Overview of Clustering

based on Loïc Cerfs slides (UFMG)

Marc Plantevit

March 11, 2021
UCBL – LIRIS – DM2L

Example of applicative problem

Student profiles

Given the marks received by students for different courses, how to group the students so that two students in a same group received about the same marks for each course and two students in different groups have different profiles.

. . . many other applications: marketing (user segmentation), ecology (identification of similar zone), insurance, urban planification, Health (tumor identification etc.), social network analysis, . . .

Outline

Outline

An optimization problem

Definition

Partitioning the objects so that each partition contains *similar* objects and objects in different partitions are *dissimilar*.

Input:

	a_1	a_2	$\dots$	a_n
o_1	$d_{1,1}$	$d_{1,2}$	$\dots$	$d_{1,n}$
o_2	$d_{2,1}$	$d_{2,2}$	$\dots$	$d_{2,n}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
o_m	$d_{m,1}$	$d_{m,2}$	$\dots$	$d_{m,n}$

An optimization problem

Definition

Partitioning the objects so that the intra-cluster *similarities* are maximized and the inter-cluster *similarities* are minimized.

Input:

	a_1	a_2	$\dots$	a_n
o_1	$d_{1,1}$	$d_{1,2}$	$\dots$	$d_{1,n}$
o_2	$d_{2,1}$	$d_{2,2}$	$\dots$	$d_{2,n}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
o_m	$d_{m,1}$	$d_{m,2}$	$\dots$	$d_{m,n}$

An optimization problem

Definition

Partitioning the objects so that the intra-cluster *similarities* are maximized and the inter-cluster *similarities* are minimized.

Output:

	a_1	a_2	$\dots$	a_n	cluster
o_1	$d_{1,1}$	$d_{1,2}$	$\dots$	$d_{1,n}$	c_1
o_2	$d_{2,1}$	$d_{2,2}$	$\dots$	$d_{2,n}$	c_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
o_m	$d_{m,1}$	$d_{m,2}$	$\dots$	$d_{m,n}$	c_1

An optimization problem

Definition

Partitioning the objects so that the intra-cluster *similarities* are maximized and the inter-cluster *similarities* are minimized.

Output:

	a_1	a_2	$\ldots$	a_n	cluster
o_1	$d_{1,1}$	$d_{1,2}$	$\ldots$	$d_{1,n}$	c_1
o_2	$d_{2,1}$	$d_{2,2}$	$\ldots$	$d_{2,n}$	c_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
o_m	$d_{m,1}$	$d_{m,2}$	$\ldots$	$d_{m,n}$	c_1

The number of clusters can be a parameter of the algorithm or has to be found.

Illustration

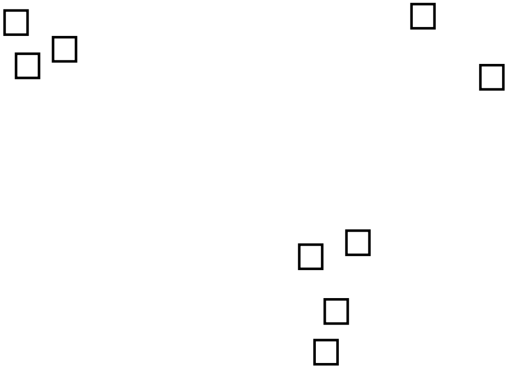
Clustering objects in a two-dimensional space using the Euclidean distance (the greater, the less similar).

	x	y
o_1	91	70
o_2	129	91
o_3	359	243
o_4	322	254
o_5	100	104
o_6	464	113
o_7	342	297
o_8	410	65
o_9	334	329
$\vdots$	$\vdots$	$\vdots$

Illustration

Clustering objects in a two-dimensional space using the Euclidean distance (the greater, the less similar).

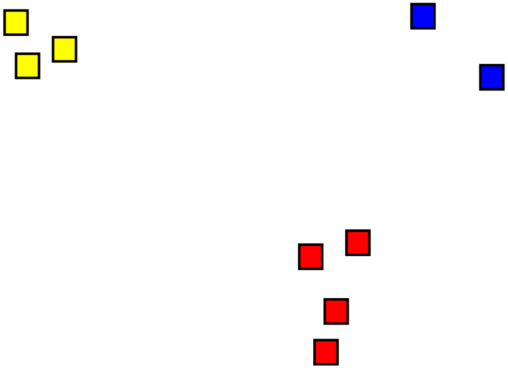
	x	y
o_1	91	70
o_2	129	91
o_3	359	243
o_4	322	254
o_5	100	104
o_6	464	113
o_7	342	297
o_8	410	65
o_9	334	329
⋮	⋮	⋮



Illustration

Clustering objects in a two-dimensional space using the Euclidean distance (the greater, the less similar).

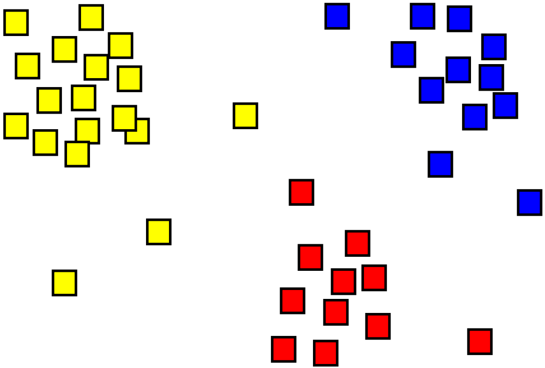
	x	y
o_1	91	70
o_2	129	91
o_3	359	243
o_4	322	254
o_5	100	104
o_6	464	113
o_7	342	297
o_8	410	65
o_9	334	329
⋮	⋮	⋮



Illustration

Clustering objects in a two-dimensional space using the Euclidean distance (the greater, the less similar).

	x	y
o_1	91	70
o_2	129	91
o_3	359	243
o_4	322	254
o_5	100	104
o_6	464	113
o_7	342	297
o_8	410	65
o_9	334	329
⋮	⋮	⋮



Inductive database vision

Querying patterns:

$$\{X \in P \mid \mathcal{Q}(X, \mathcal{D})\}$$

where:

- $\mathcal{D}$ is the dataset,
- P is the pattern space,
- $\mathcal{Q}$ is an inductive query.

