partitions of a set of $n$ data points.
- The clustering problem can be formally define as follows:

$$\max_{C \in \mathcal{C}_n} F(\mathbb{D}, C)$$
The partitioning problem

- Suppose $F(\mathbb{D}, C)$ is a function that measures the quality of the clustering $C$ given the set of data points $\mathbb{D}$.
- Let $\mathbb{C}_n$ be the set of all partitions of a set of $n$ data points.
- The clustering problem can be formally defined as follows:
  \[
  \max_{C \in \mathbb{C}_n} F(\mathbb{D}, C)
  \]
- A naive approach to solve the clustering problem is the following one:
The partitioning problem

- Suppose \( F(\mathbb{D}, C) \) is a function that measures the quality of the clustering \( C \) given the set of data points \( \mathbb{D} \).
- Let \( \mathbb{C}_n \) be the set of all partitions of a set of \( n \) data points.
- The clustering problem can be formally define as follows:

\[
\max_{C \in \mathbb{C}_n} F(\mathbb{D}, C)
\]

- A naive approach to solve the clustering problem is the following one:
  1. Enumerate all possible partitions in \( \mathbb{C}_n \)
The partitioning problem

- Suppose $F(\mathbb{D}, C)$ is a function that measures the \textbf{quality} of the clustering $C$ given the set of data points $\mathbb{D}$.
- Let $\mathcal{C}_n$ be the set of all partitions of a set of $n$ data points.
- The clustering problem can be formally define as follows:

$$\max_{C \in \mathcal{C}_n} F(\mathbb{D}, C)$$

- A naive approach to solve the clustering problem is the following one:
  1. Enumerate all possible partitions in $\mathcal{C}_n$
  2. For all $C \in \mathcal{C}_n$ compute the value $F(\mathbb{D}, C)$
The partitioning problem

Suppose $F(\mathbb{D}, C)$ is a function that measures the quality of the clustering $C$ given the set of data points $\mathbb{D}$.

Let $\mathbb{C}_n$ be the set of all partitions of a set of $n$ data points.

The clustering problem can be formally define as follows:

$$\max_{C \in \mathbb{C}_n} F(\mathbb{D}, C)$$

A naive approach to solve the clustering problem is the following one:

1. Enumerate all possible partitions in $\mathbb{C}_n$
2. For all $C \in \mathbb{C}_n$ compute the value $F(\mathbb{D}, C)$
3. Keep the partition $C^*$ such that $\forall C \in \mathbb{C}_n : F(\mathbb{D}, C^*) \geq F(\mathbb{D}, C)$
The number of partitions with \( k \) clusters of a set of \( n \) items is the Stirling number of a second kind:

\[
S(n, k) = \frac{1}{k!} \sum_{j=0}^{k} (-1)^j \binom{n}{k} j^n
\]
A combinatorial problem

- The number of partitions with $k$ clusters of a set of $n$ items is the Stirling number of a second kind:

$$S(n, k) = \frac{1}{k!} \sum_{j=0}^{k} (-1)^j \binom{n}{k} j^n$$

- The total number of partitions of a set of $n$ items is the Bell number:

$$B(n) = \sum_{k=0}^{n} S(n, k)$$
A combinatorial problem (cont’d)

- Some values of $S(n, k)$ and $B(n)$:

<table>
<thead>
<tr>
<th>$n \backslash k$</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>$B(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>7</td>
<td>6</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>15</td>
<td>25</td>
<td>10</td>
<td>1</td>
<td>0</td>
<td>52</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>1</td>
<td>31</td>
<td>90</td>
<td>65</td>
<td>15</td>
<td>1</td>
<td>203</td>
</tr>
</tbody>
</table>
A combinatorial problem (cont’d)

- Some values of $S(n, k)$ and $B(n)$:

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>$B(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>7</td>
<td>6</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>15</td>
<td>25</td>
<td>10</td>
<td>1</td>
<td>0</td>
<td>52</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>1</td>
<td>31</td>
<td>90</td>
<td>65</td>
<td>15</td>
<td>1</td>
<td>203</td>
</tr>
</tbody>
</table>

- Another example: $B(71) \approx 4 \times 10^{74}$!
A combinatorial problem (cont’d)

- Some values of $S(n, k)$ and $B(n)$:

<table>
<thead>
<tr>
<th>$n \backslash k$</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>$B(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>7</td>
<td>6</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>15</td>
<td>25</td>
<td>10</td>
<td>1</td>
<td>0</td>
<td>52</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>1</td>
<td>31</td>
<td>90</td>
<td>65</td>
<td>15</td>
<td>1</td>
<td>203</td>
</tr>
</tbody>
</table>

- Another example: $B(71) \approx 4 \times 10^{74}$!

- It is unfeasible to enumerate all partitions of a set whose cardinal is greater than some tens!
A combinatorial problem (cont’d)

- Some values of $S(n, k)$ and $B(n)$:

<table>
<thead>
<tr>
<th>$n \backslash k$</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>$B(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>7</td>
<td>6</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>15</td>
<td>25</td>
<td>10</td>
<td>1</td>
<td>0</td>
<td>52</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>1</td>
<td>31</td>
<td>90</td>
<td>65</td>
<td>15</td>
<td>1</td>
<td>203</td>
</tr>
</tbody>
</table>

- Another example: $B(71) \approx 4 \times 10^{74}$!

- It is unfeasible to enumerate all partitions of a set whose cardinal is greater than some tens!

→ In practice we use heuristics ie clustering algorithms.
Initially there is a database of objects which could be of any kind depending on the type of application.
The clustering process

We assume that objects have a (structured) numerical representation. This is our starting point but there are different types of numerical data.
The clustering process

Depending on the type of clustering algorithm we can have as input a feature matrix or a proximity matrix.
Once the clustering algorithm is done, we have to assess the clustering outputs.
The clustering process

Depending on the assessment of the clustering output, either we keep the latter result or we start over with another modeling.
Important ingredients of the clustering problem

- How to measure the **proximity** between objects?
Important ingredients of the clustering problem

- How to measure the **proximity** between objects?
  - Dissimilarity / Similarity measures
Important ingredients of the clustering problem

- How to measure the **proximity** between objects?
  → Dissimilarity / Similarity measures

- How to measure the **quality** of a partition?
Important ingredients of the clustering problem

- How to measure the **proximity** between objects?
  - Dissimilarity / Similarity measures

- How to measure the **quality** of a partition?
  - Objective function / Clustering model
Important ingredients of the clustering problem

- How to measure the **proximity** between objects?
  → Dissimilarity / Similarity measures

- How to measure the **quality** of a partition?
  → Objective function / Clustering model

- How to **construct** a good partition (wrt the quality measure)?
Important ingredients of the clustering problem

- How to measure the **proximity** between objects?
  → Dissimilarity / Similarity measures

- How to measure the **quality** of a partition?
  → Objective function / Clustering model

- How to **construct** a good partition (wrt the quality measure)?
  → Clustering algorithms / Heuristics
Important ingredients of the clustering problem

- How to measure the **proximity** between objects?
  → Dissimilarity / Similarity measures

- How to measure the **quality** of a partition?
  → Objective function / Clustering model

- How to **construct** a good partition (wrt the quality measure)?
  → Clustering algorithms / Heuristics

- How to **assess** the clustering output of the algorithm (wrt evaluation measures)?
Important ingredients of the clustering problem

- How to measure the **proximity** between objects?
  → Dissimilarity / Similarity measures

- How to measure the **quality** of a partition?
  → Objective function / Clustering model

- How to **construct** a good partition (wrt the quality measure)?
  → Clustering algorithms / Heuristics

- How to **assess** the clustering output of the algorithm (wrt evaluation measures)?
  → Internal / External assessment measures: Prof. Lallich talk
Rappel du Sommaire

1. Qu’est ce que le clustering et à quoi ça peut servir ?
2. Différentes façons de classifier les éléments
3. Formalisation du problème du clustering
4. Quelques mesures de proximité
5. Quelques algorithmes classiques de clustering
6. Evaluation du clustering par Prof. Lallich
Definition of a dissimilarity and a distance measure

\textbf{Définition. (Dissimilarity and distance measures)}

Let $\mathbb{D}$ be a set of data points and let $D : \mathbb{D} \times \mathbb{D} \rightarrow \mathbb{R}^+$ be a real function. $D$ is a \textbf{dissimilarity measure} if it satisfies the following properties :

1. \textit{Non-negativity} : $\forall x, y : D(x, y) \geq 0$
2. \textit{Symmetry} : $\forall x, y : D(x, y) = D(y, x)$
3. \textit{Identity and indiscernability} : $\forall x, y : x = y \iff D(x, y) = 0$

If a dissimilarity measure $D$ also satisfies the following condition, it is a \textbf{distance measure}.

4. \textit{Triangle inequality} : $\forall x, y, z : D(x, y) \leq D(x, z) + D(z, y)$
Definition of a dissimilarity and a distance measure

Let $\mathcal{D}$ be a set of data points and let $D : \mathcal{D} \times \mathcal{D} \rightarrow \mathbb{R}^+$ be a real function. $D$ is a **dissimilarity measure** if it satisfies the following properties:

1. **Non-negativity**: $\forall x, y : D(x, y) \geq 0$
2. **Symmetry**: $\forall x, y : D(x, y) = D(y, x)$
3. **Identity and indiscernability**: $\forall x, y : x = y \iff D(x, y) = 0$

If a dissimilarity measure $D$ also satisfies the following condition, it is a **distance measure**.

4. **Triangle inequality**: $\forall x, y, z : D(x, y) \leq D(x, z) + D(z, y)$

A dissimilarity $D$ that satisfies conditions 3 and 4 is also said to be a **metric**. Thus, a distance measure is a metric.
Définition. (Dissimilarity and distance measures)

Let $\mathbb{D}$ be a set of data points and let $D : \mathbb{D} \times \mathbb{D} \rightarrow \mathbb{R}^+$ be a real function. $D$ is a **dissimilarity measure** if it satisfies the following properties:

1. **Non-negativity** : $\forall x, y : D(x, y) \geq 0$
2. **Symmetry** : $\forall x, y : D(x, y) = D(y, x)$
3. **Identity and indiscernability** : $\forall x, y : x = y \Leftrightarrow D(x, y) = 0$

If a dissimilarity measure $D$ also satisfies the following condition, it is a **distance measure**.

4. **Triangle inequality** : $\forall x, y, z : D(x, y) \leq D(x, z) + D(z, y)$

- A dissimilarity $D$ that satisfies conditions 3 and 4 is also said to be a **metric**. Thus, a distance measure is a metric.
- If from a distance matrix $D$ with $D_{ij} = D(x_i, x_j)$ we can represent the data points in an euclidean space then $D$ is said to be **euclidean**.
No consensus on the axioms defining a similarity measure.

Définition. (Similarity measure)

Let \( \mathbb{D} \) be a set of items represented in an euclidean space and let \( S : \mathbb{D} \times \mathbb{D} \to \mathbb{R} \) be a real function. \( S \) is a similarity measure if it satisfies the following properties:

1. **Boundary conditions**: there are two fixed numbers \( a \) and \( b \) such that \( \forall x, y : a \leq S(x, y) \leq b \) (in general \( a = 0 \) or \( a = -1 \) and \( b = 1 \))

2. **Symmetry**: \( \forall x, y : S(x, y) = S(y, x) \)

3. **Identity and indiscernability**: \( \forall x, y : x = y \iff S(x, y) = b \)

The similarity measure \( S \) is said to be metric if the pairwise similarity matrix \( S \) with \( S_{ij} = S(x_i, x_j) \) satisfies the following condition:

4. **Metric**: \( S \) is positive semi-definite PSD (all eigenvalues are non-negative)
Different types of numerical data hence different kinds of proximity measures.
Continuous variables

- A family of dissimilarity measures is the **Minkowski distance**:

\[ D_{\text{mink}}^r(x, y) = \left( \sum_{j=1}^{p} |x_j - y_j|^r \right)^{1/r} \]
Continuous variables

- A family of dissimilarity measures is the **Minkowski distance**:

\[
D^r_{mink}(x, y) = \left( \sum_{j=1}^{p} |x_j - y_j|^r \right)^{1/r}
\]

- Famous particular cases of Minkowski distances are given by:
  - Euclidean distance: \( r = 2 \)
  - Manhattan distance: \( r = 1 \)
  - Max distance: \( r \rightarrow +\infty \)
Continuous variables

- A family of dissimilarity measures is the **Minkowski distance**:

  \[ D^{r \text{mink}}_{\text{mink}}(x, y) = \left( \sum_{j=1}^{p} |x_j - y_j|^r \right)^{1/r} \]

- Famous particular cases of Minkowski distances are given by:
  - Euclidean distance: \( r = 2 \)
  - Manhattan distance: \( r = 1 \)
  - Max distance: \( r \to +\infty \)

- Another famous distance is the **Mahalanobis measure**:

  \[ D_{\text{maha}}(x, y) = \sqrt{(x - y)^t \Sigma^{-1}(x - y)} \]

  where \( \Sigma \) is the covariance matrix
Continuous variables (cont’d)

- A family of similarity measures are the similarities of order $t$:

$$S^t(x, y) = \frac{\langle x, y \rangle}{\mathcal{M}^t(\langle x, x \rangle, \langle y, y \rangle)}$$

where $\mathcal{M}_t(a, b) = \left[\frac{1}{2}(a^t + b^t)\right]^{1/t}$ (generalized power mean)
Continuous variables (cont’d)

- A family of similarity measures are the similarities of order $t$:

$$S^t(x, y) = \frac{\langle x, y \rangle}{\mathcal{M}^t(\langle x, x \rangle, \langle y, y \rangle)}$$

where $\mathcal{M}_t(a, b) = \left[ \frac{1}{2} (a^t + b^t) \right]^{1/t}$ (generalized power mean).

- A famous particular case is given by the limit $t \to 0$:

$$S^{t \to 0}(x, y) = \cos(\theta(x, y))$$
Continuous variables (cont’d)

- A family of similarity measures are the **similarities of order** $t$:

$$S^t(x, y) = \frac{\langle x, y \rangle}{\mathcal{M}^t(\langle x, x \rangle, \langle y, y \rangle)}$$

where $\mathcal{M}_t(a, b) = \left[ \frac{1}{2} (a^t + b^t) \right]^{1/t}$ (generalized power mean)

- A famous particular case is given by the limit $t \to 0$:

$$S^{t \to 0}(x, y) = \cos(\theta(x, y))$$

- Why could the cosine measure be insufficient in some cases? Would you give the highest similarity value to the following vectors?

$x \quad\quad y$
Continuous variables (cont’d)

- A family of similarity measures are the similarities of order $t$:

$$
S^t(x, y) = \frac{\langle x, y \rangle}{M^t(\langle x, x \rangle, \langle y, y \rangle)}
$$

where $M_t(a, b) = \left[\frac{1}{2}(a^t + b^t)\right]^{1/t}$ (generalized power mean)

- A famous particular case is given by the limit $t \to 0$:

$$
S^{t \to 0}(x, y) = \cos(\theta(x, y))
$$

- Why could the cosine measure be insufficient in some cases? Would you give the highest similarity value to the following vectors?

- For $t > 0$, $S^t$ takes into account both the angular measure and the norms difference.
Continuous variables (cont’d)

- A family of similarity measures are the similarities of order $t$:

$$S^t(x, y) = \frac{\langle x, y \rangle}{M_t(\langle x, x \rangle, \langle y, y \rangle)}$$

where $M_t(a, b) = \left[ \frac{1}{2} (a^t + b^t) \right]^{1/t}$ (generalized power mean)

- A famous particular case is given by the limit $t \to 0$:

$$S^{t \to 0}(x, y) = \cos(\theta(x, y))$$

- Why could the cosine measure be insufficient in some cases? Would you give the highest similarity value to the following vectors?

- For $t > 0$, $S^t$ takes into account both the angular measure and the norms difference.