Inductive database vision

Querying a clustering:

$$\{X \in P \mid \mathcal{Q}(X, \mathcal{D})\}$$

where:

- $\mathcal{D}$ is the dataset,
- P is the pattern space,
- $\mathcal{Q}$ is an inductive query.

Inductive database vision

Querying a clustering:

$$\{X \in P \mid \mathcal{Q}(X, \mathcal{D})\}$$

where:

- $\mathcal{D}$ is a set of objects $\mathcal{O}$ (described with attributes and) associated with a similarity measure,
- P is the pattern space,
- $\mathcal{Q}$ is an inductive query.

Inductive database vision

Querying a clustering:

$$\{X \in P \mid \mathcal{Q}(X, \mathcal{D})\}$$

where:

- $\mathcal{D}$ is a set of objects $\mathcal{O}$ (described with attributes and) associated with a similarity measure,
- P is [1]$\{(C_1, \ldots, C_k) \in (2^{\mathcal{O}})^k \mid \begin{cases} \forall i = 1..k, C_i \neq \emptyset \\ \forall j \neq i, C_i \cap C_j \neq \emptyset \\ \cup_{l=1}^{k} C_l = \mathcal{O} \end{cases}\}$,
- $\mathcal{Q}$ is an inductive query.

[1] k is here a user-defined parameter

Inductive database vision

Querying a clustering:

$$\{X \in P \mid \mathcal{Q}(X, \mathcal{D})\}$$

where:

- $\mathcal{D}$ is a set of objects $\mathcal{O}$ (described with attributes and) associated with a similarity measure,
- P is [1]$\{(C_1, \ldots, C_k) \in (2^{\mathcal{O}})^k \mid \begin{cases} \forall i = 1..k, C_i \neq \emptyset \\ \forall j \neq i, C_i \cap C_j \neq \emptyset \\ \cup_{l=1}^{k} C_l = \mathcal{O} \end{cases}\}$,
- $\mathcal{Q}$ is a function to optimize. It quantifies how similar are pairs of objects in a same cluster and how dissimilar are those in two different clusters.

[1] k is here a user-defined parameter

Inductive database vision

Querying a clustering:

$$\{X \in P \mid \mathcal{Q}(X, \mathcal{D})\}$$

where:

- $\mathcal{D}$ is a set of objects $\mathcal{O}$ (described with attributes and) associated with a similarity measure,
- P is [1] $\{(C_1, \ldots, C_k) \in (2^{\mathcal{O}})^k \mid \begin{cases} \forall i = 1..k, C_i \neq \emptyset \\ \forall j \neq i, C_i \cap C_j \neq \emptyset \\ \cup_{l=1}^{k} C_l = \mathcal{O} \end{cases}\}$,
- $\mathcal{Q}$ is a function to optimize. It quantifies how similar are pairs of objects in a same cluster and how dissimilar are those in two different clusters.

Variants exist, e. g., authorizing some overlapping of the clusters.

[1] k is here a user-defined parameter

Inductive database vision

Querying a clustering:

$$\{X \in P \mid \mathcal{Q}(X, \mathcal{D})\}$$

where:

- $\mathcal{D}$ is a set of objects $\mathcal{O}$ (described with attributes and) associated with a similarity measure,
- P is the set of all clusterings of $\mathcal{O}$,
- $\mathcal{Q}$ is a function to optimize. It quantifies how similar are pairs of objects in a same cluster and how dissimilar are those in two different clusters.

Variants exist, e. g., authorizing some overlapping of the clusters.

Naive algorithm

Input: $\mathcal{O}, \mathcal{D}, f$ the function to maximize
Output: the clustering of $\mathcal{O}$ maximizing f
$\mathcal{C}_{\max} \leftarrow \emptyset$
$f_{\max} \leftarrow -\infty$
for all clustering $\mathcal{C}$ of $\mathcal{O}$ **do**
  if $f(\mathcal{C}, \mathcal{D}) > f_{\max}$ **then**
    $f_{\max} \leftarrow f(\mathcal{C}, \mathcal{D})$
    $\mathcal{C}_{\max} \leftarrow \mathcal{C}$
  end if
end for
output($\mathcal{C}_{\max}$)

Number of 2-clusterings

Question

Assuming the number of clusters is a parameter of the algorithm and is set to 2, how many clusterings are enumerated?

Number of 2-clusterings

Question

Assuming the number of clusters is a parameter of the algorithm and is set to 2, how many clusterings are enumerated? $2^{|\mathcal{O}|-1}-1$.

A definition for f: the BetaCV function

To quantify how similar are pairs of objects in a same cluster and how dissimilar are those in two different clusters, a possible choice of the function f to maximize returns the ratio of the average similarity intra-cluster by the average similarity inter-cluster:

$$(\mathcal{C}, \mathcal{D}) \mapsto \frac{\frac{\sum_{C \in \mathcal{C}} \sum_{\{(o,o') \in C^2 \mid o \neq o'\}} s(o,o')}{\sum_{C \in \mathcal{C}} \binom{|C|}{2}}}{\frac{\sum_{\{(C,C') \in \mathcal{C}^2 \mid C \neq C'\}} \sum_{(o,o') \in C \times C'} s(o,o')}{\sum_{\{(C,C') \in \mathcal{C}^2 \mid C \neq C'\}} |C \times C'|}}$$

Computing the BetaCV value

Input: $\mathcal{C}$ a clustering of $\mathcal{O}$, a dataset $\mathcal{D}$ describing these objects, $s \in \mathbb{R}^{\mathcal{O}\times\mathcal{O}}$ a similarity measure
Output: $\mathrm{BetaCV}(\mathcal{C}, \mathcal{D}) \in \mathbb{R}$

$(a, b, c, d) \leftarrow (0, 0, 0, 0)$
for all $(C, C') \in \mathcal{C}$ **do**
  if $C = C'$ **then**
    $a \leftarrow a + \mathrm{intra}(C, \mathcal{D}, s)$
    $b \leftarrow b + \binom{|C|}{2}$
  else
    $c \leftarrow c + \mathrm{inter}(C, C', \mathcal{D}, s)$
    $d \leftarrow d + |C| \times |C'|$
  end if
end for
return $\left(\frac{ad}{bc}\right)$

intra and inter

intra

Input: $C \subseteq \mathcal{O}, \mathcal{D}$ a dataset describing the objects in $\mathcal{O}$, $s \in \mathbb{R}^{\mathcal{O}\times\mathcal{O}}$ a similarity measure
Output: $\sum_{\{(o,o')\in C^2 \mid o\neq o'\}} s(o,o')$
for all $(o,o') \in C^2 \mid o \neq o'$ **do**
  $a \leftarrow a + s(o,o')$
end for

inter

Input: $C \subseteq \mathcal{O}, C' \subseteq \mathcal{O}, \mathcal{D}$ a dataset describing the objects in $\mathcal{O}$, $s \in \mathbb{R}^{\mathcal{O}\times\mathcal{O}}$ a similarity measure
Output: $\sum_{(o,o')\in C\times C'} s(o,o')$
for all $(o,o') \in C \times C'$ **do**
  $c \leftarrow c + s(o,o')$
end for

Complexity of the naive approach

Question

Assuming the computation of a similarity is linear in the number of attributes $|\mathcal{A}|$, what is the complexity of one BetaCV computation?

Complexity of the naive approach

Question

Assuming the computation of a similarity is linear in the number of attributes $|\mathcal{A}|$, what is the complexity of one BetaCV computation? $O(|\mathcal{A}||\mathcal{O}|^2)$.

Complexity of the naive approach

Question

Assuming the computation of a similarity is linear in the number of attributes $|\mathcal{A}|$, what is the complexity of one BetaCV computation? $O(|\mathcal{A}||\mathcal{O}|^2)$.

Question

With the previous assumptions, what is the complexity of the naive approach?

Complexity of the naive approach

Question

Assuming the computation of a similarity is linear in the number of attributes $|\mathcal{A}|$, what is the complexity of one BetaCV computation? $O(|\mathcal{A}||\mathcal{O}|^2)$.

Question

With the previous assumptions, what is the complexity of the naive approach? $O(|\mathcal{A}||\mathcal{O}|^2 2^{|\mathcal{O}|})$.

Complexity of the naive approach

Question

Assuming the computation of a similarity is linear in the number of attributes $|\mathcal{A}|$, what is the complexity of one BetaCV computation? $O(|\mathcal{A}||\mathcal{O}|^2)$.

Question

With the previous assumptions, what is the complexity of the naive approach? $O(|\mathcal{A}||\mathcal{O}|^2 2^{|\mathcal{O}|})$.

Unless there are very few objects, the optimal clustering is unreachable. Clustering algorithms do not solve the task in an exact way.

Domain decomposition

A cheap clustering method acting as a pre-process for other clustering algorithms to treat each subset.

Outline

An unsupervised task

From a machine-learning point of view, clustering, like frequent pattern mining, is an *unsupervised* task: it is about discovering an *hidden organization* of the objects.

An unsupervised task

From a machine-learning point of view, clustering, like frequent pattern mining, is an *unsupervised* task: it is about discovering an *hidden organization* of the objects.

As a consequence, it is hard to assess a clustering.

Internal criteria of quality

BetaCV the ratio of the average intra-cluster similarity and the average inter-cluster similarity;

Internal criteria of quality

BetaCV the ratio of the average intra-cluster similarity and the average inter-cluster similarity;

Dunn the ratio of the minimal similarity between two objects in the same cluster and the maximal similarity between two objects in different clusters;

Internal criteria of quality

BetaCV the ratio of the average intra-cluster similarity and the average inter-cluster similarity;

Dunn the ratio of the minimal similarity between two objects in the same cluster and the maximal similarity between two objects in different clusters;

Davies-Bouldin the average of the ratios of the average similarity to the center of the assigned cluster and the similarity between the centers for the worst pairs of clusters (one of them being fixed);

Internal criteria of quality

BetaCV the ratio of the average intra-cluster similarity and the average inter-cluster similarity;

Dunn the ratio of the minimal similarity between two objects in the same cluster and the maximal similarity between two objects in different clusters;

Davies-Bouldin the average of the ratios of the average similarity to the center of the assigned cluster and the similarity between the centers for the worst pairs of clusters (one of them being fixed);

Silhouette for each object, the difference between the average similarity to the objects in the same cluster and the greatest average similarity to the objects in another cluster divided by the greatest term.