- Furthermore, for $t > 0$, the matrix $S$ of general term $S_{ij} = S^t(x_i, x_j)$ is PSD! (ie a metric).
Discrete variables

- Nominal variables (e.g., "color of eye") can be reduced to binary variables (presence/absence of each category).
Discrete variables

- Nominal variables (e.g., “color of eye”) can be reduced to binary variables (presence/absence of each category).
- For discrete variables, vectors are elements of \(\{0, 1\}^p\).
Discrete variables

- Nominal variables (e.g., “color of eye”) can be reduced to binary variables (presence/absence of each category).
- For discrete variables, vectors are elements of \( \{0, 1\}^p \).
- Let \( x, y \) in \( \{0, 1\}^p \), we can introduce the \((2 \times 2)\) contingency table:

<table>
<thead>
<tr>
<th></th>
<th>( y_j )</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_j )</td>
<td>1</td>
<td>( a )</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>( c )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>( x_j )</th>
<th>( y_j )</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>( y_j )</td>
<td>1</td>
<td>( a )</td>
<td>( a + b )</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>( c )</td>
<td>( c + d )</td>
</tr>
</tbody>
</table>

where:

- \( a = \sum_{j=1}^{p} x_j y_j \) = Nb. of shared labels
- \( b = \sum_{j=1}^{p} x_j (1 - y_j) \) = Nb. of labels \( x \) has but \( y \) hasn’t
- \( c = \sum_{j=1}^{p} (1 - x_j) y_j \) = Nb. of labels \( y \) has but \( x \) hasn’t
- \( d = \sum_{j=1}^{p} (1 - x_j)(1 - y_j) \) = Nb. of labels that neither \( x \) nor \( y \) has
Below are some similarity measures for binary vectors

- **Jaccard**: $S_{\text{jaccard}}(x, y) = \frac{a}{a+b+c} \in [0, 1]$

- **Dice**: $S_{\text{dice}}(x, y) = \frac{2a}{2a+b+c} \in [0, 1]$

- **Ochiai**: $S_{\text{ochiai}}(x, y) = \frac{a}{\sqrt{(a+b)(a+c)}} \in [0, 1]$

- **Kulczynski**: $S_{\text{kulczynski}}(x, y) = \frac{1}{2} \left( \frac{a}{a+b} + \frac{a}{a+c} \right) \in [0, 1]$

- **Sokal-Michener**: $S_{\text{soc-mich}}(x, y) = \frac{a+d}{a+b+c+d} \in [0, 1]$

- **Rogers-Tanimoto**: $S_{\text{rog-tan}}(x, y) = \frac{a+d}{a+2(b+c)+d} \in [0, 1]$

- **Phi**: $S_{\text{phi}}(x, y) = \frac{ad-bc}{\sqrt{(a+b)(c+d)(a+c)(b+d)}} \in [-1, 1]$
1. Qu’est ce que le clustering et à quoi ça peut servir ?

2. Différentes façons de classifier les éléments

3. Formalisation du problème du clustering

4. Quelques mesures de proximité

5. Quelques algorithmes classiques de clustering

6. Evaluation du clustering par Prof. Lallich
Different types of clustering algorithm

- HARD CLUSTERING
  - HIERARCHICAL
  - PARTITIONAL
- SOFT CLUSTERING
Rappel du Sommaire

1. Qu’est ce que le clustering et à quoi ça peut servir ?
2. Différentes façons de classifier les éléments
3. Formalisation du problème du clustering
4. Quelques mesures de proximité
5. Quelques algorithmes classiques de clustering
   - Classification hiérarchique
   - Partitionnement “dur”
   - Partitionnement “flou”
6. Evaluation du clustering par Prof. Lallich
Hierarchical clustering

- Two types of hierarchical clustering algorithms:
  - Agglomerative: “bottom-up”
  - Divisive: “top-down”
Pseudo-code of AHC

- Pseudo-code of agglomerative hierarchical clusterings (AHC):

1. **Input**: \( \mathbb{D} \)
2. Initialize the tree representation with \( n \) leaves
3. **While** not all data points are grouped together do
   4. Merge the two closest clusters according to a distance measure
   5. Add a parent node in the tree representation accordingly
4. **End While**
5. **Output**: tree representation
Pseudo-code of AHC

Pseudo-code of agglomerative hierarchical clusterings (AHC) :

1. **Input**: $D$
2. Initialize the tree representation with $n$ leaves
3. **While** not all data points are grouped together **do**
4. Merge the two closest clusters according to a distance measure
5. Add a parent node in the tree representation accordingly
6. **End While**
7. **Output**: tree representation

→ The critical point for AHC algorithms is the distance measure between clusters.
The Lance-Williams formula

Définition. (Lance-Williams formula)

It is a general recurrence equation used to calculate the dissimilarity between a cluster $C_k$ and a cluster newly formed by merging two other clusters $C_l \cup C_{l'}$ ($C_k$, $C_l$, $C_{l'}$ are 3 clusters belonging to the same level of the HC):

$$D_{LW}(C_k, C_l \cup C_{l'}) = \alpha_l D_{LW}(C_k, C_l) + \alpha_{l'} D_{LW}(C_k, C_{l'}) + \beta D_{LW}(C_l, C_{l'}) + \gamma |D_{LW}(C_k, C_l) - D_{LW}(C_k, C_{l'})|$$

where $\alpha_l, \alpha_{l'}, \beta, \gamma$ are real numbers.
### AHC methods and the Lance-Williams formula

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$\alpha_I$</th>
<th>$\alpha_{I'}$</th>
<th>$\beta$</th>
<th>$\gamma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single link</td>
<td>1/2</td>
<td>1/2</td>
<td>0</td>
<td>−1/2</td>
</tr>
<tr>
<td>Complete link</td>
<td>1/2</td>
<td>1/2</td>
<td>0</td>
<td>1/2</td>
</tr>
<tr>
<td>UPGMA</td>
<td>$\frac{</td>
<td>C_I</td>
<td>}{</td>
<td>C_I</td>
</tr>
<tr>
<td>WPGMA</td>
<td>1/2</td>
<td>1/2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Ward</td>
<td>$\frac{</td>
<td>C_I</td>
<td>+</td>
<td>C_k</td>
</tr>
<tr>
<td>Centroid</td>
<td>$\frac{</td>
<td>C_I</td>
<td>}{</td>
<td>C_I</td>
</tr>
<tr>
<td>Median</td>
<td>1/2</td>
<td>1/2</td>
<td>−1/4</td>
<td>0</td>
</tr>
</tbody>
</table>
Single link method

- One of the simplest AHC method proposed by Sneath in 1957
Single link method

- One of the simplest AHC method proposed by Sneath in 1957
- According to the LW formula we have:

\[
D_{sl}(C_k, C_l \cup C_{l'}) = \frac{1}{2} D_{sl}(C_k, C_l) + \frac{1}{2} D_{sl}(C_k, C_{l'})
- \frac{1}{2} |D_{sl}(C_k, C_l) - D_{sl}(C_k, C_{l'})|
\]

\[
= \min\{D_{sl}(C_k, C_l), D_{sl}(C_k, C_{l'})\}
\]
Quelques algorithmes classiques de clustering

Classification hiérarchique

Single link method

- One of the simplest AHC method proposed by Sneath in 1957
- According to the LW formula we have:

\[
D_{sl}(C_k, C_l \cup C_{l'}) = \frac{1}{2} D_{sl}(C_k, C_l) + \frac{1}{2} D_{sl}(C_k, C_{l'}) - \left| \frac{1}{2} (D_{sl}(C_k, C_l) - D_{sl}(C_k, C_{l'})) \right|
\]

\[
= \min\{D_{sl}(C_k, C_l), D_{sl}(C_k, C_{l'})\}
\]

- Let \( C_k \) and \( C_{k'} \) be two clusters, the recurrence formula amounts to:

\[
D_{sl}(C_k, C_{k'}) = \min_{x \in C_k, y \in C_{k'}} \{D(x, y)\}
\]
Single link method

- One of the simplest AHC method proposed by Sneath in 1957
- According to the LW formula we have:

\[ D_{sl}(C_k, C_l \cup C_{l'}) = \frac{1}{2} D_{sl}(C_k, C_l) + \frac{1}{2} D_{sl}(C_k, C_{l'}) - \frac{1}{2} |D_{sl}(C_k, C_l) - D_{sl}(C_k, C_{l'})| \]

\[ = \min\{D_{sl}(C_k, C_l), D_{sl}(C_k, C_{l'})\} \]

- Let \( C_k \) and \( C_{k'} \) be two clusters, the recurrence formula amounts to:

\[ D_{sl}(C_k, C_{k'}) = \min_{x \in C_k, y \in C_{k'}} \{D(x, y)\} \]
Example

We consider 5 data points in $\mathbb{R}^2$:

- $x_1 = (1, 2)$
- $x_2 = (1, 2.5)$
- $x_3 = (3, 1)$
- $x_4 = (4, 0.5)$
- $x_5 = (4, 2)$

We consider the euclidean distance between data points.
Example (cont’d)

The starting distance matrix $D$ is the euclidean distance matrix between points:

$$
D = D_{\text{eucl}} = D_{\text{sl}} = \begin{pmatrix}
    & x_1 & x_2 & x_3 & x_4 & x_5 \\
    x_1 & 0 & 0.5 & 2.24 & 3.35 & 3 \\
    x_2 & 0.5 & 0 & 2.5 & 3.61 & 3.04 \\
    x_3 & 2.24 & 2.5 & 0 & 1.12 & 1.41 \\
    x_4 & 3.35 & 3.61 & 1.12 & 0 & 1.5 \\
    x_5 & 3 & 3.04 & 1.41 & 1.5 & 0
\end{pmatrix}
$$
Example (cont’d)

The starting distance matrix $D$ is the euclidean distance matrix between points:

$$
D = D_{\text{eucl}} = D_{\text{sl}} =
\begin{pmatrix}
0 & 0.5 & 2.24 & 3.35 & 3 \\
0.5 & 0 & 2.5 & 3.61 & 3.04 \\
2.24 & 2.5 & 0 & 1.12 & 1.41 \\
3.35 & 3.61 & 1.12 & 0 & 1.5 \\
3 & 3.04 & 1.41 & 1.5 & 0
\end{pmatrix}
$$

2. $dend(h) = \{ \{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, \{x_5\} \} \quad \text{if } 0 \leq h$

4. Merge $x_1$ and $x_2$

$$
dend(h) = \begin{cases} 
\{ \{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, \{x_5\} \} & \text{if } 0 \leq h < 0.5 \\
\{x_1, x_2\}, \{x_3\}, \{x_4\}, \{x_5\} & \text{if } 0.5 \leq h 
\end{cases}
$$
Example (cont’d)

With the single link method, the distance matrix $D_{sl}$ becomes:

- $D_{sl}(\{x_1, x_2\}, x_3) = \min\{D_{sl}(x_1, x_3), D_{sl}(x_2, x_3)\} = 2.24$
- $D_{sl}(\{x_1, x_2\}, x_4) = \min\{D_{sl}(x_1, x_4), D_{sl}(x_2, x_4)\} = 3.35$
- $D_{sl}(\{x_1, x_2\}, x_5) = \min\{D_{sl}(x_1, x_5), D_{sl}(x_2, x_5)\} = 3$
Example (cont’d)

With the single link method, the distance matrix $D_{sl}$ becomes:

- $D_{sl}(\{x_1, x_2\}, x_3) = \min\{D_{sl}(x_1, x_3), D_{sl}(x_2, x_3)\} = 2.24$
- $D_{sl}(\{x_1, x_2\}, x_4) = \min\{D_{sl}(x_1, x_4), D_{sl}(x_2, x_4)\} = 3.35$
- $D_{sl}(\{x_1, x_2\}, x_5) = \min\{D_{sl}(x_1, x_5), D_{sl}(x_2, x_5)\} = 3$

\[
D_{sl} = \begin{pmatrix}
0 & 2.24 & 3.35 & 3 \\
2.24 & 0 & 1.12 & 1.41 \\
3.35 & 1.12 & 0 & 1.5 \\
3 & 1.41 & 1.5 & 0
\end{pmatrix}
\]
Example (cont’d)

With the single link method, the distance matrix $D_{sl}$ becomes:

- $D_{sl}([x_1, x_2], x_3) = \min\{D_{sl}(x_1, x_3), D_{sl}(x_2, x_3)\} = 2.24$
- $D_{sl}([x_1, x_2], x_4) = \min\{D_{sl}(x_1, x_4), D_{sl}(x_2, x_4)\} = 3.35$
- $D_{sl}([x_1, x_2], x_5) = \min\{D_{sl}(x_1, x_5), D_{sl}(x_2, x_5)\} = 3$

\[
D_{sl} = \begin{pmatrix}
  & x_3 & x_4 & x_5 \\
{x_1, x_2} & 0 & 2.24 & 3.35 & 3 \\
x_3 & 2.24 & 0 & 1.12 & 1.41 \\
x_4 & 3.35 & 1.12 & 0 & 1.5 \\
x_5 & 3 & 1.41 & 1.5 & 0 \\
\end{pmatrix}
\]

4. Merge $x_3$ and $x_4$

\[
dend(h) = \begin{cases}
  \{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, \{x_5\} & \text{if } 0 \leq h < 0.5 \\
  \{x_1, x_2\}, \{x_3\}, \{x_4\}, \{x_5\} & \text{if } 0.5 \leq h < 1.12 \\
  \{x_1, x_2\}, \{x_3, x_4\}, \{x_5\} & \text{if } 1.12 \leq h
\end{cases}
\]
Example (cont’d)

With the single link method, the distance matrix $D_{sl}$ becomes:

- $D_{sl}(\{x_3, x_4\}, \{x_1, x_2\}) = \min\{D_{sl}(x_3, x_1), D_{sl}(x_3, x_2), D_{sl}(x_4, x_1), D_{sl}(x_4, x_2)\} = 2.24$

- $D_{sl}(\{x_3, x_4\}, x_5) = \min\{D_{sl}(x_3, x_5), D_{sl}(x_4, x_5)\} = 1.41$
Example (cont’d)

With the single link method, the distance matrix $D_{sl}$ becomes:

- $D_{sl}(\{x_3, x_4\}, \{x_1, x_2\}) = \min\{D_{sl}(x_3, x_1), D_{sl}(x_3, x_2), D_{sl}(x_4, x_1), D_{sl}(x_4, x_2)\} = \min\{D_{sl}(x_3, \{x_1, x_2\}), D_{sl}(x_4, \{x_1, x_2\})\} = 2.24$

- $D_{sl}(\{x_3, x_4\}, x_5) = \min\{D_{sl}(x_3, x_5), D_{sl}(x_4, x_5)\} = 1.41$

\[
D_{sl} = \begin{pmatrix}
\{x_1, x_2\} & \{x_3, x_4\} & x_5 \\
\{x_1, x_2\} & 0 & 2.24 & 3 \\
\{x_3, x_4\} & 2.24 & 0 & 1.41 \\
x_5 & 3 & 1.41 & 0
\end{pmatrix}
\]
Example (cont’d)

With the single link method, the distance matrix $D_{sl}$ becomes:

- $D_{sl}(\{x_3, x_4\}, \{x_1, x_2\}) = \min\{D_{sl}(x_3, x_1), D_{sl}(x_3, x_2), D_{sl}(x_4, x_1), D_{sl}(x_4, x_2)\} = \min\{D_{sl}(x_3, \{x_1, x_2\}), D_{sl}(x_4, \{x_1, x_2\})\} = 2.24$

- $D_{sl}(\{x_3, x_4\}, x_5) = \min\{D_{sl}(x_3, x_5), D_{sl}(x_4, x_5)\} = 1.41$

$$D_{sl} = \begin{pmatrix}
\{x_1, x_2\} & \{x_3, x_4\} & x_5 \\
\{x_1, x_2\} & 0 & 2.24 & 3 \\
\{x_3, x_4\} & 2.24 & 0 & 1.41 \\
x_5 & 3 & 1.41 & 0
\end{pmatrix}$$

4. Merge $\{x_3, x_4\}$ and $x_5$

$$dend(h) = \begin{cases}
\{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, \{x_5\} & \text{if } 0 \leq h < 0.5 \\
\{x_1, x_2\}, \{x_3\}, \{x_4\}, \{x_5\} & \text{if } 0.5 \leq h < 1.12 \\
\{x_1, x_2\}, \{x_3, x_4\}, \{x_5\} & \text{if } 1.12 \leq h < 1.41 \\
\{x_1, x_2\}, \{x_3, x_4, x_5\} & \text{if } 1.41 \leq h
\end{cases}$$
Example (cont’d)

With the single link method, the distance matrix $D_{sl}$ becomes:

$$D_{sl}(\{x_3, x_4, x_5\}, \{x_1, x_2\}) = \min\{D_{sl}(\{x_3, x_4\}, \{x_1, x_2\}), D_{sl}(x_5, \{x_1, x_2\})\} = 2.24$$
Example (cont’d)

With the single link method, the distance matrix $D_{sl}$ becomes:

- $D_{sl}(\{x_3, x_4, x_5\}, \{x_1, x_2\}) = \min\{D_{sl}(\{x_3, x_4\}, \{x_1, x_2\}), D_{sl}(x_5, \{x_1, x_2\})\} = 2.24$

$$D_{sl} = \begin{pmatrix}
\{x_1, x_2\} & \{x_3, x_4, x_5\} \\
\{x_3, x_4, x_5\} & 0 & 2.24 \\
& 2.24 & 0
\end{pmatrix}$$
Example (cont’d)

With the single link method, the distance matrix $D_{sl}$ becomes:

$D_{sl}(\{x_3, x_4, x_5\}, \{x_1, x_2\}) = \min\{D_{sl}(\{x_3, x_4\}, \{x_1, x_2\}), D_{sl}(x_5, \{x_1, x_2\})\} = 2.24$

$D_{sl} = \begin{pmatrix}
{x_1, x_2} & \{x_3, x_4, x_5\} \\
{x_3, x_4, x_5} & \begin{pmatrix} 0 & 2.24 \\ 2.24 & 0 \end{pmatrix}
\end{pmatrix}$

4 Merge $\{x_3, x_4, x_5\}$ and $\{x_1, x_2\}$

$dend(h) = \begin{cases}
\{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, \{x_5\} & \text{if } 0 \leq h < 0.5 \\
\{x_1, x_2\}, \{x_3\}, \{x_4\}, \{x_5\} & \text{if } 0.5 \leq h < 1.12 \\
\{x_1, x_2\}, \{x_3, x_4\}, \{x_5\} & \text{if } 1.12 \leq h < 1.41 \\
\{x_1, x_2\}, \{x_3, x_4, x_5\} & \text{if } 1.41 \leq h < 2.24 \\
\{x_1, x_2, x_3, x_4, x_5\} & \text{if } 2.24 \leq h
\end{cases}$
Example (cont’d)

The dendrogram:

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

Levels:
- 0
- 0.5
- 1.12
- 1.41
- 2.24
- 3
Rappel du Sommaire

1. Qu’est ce que le clustering et à quoi ça peut servir ?
2. Différentes façons de classifier les éléments
3. Formalisation du problème du clustering
4. Quelques mesures de proxiimité
5. Quelques algorithmes classiques de clustering
   - Classification hiérarchique
   - Partitionnement “dur”
   - Partitionnement “flou”
6. Evaluation du clustering par Prof. Lallich
**k-means**

- First proposed by Forgy in 1965 and MacQueen in 1967
**k-means**

- First proposed by Forgy in 1965 and MacQueen in 1967
- The conventional $k$-means algorithm is described in Hartigan, 1975
**k-means**

- First proposed by Forgy in 1965 and MacQueen in 1967
- The conventional $k$-means algorithm is described in Hartigan, 1975
- The $k$ is the number of clusters which is a parameter to be set
**k-means**

- First proposed by Forgy in 1965 and MacQueen in 1967
- The conventional $k$-means algorithm is described in Hartigan, 1975
- The $k$ is the number of clusters which is a parameter to be set
- The conventional $k$-means algorithm is applied to a set of objects $\mathcal{D}$ described by continuous variables and it seeks to minimize the $SSE$ (Sum of Square Error) function:

\[
SSE(C) = \sum_{l=1}^{\lvert C \rvert} \sum_{x \in C_l} \| x - \mu(C_l) \|_2^2
\]

where $\mu(C_l) = \frac{1}{\lvert C_l \rvert} \sum_{x \in C_l} x$ is the mean vector of cluster $C_l$
The $k$-means algorithm is an **hill-climbing** optimization heuristics which finds a local minimum of $SSE$.
The $k$-means algorithm is an **hill-climbing** optimization heuristics which finds a local minimum of $SSE$

1. **Input**: $X$ and $k$
2. Initialize $C$ with $k$ different clusters
3. **While** a stopping criterion is not reached **do**
   
4. **For all** $x \in \mathbb{D}$ **do**
   
5. **For all** $C_l \in C$ **do**
   
6. **Compute** $D^2_{eucl}(x, \mu(C_l))$

7. **End For**

8. **Find** $C_{l*} = \arg \min_{C_l \in C} \{D^2_{eucl}(x, \mu(C_l))\}$

9. Move $x$ from its current cluster to $C_{l*}$

10. Update the mean vectors accordingly

11. **End For**

12. **End While**

13. **Output**: $C$
The conventional $k$-means algorithm can be divided into 2 phases:

1. **Initialization phase**: the algorithm randomly assigns objects of $D$ to $k$ clusters.
2. **Iteration phase**: re-assignment of each object to the nearest cluster (represented by mean vectors), using the euclidean distance.
**k-means (cont’d)**

- The conventional $k$-means algorithm can be divided into 2 phases:
  1. The **initialization phase**: the algorithm randomly assigned objects of $\mathcal{D}$ to $k$ clusters.
  2. The **iteration phase**: re-assignment of each object to the nearest cluster (represented by mean vectors), using the euclidean distance.

- The algorithm stops when:
  - A maximal number of iterations is reached
  - The $SSE$ value does not change significantly
  - The clusters do not change any longer
Example

- **Data table (matrix of objects):**

\[
X = \begin{pmatrix}
x_1 & 1 & 1 \\
x_2 & 1.5 & 2 \\
x_3 & 3 & 4 \\
x_4 & 5 & 7 \\
x_5 & 3.5 & 5 \\
x_6 & 4.5 & 5 \\
x_7 & 3.5 & 4.5 \\
\end{pmatrix}
\]

- **Initialization of Q (matrix of mean vectors):**

\[
q_1 \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad q_2 \begin{pmatrix} 5 \\ 7 \end{pmatrix}
\]
Example (cont’d)

5-7 \[ D^2_{eucl}(x_1, q_1) = 0 \]
\[ D^2_{eucl}(x_1, q_2) = 52 \]

8-10 \[ C_1 = \{ x_1 \} \]
\[ q_1 = x_1 \]
\[ = (1, 1) \]
Example (cont’d)

5-7  \[ D^2_{eucl}(x_1, q_1) = 0 \]
     \[ D^2_{eucl}(x_1, q_2) = 52 \]

8-10  \( C_1 = \{x_1\} \)
       \[ q_1 = x_1 \]
       \[ = (1, 1) \]

5-7  \[ D^2_{eucl}(x_2, q_1) = 1.25 \]
     \[ D^2_{eucl}(x_2, q_2) = 37.25 \]

8-10  \( C_1 = \{x_1, x_2\} \)
       \[ q_1 = \frac{x_1 + x_2}{2} \]
       \[ = (1.25, 1.5) \]
Example (cont’d)

5-7 \[ D^2_{eucl}(x_1, q_1) = 0 \]
\[ D^2_{eucl}(x_1, q_2) = 52 \]

8-10 \[ C_1 = \{x_1\} \]
\[ q_1 = x_1 = (1, 1) \]

5-7 \[ D^2_{eucl}(x_2, q_1) = 1.25 \]
\[ D^2_{eucl}(x_2, q_2) = 37.25 \]

8-10 \[ C_1 = \{x_1, x_2\} \]
\[ q_1 = \frac{x_1 + x_2}{2} = (1.25, 1.5) \]

5-7 \[ D^2_{eucl}(x_3, q_1) = 9.31 \]
\[ D^2_{eucl}(x_3, q_2) = 13 \]

8-10 \[ C_1 = \{x_1, x_2, x_3\} \]
\[ q_1 = \frac{x_1 + x_2 + x_3}{3} = (1.83, 2.33) \]
Example (cont’d)

5-7 \[ D^2_{eucl}(x_4, q_1) = 31.80 \]
\[ D^2_{eucl}(x_4, q_2) = 0 \]
8-10 \[ C_2 = \{x_4\} \]
\[ q_2 = x_4 = (5, 7) \]
Example (cont’d)

5-7 \[ D^2_{eucl}(x_4, q_1) = 31.80 \]
\[ D^2_{eucl}(x_4, q_2) = 0 \]

8-10 \[ C_2 = \{x_4\} \]
\[ q_2 = x_4 \]
\[ = (5, 7) \]

5-7 \[ D^2_{eucl}(x_5, q_1) = 9.89 \]
\[ D^2_{eucl}(x_5, q_2) = 6.25 \]

8-10 \[ C_2 = \{x_4, x_5\} \]
\[ q_2 = \frac{x_4 + x_5}{2} \]
\[ = (4.25, 6) \]
Example (cont’d)

5-7  \[ D^2_{eucl}(x_4, q_1) = 31.80 \]
\[ D^2_{eucl}(x_4, q_2) = 0 \]