Comparing clusterings

The quality measures are not meaningful, unless compared to the measure obtained on another clustering:

of the same dataset to select the best clustering;

Comparing clusterings

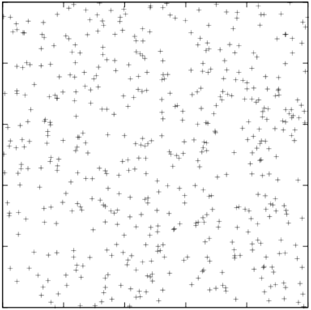
The quality measures are not meaningful, unless compared to the measure obtained on another clustering:

of the same dataset to select the best clustering;

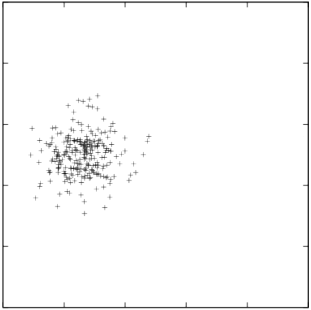
of a randomized version of the dataset to have an information about the tendency of the objects to be clustered.

Randomization of a dataset

Uniform distribution between the extrema of each attribute:

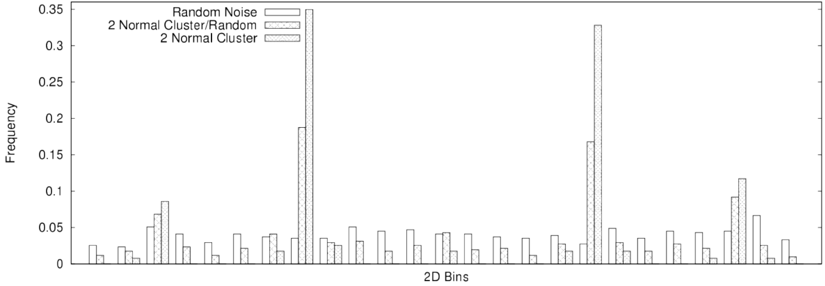


Normal distribution parametrized from the dataset:



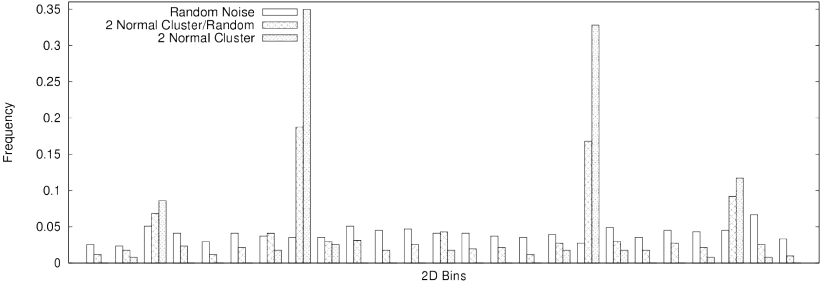
Clustering tendency without clustering

An histogram of the number of objects in bins of the dataset:



Clustering tendency without clustering

An histogram of the number of objects in bins of the dataset:



The histogram of the dataset is compared to that of a randomized version of it (e. g., using the sum of the quadratic errors in each bin).

Clustering tendency without clustering

Input: $\mathcal{D}$ a dataset describing the objects in $\mathcal{O}$, $\mathcal{B}$ a set of bins of the dataset, f a probability density function
Output: the clustering tendency of $\mathcal{D}$ w.r.t. f and binned according to $\mathcal{B}$

```
for all o ∈ O do
   for all B ∈ B do
      if o ∈ B then
         H[B] ← H[B] + 1
      end if
   end for
end for
return compute_tendency(B, f, H)
```

compute_tendency

Input: $\mathcal{B}$ a set of bins of the dataset $\mathcal{D}$, f a probability density function, H containing the number of objects in each bin in $\mathcal{B}$
Output: the clustering tendency of $\mathcal{D}$ w.r.t. f and binned according to $\mathcal{B}$
$t \leftarrow 0$
for all $B \in \mathcal{B}$ **do**
$$t \leftarrow t + \left(\frac{H[B]}{|\mathcal{O}|} - \int_{x \in B} f(x)\, dx\right)^2$$
end for
return(t)

Similarity between clusterings

If one clustering is taken as a reference, the entropy, in every reference cluster, can be computed.

Similarity between clusterings

If one clustering is taken as a reference, the entropy, in every reference cluster, can be computed.

Several indexes (the Jaccard index, the Folks and Mallows index, the Rand index, the adjusted Rand index) measure the similarity between two clusterings. They all are based on the number of pairs of objects that are in the same/different partition(s) in one clustering and in the same/different partition(s) in the other clustering.

Stability of a clustering

Some clustering algorithms involve (pseudo) randomness. Running them several times does not necessarily return the same clustering. However, if clusters exist in the data, the results should be close to each others.

Stability of a clustering

Some clustering algorithms involve (pseudo) randomness. Running them several times does not necessarily return the same clustering. However, if clusters exist in the data, the results should be close to each others.

A way of assessing a clustering obtained with such an *unstable algorithm* consists in running it several times and checking whether the obtained clusterings are similar.

Outline

Similarity and distance

Definition

Partitioning the objects so that the intra-cluster *similarities* are maximized and the inter-cluster *similarities* are minimized.

Similarity and distance

Similar (but more constrained!) definition

Partitioning the objects so that the intra-cluster *distances* are minimized and the inter-cluster *distances* are maximized.