8-10  \[ C_2 = \{ x_4 \} \]
\[ q_2 = x_4 \]
\[ = (5, 7) \]

5-7  \[ D^2_{eucl}(x_5, q_1) = 9.89 \]
\[ D^2_{eucl}(x_5, q_2) = 6.25 \]

8-10  \[ C_2 = \{ x_4, x_5 \} \]
\[ q_2 = \frac{x_4 + x_5}{2} \]
\[ = (4.25, 6) \]

5-7  \[ D^2_{eucl}(x_6, q_1) = 14.22 \]
\[ D^2_{eucl}(x_6, q_2) = 1.06 \]

8-10  \[ C_2 = \{ x_4, x_5, x_6 \} \]
\[ q_2 = \frac{x_4 + x_5 + x_6}{3} \]
\[ = (4.33, 5.67) \]
Example (cont’d)

5-7  \[ D_{eucl}^2(x_7, q_1) = 7.47 \]
\[ D_{eucl}^2(x_7, q_2) = 2.05 \]

8-10  \[ C_2 = \{x_4, x_5, x_6, x_7\} \]
\[ q_2 = \frac{\sum_{i=4}^{7} x_i}{4} \]
\[ = (4.12, 5.37) \]
Example (cont’d)

5-7 \[ D_{eucl}^2(x_7, q_1) = 7.47 \]
\[ D_{eucl}^2(x_7, q_2) = 2.05 \]

8-10 \[ C_2 = \{x_4, x_5, x_6, x_7\} \]
\[ q_2 = \frac{\sum_{i=4}^{7} x_i}{4} \]
\[ = (4.12, 5.37) \]

At the end of the first scan, we have the two clusters:
\[ \{\{x_1, x_2, x_3\}, \{x_4, x_5, x_6, x_7\}\}. \]
Example (cont’d)

5-7 \[ D_{eucl}^2(x_7, q_1) = 7.47 \]
\[ D_{eucl}^2(x_7, q_2) = 2.05 \]

8-10 \[ C_2 = \{x_4, x_5, x_6, x_7\} \]
\[ q_2 = \frac{\sum_{i=4}^{7} x_i}{4} \]
\[ = (4.12, 5.37) \]

At the end of the first scan, we have the two clusters:
\[ \{\{x_1, x_2, x_3\}, \{x_4, x_5, x_6, x_7\}\} \]

We start over from line 3 in the k-means algorithm . . .
Another view of $k$-means

- We can cast the $SSE$ objective function as follows:

$$SSE(U, Q) = \sum_{q_l \in Q} \sum_{x_i \in D} u_{il} D_{eucl}^2(x_i, q_l)$$

where:

- $Q = \{q_1, \ldots, q_k\}$ is the set of cluster prototypes which are points of the input space ($\mathbb{R}^p$)
- $Q$ is the ($k \times p$) matrix whose rows are the cluster prototypes coordinates
- $U$ is an assignment matrix whith general term $u_{il} = 1$ if $x_i$ is assigned to cluster $C_l$ and such that:
  
  1. $\forall i = 1, \ldots, n; \forall l = 1, \ldots, k : u_{il} \in \{0, 1\}$
  2. $\forall i = 1, \ldots, n : \sum_{l=1}^{k} u_{il} = 1$
Another view of $k$-means

- We can cast the $SSE$ objective function as follows:

$$SSE(U, Q) = \sum_{q_l \in Q} \sum_{x_i \in D} u_{il} D_{eucl}^2(x_i, q_l)$$

where:
- $Q = \{q_1, \ldots, q_k\}$ is the set of cluster prototypes which are points of the input space ($\mathbb{R}^p$)
- $Q$ is the $(k \times p)$ matrix whose rows are the cluster prototypes coordinates
- $U$ is an assignment matrix with general term $u_{il} = 1$ if $x_i$ is assigned to cluster $C_l$ and such that:
  1. $\forall i = 1, \ldots, n; \forall l = 1, \ldots, k : u_{il} \in \{0, 1\}$
  2. $\forall i = 1, \ldots, n : \sum_{l=1}^{k} u_{il} = 1$

- We have to minimize $SSE(U, Q)$ with respect to $U$ and $Q$ under the constraints (1) and (2).
Another view of \( k \)-means (cont’d)

- One can solve the optimization problem approximatively by iteratively considering the two following subproblems (alternating minimization approach):
  1. Set \( Q = \hat{Q} \) and solve the reduced problem \( SSE(U, \hat{Q}) \)
  2. Set \( U = \hat{U} \) and solve the reduced problem \( SSE(\hat{U}, Q) \)
Another view of $k$-means (cont’d)

- One can solve the optimization problem approximatively by iteratively considering the two following subproblems (alternating minimization approach):
  1. Set $Q = \hat{Q}$ and solve the reduced problem $SSE(U, \hat{Q})$
  2. Set $U = \hat{U}$ and solve the reduced problem $SSE(\hat{U}, Q)$

- Since the sequence of $SSE$ is strictly decreasing the previous algorithm will converge to a local minimum after a finite number of iterations.
Another view of \( k \)-means (cont’d)

- One can solve the optimization problem approximatively by iteratively considering the two following subproblems (\textit{alternating minimization approach}):
  1. Set \( Q = \hat{Q} \) and solve the reduced problem \( SSE(U, \hat{Q}) \)
  2. Set \( U = \hat{U} \) and solve the reduced problem \( SSE(\hat{U}, Q) \)

- Since the sequence of \( SSE \) is strictly decreasing the previous algorithm will converge to a local minimum after a finite number of iterations.

- In the case of \( SSE \) using the euclidean distance and a hard assignment matrix, we can solve these two subproblems efficiently.
Another view of $k$-means (cont’d)

- In subproblem 1, $\hat{Q}$ is constant and $SSE(U, \hat{Q})$ is minimized iff:

$$\forall i : 1, \ldots, n : u_{il} = \begin{cases} 
1 & \text{if } D^2_{eucl}(x_i, \hat{q}_l) = \min_{\hat{q}_l' \in \hat{Q}} \left\{ D^2_{eucl}(x_i, \hat{q}_l') \right\} \\
0 & \text{otherwise}
\end{cases}$$
Another view of $k$-means (cont’d)

- In subproblem 1, $\hat{Q}$ is constant and $SSE(U, \hat{Q})$ is minimized iff:
  \[
  \forall i : 1, \ldots, n : u_{il} = \begin{cases} 
  1 & \text{if } D_{eucl}^2(x_i, \hat{q}_l) = \min_{q_l' \in \hat{Q}} \{ D_{eucl}^2(x_i, \hat{q}_{l'}) \} \\
  0 & \text{otherwise}
  \end{cases}
  \]

- In subproblem 2, $\hat{U}$ is constant and $SSE(\hat{U}, Q)$ is minimized iff:
  \[
  \forall l : 1, \ldots, k : \hat{q}_l = \frac{1}{\sum_{i=1}^{n} \hat{u}_{il}} \sum_{i=1}^{n} \hat{u}_{il} x_i
  \]
Another view of $k$-means (cont’d)

- In subproblem 1, $\hat{Q}$ is constant and $SSE(U, \hat{Q})$ is minimized iff:

$$\forall i : 1, \ldots, n : u_{il} = \begin{cases} 1 & \text{if } D^2_{eucl}(x_i, \hat{q}_l) = \min_{\hat{q}_l' \in \hat{Q}} \{D^2_{eucl}(x_i, \hat{q}_l')\} \\ 0 & \text{otherwise} \end{cases}$$

- In subproblem 2, $\hat{U}$ is constant and $SSE(\hat{U}, Q)$ is minimized iff:

$$\forall l : 1, \ldots, k : \hat{q}_l = \frac{1}{\sum_{i=1}^{n} \hat{u}_{il}} \sum_{i=1}^{n} \hat{u}_{il} x_i$$

- Both approaches minimize $SSE$ approximatively but the conventional $k$-means algorithm updates both $U$ and $Q$ after each data point reallocation.
Rappel du Sommaire

1. Qu’est ce que le clustering et à quoi ça peut servir ?
2. Différentes façons de classifier les éléments
3. Formalisation du problème du clustering
4. Quelques mesures de proximité
5. Quelques algorithmes classiques de clustering
   - Classification hiérarchique
   - Partitionnement “dur”
   - Partitionnement “flou”
6. Evaluation du clustering par Prof. Lallich
Quelques algorithmes classiques de clustering

Partitionnement “flou”

Fuzzy k-means (also know as fuzzy c-means)

- Proposed by Bezdek in 1973
Fuzzy $k$-means (also known as fuzzy $c$-means)

- Proposed by Bezdek in 1973
- The objective function is the following one:

$$E(U, Q) = \sum_{q_l \in Q} \sum_{x_i \in D} u_{il}^{\alpha} D_{eucl}^2(x_i, q_l)$$

where:

- $Q$ is again the set of cluster prototypes belonging to the input space
- BUT $U$ is a fuzzy assignment matrix where $u_{il}$ is the membership value of object $x_i$ to cluster $C_l$ and such that:
  
  $$(1) \quad \forall i = 1, \ldots, n; \forall l = 1, \ldots, k : u_{il} \in [0, 1]$$
  
  $$(2) \quad \forall i = 1, \ldots, n : \sum_{l=1}^{k} u_{il} = 1$$
  
  $$(3) \quad \forall l = 1, \ldots, k : \sum_{i=1}^{n} u_{il} > 0$$

- $\alpha$ is called the fuzzifier. Typically $\alpha = 2$ (setting $\alpha = 1$ leads to the crisp solution).
Fuzzy \( k \)-means (cont’d)

For \( \alpha > 1 \), Bezdek gave the two following necessary conditions for a minimum \((U^*, Q^*)\) of \( E(U, Q) \).

- Regarding \( Q^* \):
  
  \[
  \forall l = 1, \ldots, k : q^*_i = \frac{\sum_{x_i \in S} (u^*_i)^{\alpha} x}{\sum_{x_i \in S} (u^*_i)^{\alpha}}
  \]  
  \( (1) \)
Fuzzy $k$-means (cont’d)

- For $\alpha > 1$, Bezdek gave the two following necessary conditions for a minimum $(U^*, Q^*)$ of $E(U, Q)$.
  - Regarding $Q^*$:
    \[
    \forall l = 1, \ldots, k : q^*_l = \frac{\sum_{x_i \in S(u^*_il)}^\alpha x}{\sum_{x_i \in S(u^*_il)}^\alpha} \quad (1)
    \]
  - Regarding $U^*$:
    - If $\forall l : D_{eucl}^2(x_i, q^*_l) > 0$ then we have:
      \[
      \forall l : u^*_{il} = \frac{(D_{eucl}^2(x_i, q^*_l))^{-1}}{\sum_{q^*_i \in Q^*} (D_{eucl}^2(x_i, q^*_l))^{-\frac{1}{\alpha-1}}} \quad (2)
      \]
    - If $\exists l : D_{eucl}^2(x_i, q^*_l) = 0$ then $u^*_{il}$ are any non negative numbers such that $\sum_{l=1}^k u^*_{il} = 1$ and $u^*_{il} = 0$ if $D_{eucl}^2(x_i, q^*_l) > 0$. 
Fuzzy \(k\)-means (cont’d)

1. **Input**: \(X, k, \alpha\)
2. Initialize \(k\) different clusters
Fuzzy $k$-means (cont'd)

1. **Input**: $X$, $k$, $\alpha$
2. Initialize $k$ different clusters
3. **While** a stopping criterion is not reached **do**
   4. **For all** $i = 1, \ldots, k$ **do**
   5. Compute $q_i$ using Eq. (1)
   6. **End For**
Fuzzy $k$-means (cont’d)

1. **Input**: $X$, $k$, $\alpha$
2. Initialize $k$ different clusters
3. **While** a stopping criterion is not reached do
4. 
   For all $l = 1, \ldots, k$ do
5. 
   Compute $q_l$ using Eq. (1)
6. 
   End For
7. 
   For all $x_i \in D$ do
8. 
   For all $l = 1, \ldots, k$ do
9. 
   Compute $u_{il}$ using Eq. (2)
10. 
   End For
11. 
   End For
12. End While
13. **Output**: $U$ and $Q$
Model based clustering or density mixture models

- Clustering algorithms based upon probability models.
Model based clustering or density mixture models

- Clustering algorithms based upon probability models.
- Data are viewed as coming from a finite mixture of probability distributions.
Model based clustering or density mixture models

- Clustering algorithms based upon probability models.
- Data are viewed as coming from a finite mixture of probability distributions.
- Each distribution represents a cluster.
Model based clustering or density mixture models

- Clustering algorithms based upon probability models.
- Data are viewed as coming from a finite mixture of probability distributions.
- Each distribution represents a cluster.
- The clustering problem becomes that of estimating the parameters of the assumed mixture.
Model based clustering or density mixture models

- Clustering algorithms based upon probability models.
- Data are viewed as coming from a finite mixture of probability distributions.
- Each distribution represents a cluster.
- The clustering problem becomes that of estimating the parameters of the assumed mixture.
- Once the parameters of the model are estimated, we can compute the posterior probabilities of cluster membership of the objects.
Model based clustering or density mixture models (cont’d)

- Finite mixture models are probability density functions of the form:

\[
f(x; p, \theta) = \sum_{l=1}^{k} p_l g_l(x; \theta_l)
\]

where:

- \( x \) is a \( p \)-dimensional random variable
- \( p = (p_1, \ldots, p_k) \) is the vector of mixing proportions such that \( \sum_{l=1}^{k} p_l = 1 \)
- \( g_l; l = 1, \ldots, k \) are the different component densities of the mixture
- Each \( g_l \) is parametrized by a vector of parameters \( \theta_l \)
Finite mixture models are probability density functions of the form:

\[
f(x; p, \theta) = \sum_{l=1}^{k} p_l g_l(x; \theta_l)
\]

where:

- \(x\) is a \(p\)-dimensional random variable
- \(p = (p_1, \ldots, p_k)\) is the vector of mixing proportions such that \(\sum_{l=1}^{k} p_l = 1\)
- \(g_l; l = 1, \ldots, k\) are the different component densities of the mixture
- Each \(g_l\) is parametrized by a vector of parameters \(\theta_l\)

Note that all \(g_l\) can be of the same density family but they differ from their parameters.
Model based clustering or density mixture models (cont’d)

If we have the estimation of all the parameters of the models (ie \( \hat{p} \) and the \( \hat{\theta}_l \)) then we can deduce:

\[
\Pr(C_l|x_i) = \frac{\hat{p}_l g_l(x_i; \hat{\theta}_l)}{f(x_i; \hat{p}, \hat{\theta})}
\]
If we have the estimation of all the parameters of the models (ie \( \hat{p} \) and the \( \hat{\theta}_l \)) then we can deduce:

\[
\Pr(C_l|\mathbf{x}_i) = \frac{\hat{p}_l g_l(\mathbf{x}_i; \hat{\theta}_l)}{f(\mathbf{x}_i; \hat{p}, \hat{\theta})}
\]

\( \Pr(C_l|\mathbf{x}_i) \) is the posterior probability of having \( C_l \) given \( \mathbf{x}_i \) and it represents the membership value of \( \mathbf{x}_i \) to cluster \( C_l \).
Model based clustering or density mixture models (cont’d)

- If we have the estimation of all the parameters of the models (ie \( \hat{p} \) and the \( \hat{\theta}_l \)) then we can deduce:

  \[
  \text{Pr}(C_l|x_i) = \frac{\hat{p}_l g_l(x_i; \hat{\theta}_l)}{f(x_i; \hat{p}, \hat{\theta})}
  \]

- \( \text{Pr}(C_l|x_i) \) is the posterior probability of having \( C_l \) given \( x_i \) and it represents the membership value of \( x_i \) to cluster \( C_l \).

- Suppose now that we are given \( D = \{x_1, \ldots, x_n\} \). We thus have the following log-likelihood function \( l \) :

  \[
  l(p, \theta) = \sum_{x_i \in D} \ln f(x_i; p, \theta)
  \]
If we have the estimation of all the parameters of the models (ie \( \hat{p} \) and the \( \hat{\theta}_l \)) then we can deduce:

\[
\Pr(C_l | x_i) = \frac{\hat{p}_l g_l(x_i; \hat{\theta}_l)}{f(x_i; \hat{p}, \hat{\theta})}
\]

\( \Pr(C_l | x_i) \) is the posterior probability of having \( C_l \) given \( x_i \) and it represents the membership value of \( x_i \) to cluster \( C_l \).