Distance

A distance is a (square) matrix:

	o_1	o_2	$\dots$	o_m
o_1	$D(o_1, o_1)$	$D(o_1, o_2)$	$\dots$	$D(o_1, o_m)$
o_2	$D(o_2, o_1)$	$D(o_2, o_2)$	$\dots$	$D(o_2, o_m)$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
o_m	$D(o_m, o_1)$	$D(o_m, o_2)$	$\dots$	$D(o_m, o_m)$

such that:

$$\left\{ \begin{array}{l} \\ \\ \\ \end{array} \right. .$$

Distance

A distance is a (square) matrix:

	o_1	o_2	$\dots$	o_m
o_1	0	$D(o_1, o_2)$	$\dots$	$D(o_1, o_m)$
o_2	$D(o_2, o_1)$	0	$\dots$	$D(o_2, o_m)$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
o_m	$D(o_m, o_1)$	$D(o_m, o_2)$	$\dots$	0

such that:

$$\begin{cases} \forall (o_i, o_j) \in \mathcal{O}^2,\ D(o_i, o_j) = 0 \Leftrightarrow o_i = o_j \\ \\ \\ \end{cases}$$

Distance

A distance is a (square) matrix:

	o_1	o_2	$\dots$	o_m
o_1	0	$D(o_1, o_2)$	$\dots$	$D(o_1, o_m)$
o_2	$D(o_2, o_1)$	0	$\dots$	$D(o_2, o_m)$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
o_m	$D(o_m, o_1)$	$D(o_m, o_2)$	$\dots$	0

such that:

$$\begin{cases} \forall (o_i, o_j) \in \mathcal{O}^2,\ D(o_i, o_j) = 0 \Leftrightarrow o_i = o_j \\ \forall (o_i, o_j) \in \mathcal{O}^2,\ D(o_i, o_j) \geq 0 \end{cases}$$

Distance

A distance is a (square) matrix:

	o_1	o_2	$\dots$	o_m
o_1	0	$D(o_1, o_2)$	$\dots$	$D(o_1, o_m)$
o_2	$D(o_1, o_2)$	0	$\dots$	$D(o_2, o_m)$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
o_m	$D(o_1, o_m)$	$D(o_2, o_m)$	$\dots$	0

such that:

$$\begin{cases} \forall (o_i, o_j) \in \mathcal{O}^2,\ D(o_i, o_j) = 0 \Leftrightarrow o_i = o_j \\ \forall (o_i, o_j) \in \mathcal{O}^2,\ D(o_i, o_j) \geq 0 \\ \forall (o_i, o_j) \in \mathcal{O}^2,\ D(o_i, o_j) = D(o_j, o_i) \\ \end{cases}$$

Distance

A distance is a (square) matrix:

$$
\begin{array}{c|cccc}
 & o_1 & o_2 & \dots & o_m \\
\hline
o_1 & 0 & D(o_1, o_2) & \dots & D(o_1, o_m) \\
o_2 & D(o_1, o_2) & 0 & \dots & D(o_2, o_m) \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
o_m & D(o_1, o_m) & D(o_2, o_m) & \dots & 0
\end{array}
$$

such that:

$$
\begin{cases}
\forall (o_i, o_j) \in \mathcal{O}^2,\ D(o_i, o_j) = 0 \Leftrightarrow o_i = o_j \\
\forall (o_i, o_j) \in \mathcal{O}^2,\ D(o_i, o_j) \geq 0 \\
\forall (o_i, o_j) \in \mathcal{O}^2,\ D(o_i, o_j) = D(o_j, o_i) \\
\forall (o_i, o_j, o_k) \in \mathcal{O}^3,\ D(o_i, o_j) + D(o_j, o_k) \geq D(o_i, o_k)
\end{cases}
.
$$

A distance

Question

What is the shortest path between two points on earth?

A distance

Question

What is the shortest path between two points on earth?

Answer

It is that of a “segment” of a great circle.

Another distance

Question

What is the distance to travel between the earth and moon?

Another distance

Question

What is the distance to travel between the earth and moon?

Answer

Considering that, during the travel, they are not moving w.r.t. each other it is the distance between their centers minus their radius. If the latter assumption cannot be made (spaceship), ask a physicist (and the answer may not be a symmetric function, hence not a distance!).

Minkowski distance of order p

Let o_i and o_j two objects described with numerical attributes:

	a_1	a_2	...	a_n
o_i	$d_{i,1}$	$d_{i,2}$	...	$d_{i,n}$
o_j	$d_{j,1}$	$d_{j,2}$	...	$d_{j,n}$

Definition

The Minkowski distance of order p between o_i and o_j described with numerical attributes is:

$$\left(\sum_{k=1}^{n} |d_{i,k} - d_{j,k}|^p\right)^{\frac{1}{p}} \quad .$$

Euclidean distance: definition

Definition

The Euclidean distance is the Minkowski distance of order 2.

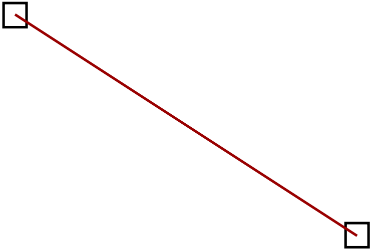
	x	y
o_1	91	70
o_3	359	243

Euclidean distance: definition

Definition

The Euclidean distance is the Minkowski distance of order 2.

	x	y
o_1	91	70
o_3	359	243



Euclidean distance: use

The "default" (most natural) natural distance.

Euclidean distance: use

The “default” (most natural) natural distance.

When only comparisons between distances are needed, the squared Euclidean distance is used because it is simpler to compute.

Manhattan distance

Definition

The Manhattan distance is the Minkowski distance of order 1.

□

	x	y
o_1	91	70
o_3	359	243

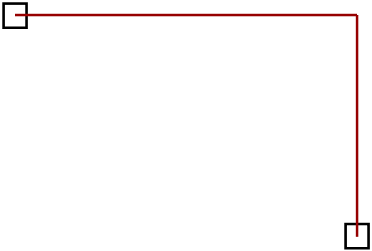
□

Manhattan distance

Definition

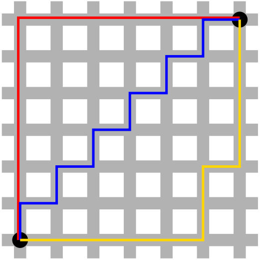
The Manhattan distance is the Minkowski distance of order 1.

	x	y
o_1	91	70
o_3	359	243



Manhattan distance: use

The Manhattan distance is the sum of the absolute differences according to each attribute, like the length of a taxicab ride in Manhattan.