Suppose now that we are given \( D = \{x_1, \ldots, x_n\} \). We thus have the following log-likelihood function \( l \) :

\[
l(p, \theta) = \sum_{x_i \in D} \ln f(x_i; p, \theta)
\]

The log-likelihood function is too complicated to employ the usual methods for its maximization (null gradient).
Model based clustering or density mixture models (cont’d)

- If we have the estimation of all the parameters of the models (ie \( \hat{p} \) and the \( \hat{\theta}_l \)) then we can deduce :

\[
Pr(C_l|x_i) = \frac{\hat{p}_l g_l(x_i; \hat{\theta}_l)}{f(x_i; \hat{p}, \hat{\theta})}
\]

- \( Pr(C_l|x_i) \) is the posterior probability of having \( C_l \) given \( x_i \) and it represents the membership value of \( x_i \) to cluster \( C_l \).

- Suppose now that we are given \( D = \{x_1, \ldots, x_n\} \). We thus have the following log-likelihood function \( l \) :

\[
l(p, \theta) = \sum_{x_i \in D} \ln f(x_i; p, \theta)
\]

- The log-likelihood function is too complicated to employ the usual methods for its maximization (null gradient).

→ To estimate the parameters, the most widely used approach is the iterative expectation - maximization (EM) algorithm.
Gaussians mixtures and EM algorithm

- Gaussians mixture is one of the most used mixture model:

\[
f(x; p, \mu, \Sigma) = \sum_{l=1}^{k} p_l \Phi(x; \mu_l, \Sigma_l) \]

in that case:

- \( \forall l : g_l(x; \theta_l) = \Phi(x; \mu_l, \Sigma_l) = \exp\left[-\frac{1}{2} (x-\mu_l)^T \Sigma_l^{-1} (x-\mu_l)\right] \sqrt{(2\pi)^p |\Sigma_l|} \)
- \( \forall l : \mu_l \) is the mean vector related to \( C_l \)
- \( \forall l : \Sigma_l \) is the covariance matrix related to \( C_l \)
In the case of Gaussians mixtures, the EM algorithm iteratively updates the following quantities:

- **Expectation step**:

  \[
  \Pr(C_l|x_i) = \frac{\hat{p}_l \Phi(x_i; \hat{\mu}_l, \hat{\Sigma}_l)}{\sum_{l=1}^k p_l \Phi(x_i; \hat{\mu}_l, \hat{\Sigma}_l)}
  \]

- **Maximization step**:

  \[
  \hat{p}_l = \frac{1}{n} \sum_{x_i \in D} \Pr(C_l|x_i)
  \]

  \[
  \hat{\mu}_l = \frac{1}{n \hat{p}_l} \sum_{x_i \in D} x_i \Pr(C_l|x_i)
  \]

  \[
  \hat{\Sigma}_l = \frac{1}{n} \sum_{x_i \in D} (x_i - \mu_l)(x_i - \mu_l)^t \Pr(C_l|x_i)
  \]
Gaussians mixtures and EM algorithm (cont’d)

1. **Input**: $X$, $k$
2. Initialize the means $\mu_l$, covariances $\Sigma_l$ and mixing coefficients $p_l$
Gaussians mixtures and EM algorithm (cont’d)

1. **Input**: $X, k$
2. Initialize the means $\mu_l$, covariances $\Sigma_l$ and mixing coefficients $p_l$
3. **While** a stopping criterion is not reached do
   4. **For all** $i = 1, \ldots, n$ do (E step)
      5. **For all** $l = 1, \ldots, k$ do
         6. Compute $\Pr(C_l|x_i)$ using Eq. (3)
       7. End For
   8. End For
10. End While
11. **Output**: $U$ and $Q$
Quelques algorithmes classiques de clustering

Partitionnement “flou”

Gaussians mixtures and EM algorithm (cont’d)

1. **Input**: $X$, $k$
2. Initialize the means $\mu_l$, covariances $\Sigma_l$ and mixing coefficients $p_l$
3. **While** a stopping criterion is not reached **do**
4. **For all** $i = 1, \ldots, n$ **do** (E step)
5. **For all** $l = 1, \ldots, k$ **do**
6. Compute $\Pr(C_l|x_i)$ using Eq. (3)
7. **End For**
8. **End For**
4. **For all** $l = 1, \ldots, k$ **do** (M step)
8. Compute $\hat{p}_l$ using Eq. (4)
8. Compute $\hat{\mu}_l$ using Eq. (5)
8. Compute $\hat{\Sigma}_l$ using Eq. (6)
11. **End For**
12. **End While**
13. **Output**: $U$ and $Q$
Gaussians mixtures - EM algorithm and \( k \)-means

- EM algorithm for Gaussians mixtures and \( k \)-means are related.
Gaussians mixtures - EM algorithm and $k$-means

- EM algorithm for Gaussians mixtures and $k$-means are related.
- The former can be viewed as a soft version of the latter technique.
Gaussians mixtures - EM algorithm and $k$-means

- EM algorithm for Gaussians mixtures and $k$-means are related.
- The former can be viewed as a soft version of the latter technique.
- $k$-means is a particular limit of EM for Gaussians mixtures.
Gaussians mixtures - EM algorithm and $k$-means

- EM algorithm for Gaussians mixtures and $k$-means are related.
- The former can be viewed as a soft version of the latter technique.
- $k$-means is a particular limit of EM for Gaussians mixtures.
- If we assume that $\Sigma_l = \epsilon l$ for all $l = 1, \ldots, k$, we obtain:

\[
\Phi(x; \mu_l, \Sigma_l) = \frac{\exp\left[-\frac{1}{2\epsilon}\|x - \mu_l\|^2\right]}{\sqrt{(2\pi)^p\epsilon}}
\]

\[
\Pr(C_l|x_i) = \frac{\hat{p}_l \exp\left[-\frac{1}{2\epsilon}\|x - \mu_l\|^2\right]}{\sum_{l=1}^k \hat{p}_l \exp\left[-\frac{1}{2\epsilon}\|x - \mu_l\|^2\right]}
\]
Gaussians mixtures - EM algorithm and $k$-means

- EM algorithm for Gaussians mixtures and $k$-means are related.
- The former can be viewed as a soft version of the latter technique.
- $k$-means is a particular limit of EM for Gaussians mixtures.
- If we assume that $\Sigma_l = \epsilon I$ for all $l = 1, \ldots, k$, we obtain:

$$
\Phi(x; \mu_l, \Sigma_l) = \frac{\exp\left[-\frac{1}{2\epsilon} \|x - \mu_l\|^2\right]}{\sqrt{(2\pi)^p \epsilon}}
$$

$$
\Pr(C_l|\mathbf{x}_i) = \frac{\hat{p}_l \exp\left[-\frac{1}{2\epsilon} \|x - \mu_l\|^2\right]}{\sum_{l=1}^k \hat{p}_l \exp\left[-\frac{1}{2\epsilon} \|x - \mu_l\|^2\right]}
$$

- If $\epsilon \to 0$ then:
  - $\Pr(C_l|\mathbf{x}_i)$ converges to 0 except for $C_{l^*} = \arg \min_{C_l \in C} \{\|\mathbf{x}_i - \mu_l\|^2\}$
  - The expected log likelihood tends to:

$$
-\frac{1}{2} \sum_{i=1}^n \sum_{l=1}^k u_{il} \|x_i - \mu_l\|^2 + \text{const}
$$
Quelques algorithmes classiques de clustering

Partitionnement “flou”

Somme references I

Ah-Pine, J. (2010)
Normalized kernels as similarity indices.
In : PAKDD’10 : Proceedings of the 14th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Hyderabad, India (2010)

Pattern recognition and machine learning.


Maximum Likelihood from Incomplete Data via the EM Algorithm.


Metric and euclidean properties of dissimilarity coefficients.
Somme references II

Morgan Kaufmann, 2 edition.


Algorithms for hierarchical clustering : an overview.
Wiley Interdisciplinary Reviews : Data Mining and Knowledge Discovery, 2(1) :86–97.

Similarity measures.

Survey of clustering algorithms.
IEEE Transactions on Neural Networks, 16(3) :645–678.
Rappel du Sommaire

1 Qu’est ce que le clustering et à quoi ça peut servir ?

2 Différentes façons de classifier les éléments

3 Formalisation du problème du clustering

4 Quelques mesures de proximité

5 Quelques algorithmes classiques de clustering

6 Evaluation du clustering par Prof. Lallich