Uniform distance

Definition

The uniform distance is the Minkowski distance when its order goes to infinity.

□

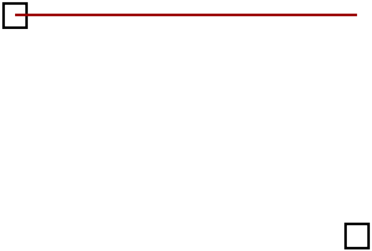
	x	y
o_1	91	70
o_3	359	243

□

Uniform distance

Definition

The uniform distance is the Minkowski distance when its order goes to infinity.



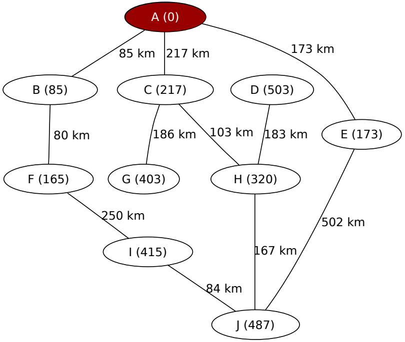
	x	y
o_1	91	70
o_3	359	243

Uniform distance: use

The single greatest difference on each attribute defines the uniform distance.

Distances between vertices in a graph

In a (resp. weighted) graph, the distance between two vertices is the number of edges (resp. the sum of their weights) on the shortest path connecting them.



Similar is enough

Definition

Partitioning the objects so that the intra-cluster *similarities* are maximized and the inter-cluster *similarities* are minimized.

A distance is a real function on couples of objects. It satisfies:

$$\begin{cases} \forall (o_i, o_j) \in \mathcal{O}^2,\ D(o_i, o_j) = 0 \Leftrightarrow o_i = o_j \\ \forall (o_i, o_j) \in \mathcal{O}^2,\ D(o_i, o_j) \geq 0 \\ \forall (o_i, o_j) \in \mathcal{O}^2,\ D(o_i, o_j) = D(o_j, o_i) \\ \forall (o_i, o_j, o_k) \in \mathcal{O}^3,\ D(o_i, o_j) + D(o_j, o_k) \geq D(o_i, o_k) \end{cases} .$$

Similar is enough

Definition

Partitioning the objects so that the intra-cluster *similarities* are maximized and the inter-cluster *similarities* are minimized.

A similarity is a real function on couples of objects. It satisfies:

$$\begin{cases} \\ \forall (o_i, o_j) \in \mathcal{O}^2,\ D(o_i, o_j) = D(o_j, o_i) \\ \\ \end{cases} .$$

Cosine similarity

Definition

The cosine similarity is the cosine of the angle between the objects seen as vectors.

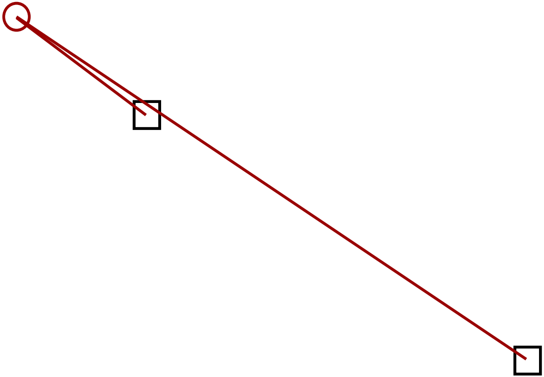
	x	y
o_1	91	70
o_3	359	243

Cosine similarity

Definition

The cosine similarity is the cosine of the angle between the objects seen as vectors.

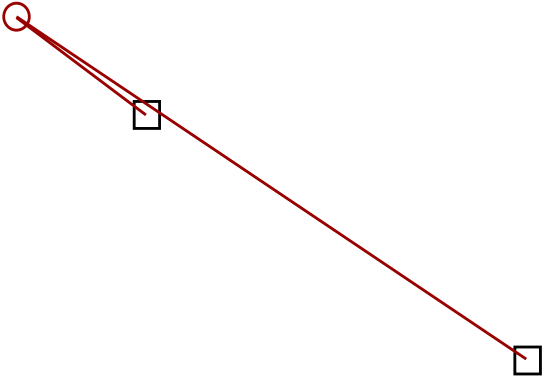
	x	y
o_1	91	70
o_3	359	243



Cosine similarity

Definition

The cosine similarity between two objects o_i and o_j seen as vectors is $\frac{o_i \cdot o_j}{||o_i||_2 ||o_j||_2}$.



	x	y
o_1	91	70
o_3	359	243

Cosine similarity: use

The objects are seen as vectors whose norms are irrelevant. This similarity measure is *not* related to a distance measure.

From categorical to numerical

Ordered categorical attributes can be turned numerical and the same similarities can be used.

From categorical to numerical

Ordered categorical attributes can be turned numerical and the same similarities can be used.

Unordered categorical attributes can be turned Boolean (every category becomes an attribute whose domain is $\{0, 1\}$) and the same similarities can be used but:

From categorical to numerical

Ordered categorical attributes can be turned numerical and the same similarities can be used.

Unordered categorical attributes can be turned Boolean (every category becomes an attribute whose domain is $\{0, 1\}$) and the same similarities can be used but:

- the curse of dimensionality strikes when the domains of the categorical attributes are large;

From categorical to numerical

Ordered categorical attributes can be turned numerical and the same similarities can be used.

Unordered categorical attributes can be turned Boolean (every category becomes an attribute whose domain is $\{0, 1\}$) and the same similarities can be used but:

- the curse of dimensionality strikes when the domains of the categorical attributes are large;
- a categorical attribute has a weight that is proportional with the cardinality of its domain.

Hamming distance

The objects, described with (unordered) categorical attributes, can be seen as strings (each value is a character of the string) of the same length.

Hamming distance

The objects, described with (unordered) categorical attributes, can be seen as strings (each value is a character of the string) of the same length.

Definition

The Hamming distance is the number of substitutions to turn one string into the other.

	sex	*job*
o_1	male	teacher
o_3	female	teacher

Hamming distance

The objects, described with (unordered) categorical attributes, can be seen as strings (each value is a character of the string) of the same length.

Definition

The Hamming distance is the number of substitutions to turn one string into the other.

	sex	*job*
o_1	male	teacher
o_3	female	teacher

$$(1 - \delta_{\text{male,female}}) + (1 - \delta_{\text{teacher,teacher}}) = 1$$

Hamming distance: use

The “default” (most natural) natural distance. With Boolean attributes, it is the Manhattan distance. The Lee distance generalizes the Hamming distance but requires a metric on each categorical attribute.

Jaccard index

Definition

The ratio between the number of identical aligned characters and the total number of characters.

	sex	*job*
o_1	male	teacher
o_3	female	teacher

Jaccard index

Definition

The ratio between the number of identical aligned characters and the total number of characters.

	sex	*job*
o_1	male	teacher
o_3	female	teacher

$$\frac{\delta_{\text{male,female}} + \delta_{\text{teacher,teacher}}}{2} = \frac{1}{2} \; .$$

Jaccard index

Definition

The ratio between the number of identical aligned characters and the total number of characters.

	sex	*job*
o_1	male	teacher
o_3	female	teacher

$$\frac{\delta_{\text{male,female}} + \delta_{\text{teacher,teacher}}}{2} = \frac{1}{2} .$$

The Jaccard index encodes the same information as the Hamming distance.

Other distances between strings

Other distance are defined on strings of varying sizes. Some (such as the Damerau, the Levenshtein and the Damerau-Levenshtein distance) count some or all the four following edit operations to transform one string into the other one: insertion, deletions, substitutions and (adjacent or not) transpositions. Other distance measures are based on aligning the two words. They are computationally costly.

Outline

Choice of attributes

Too many attributes lead to nothing because of the “curse of dimensionality” (when the dimensionality goes to infinity, the distance between any pair of objects becomes the same).

Choice of attributes

Too many attributes lead to nothing because of the "curse of dimensionality" (when the dimensionality goes to infinity, the distance between any pair of objects becomes the same).

Selecting two highly correlated attributes is like taking into account the same information twice.

Choice of attributes

Too many attributes lead to nothing because of the "curse of dimensionality" (when the dimensionality goes to infinity, the distance between any pair of objects becomes the same).

Selecting two highly correlated attributes is like taking into account the same information twice.

Dimensionality reduction techniques help but, if they create new attributes (i. e., unlike feature selection), the discovered clustering is harder to interpret.

Scaling

When there is no reason to do otherwise, attributes are normalized before computing distances. In this way, every attribute has the same weight.

Scaling

When there is no reason to do otherwise, attributes are normalized before computing distances. In this way, every attribute has the same weight.

Giving more (resp. less) weight to an attribute is simply achieved by multiplying (scaling) its normalized values by a constant greater (resp. smaller) than 1.

Min-max normalization

Definition

An affine transformation of the values so that the extremal ones are chosen.

Min-max normalization

Definition

An affine transformation of the values so that the extremal ones are chosen.

It should be used when the extrema are known to be so “in theory”.

Z-score normalization

Definition

The number of standard deviations above (positive Z-score) or below (negative Z-score) the mean.

- Center: $x - \mu$ with $\mu = \frac{1}{N}\Sigma_{i=1}^{N} x_i$
- Reduce: $\frac{x-\mu}{\sigma}$ with $\sigma = \sqrt{\frac{1}{N}\Sigma_{i=1}^{N}(x_i - \mu)^2}$

Z-score normalization

Definition

The number of standard deviations above (positive Z-score) or below (negative Z-score) the mean.

- Center: $x - \mu$ with $\mu = \frac{1}{N}\Sigma_{i=1}^{N} x_i$
- Reduce: $\frac{x-\mu}{\sigma}$ with $\sigma = \sqrt{\frac{1}{N}\Sigma_{i=1}^{N}(x_i - \mu)^2}$

The default normalization.

Z-score normalization

Definition

The number of standard deviations above (positive Z-score) or below (negative Z-score) the mean.

- Center: $x - \mu$ with $\mu = \frac{1}{N}\Sigma_{i=1}^{N} x_i$
- Reduce: $\frac{x-\mu}{\sigma}$ with $\sigma = \sqrt{\frac{1}{N}\Sigma_{i=1}^{N}(x_i - \mu)^2}$

The default normalization.

The Mahalanobis distance takes into account the distribution of the objects along *all* their attributes.

Example

	Age	Salary
p_1	50	11000
p_2	70	11100
p_3	60	11122
p_4	60	11074

Without normalization:

with Manhattan distance:

$d(p_1, p_2) = (20 + 100) = 120$

$d(p_1, p_3) = (10 + 122) = 132$

$d(p_1, p_2) < d(p_1, p_3)$ ☹

Example

	Age	Salary
p_1	50	11000
p_2	70	11100
p_3	60	11122
p_4	60	11074

	μ	σ
Age	60	$5\sqrt{2} = 7.07$
Salary	11074	45.97

Example

	Age	Salary
p_1	50	11000
p_2	70	11100
p_3	60	11122
p_4	60	11074

	μ	σ
Age	60	$5\sqrt{2} = 7.07$
Salary	11074	45.97

	Age	Salary
p_1	-1.4	-1.6
p_2	1.4	0.6
p_3	0	1.04
p_4	0	0

$d(p_1, p_2) = (2.8 + 2.2) = 5$

$d(p_1, p_3) = (1.4 + 2.64) = 4.04$

$d(p_1, p_2) > d(p_1, p_3)$ ☺

Distorting the distances

Frequently, the difference of values of one attribute is not a relevant measure. The analyst often wants to compress/dilate the differences of smaller/larger values.

Distorting the distances

Frequently, the difference of values of one attribute is not a relevant measure. The analyst often wants to compress/dilate the differences of smaller/larger values.

This typically is the case when the distribution of the values follows a power-law (often resulting from a preferential attachment phenomenon).

Useless transformation

When computing distances between objects, adding a constant to a numerical attribute is useless.

Useless transformation

When computing distances between objects, adding a constant to a numerical attribute is useless.

Assuming a normalization of (or the choice of the weights for) the numerical attributes occur after the transformation, a multiplicative factor is useless too.

Compressing distances between smaller values

To compress (resp. dilate) distances between smaller (resp. larger) values, exponential functions are often applied:

$$x \mapsto e^{kx} \quad .$$

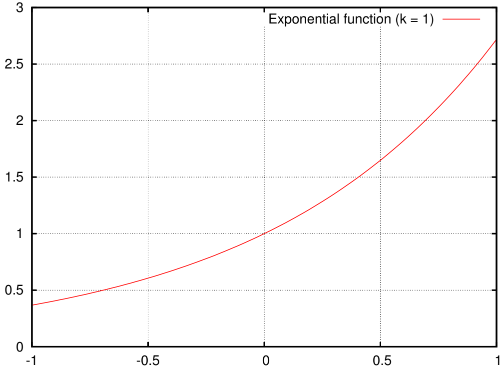
Compressing distances between smaller values

To compress (resp. dilate) distances between smaller (resp. larger) values, exponential functions are often applied:

$$x \mapsto e^{kx} \quad .$$

An additive parameter to x is useless.

Exponential function



Compressing distances between larger values

To compress (resp. dilate) distances between larger (resp. smaller) values, logarithmic functions are often applied:

$$x \mapsto ln(x + k) \ .$$

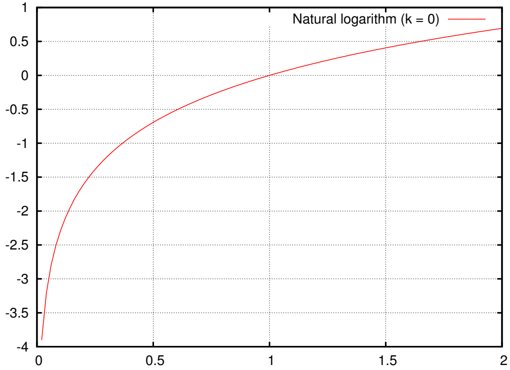
Compressing distances between larger values

To compress (resp. dilate) distances between larger (resp. smaller) values, logarithmic functions are often applied:

$$x \mapsto ln(x + k) \ .$$

A multiplicative parameter to x is useless.

Logarithm



Outline

Summary

- Clustering is partitioning the objects so that each partition contains similar objects and objects in different partitions are dissimilar;

Summary

- Clustering is partitioning the objects so that each partition contains similar objects and objects in different partitions are dissimilar;
- Unless there are very few objects, finding such a global organization can only be achieved in an approximate way (optimization);

Summary

- Clustering is partitioning the objects so that each partition contains similar objects and objects in different partitions are dissimilar;
- Unless there are very few objects, finding such a global organization can only be achieved in an approximate way (optimization);
- A clustering can be assessed in different ways (quality measure, stability, etc.);

Summary

- Clustering is partitioning the objects so that each partition contains similar objects and objects in different partitions are dissimilar;
- Unless there are very few objects, finding such a global organization can only be achieved in an approximate way (optimization);
- A clustering can be assessed in different ways (quality measure, stability, etc.);
- A clustering tendency can be computed by comparison with a randomized version of the dataset;

Summary

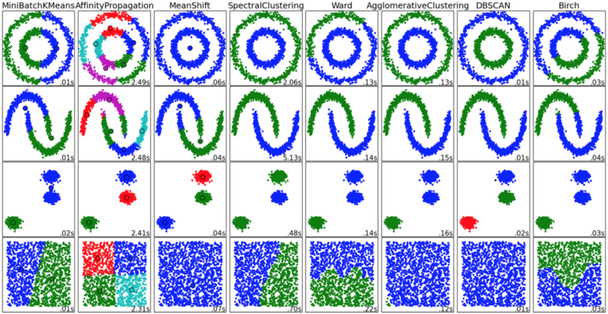
- Clustering is partitioning the objects so that each partition contains similar objects and objects in different partitions are dissimilar;
- Unless there are very few objects, finding such a global organization can only be achieved in an approximate way (optimization);
- A clustering can be assessed in different ways (quality measure, stability, etc.);
- A clustering tendency can be computed by comparison with a randomized version of the dataset;
- Clustering algorithms are parametrized with a similarity measure to be wisely chosen;

Summary

- Clustering is partitioning the objects so that each partition contains similar objects and objects in different partitions are dissimilar;
- Unless there are very few objects, finding such a global organization can only be achieved in an approximate way (optimization);
- A clustering can be assessed in different ways (quality measure, stability, etc.);
- A clustering tendency can be computed by comparison with a randomized version of the dataset;
- Clustering algorithms are parametrized with a similarity measure to be wisely chosen;
- Attributes often need to be chosen, scaled (usually normalized) and/or distorted.

Characteristics of clustering methods

- Extensibility
- Ability to handle different data types
- Ability to discover cluster of different forms (convex, ...)
- Parameter setting
- Robustness (noisy data, outliers)



http://scikit-learn.org/stable/modules/clustering.html

The end.