Clustering algorithms

Violaine Antoine

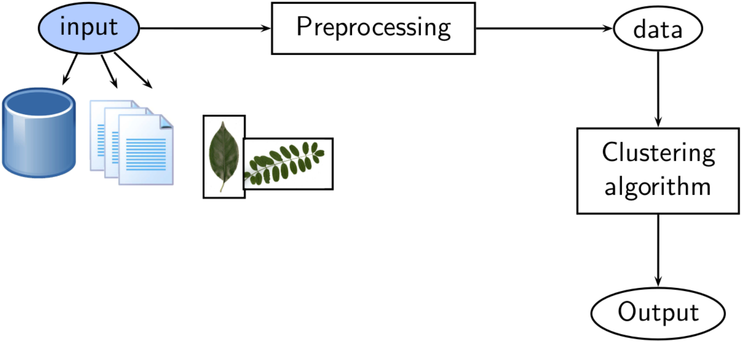
ISIMA / LIMOS

January, 2019

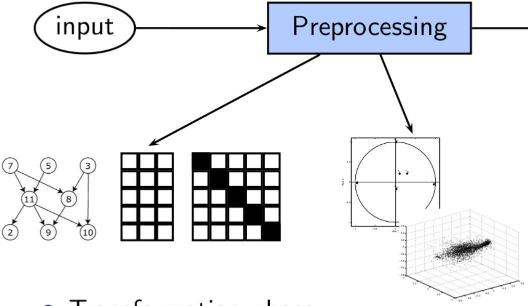
Outline

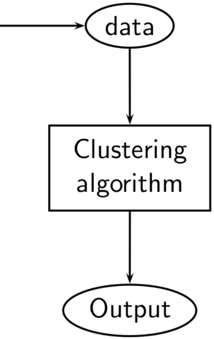
Outline

Global scheme



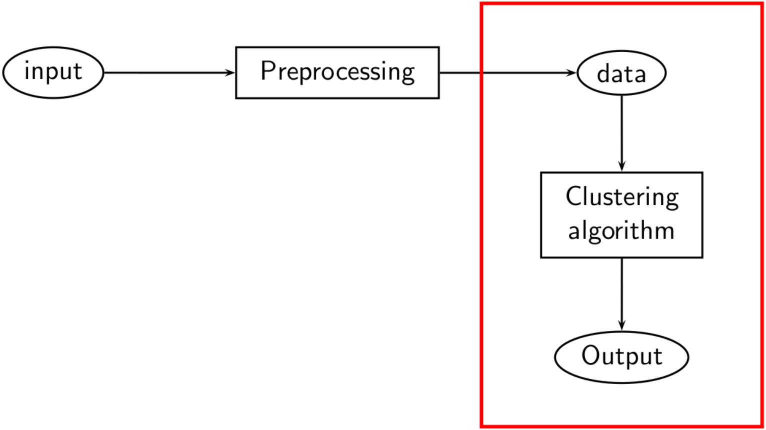
Global scheme





- Transformation phase
- Data analysis
  - PCA, CA, correlation measures, . . .
- normalization, feature selection, . . .

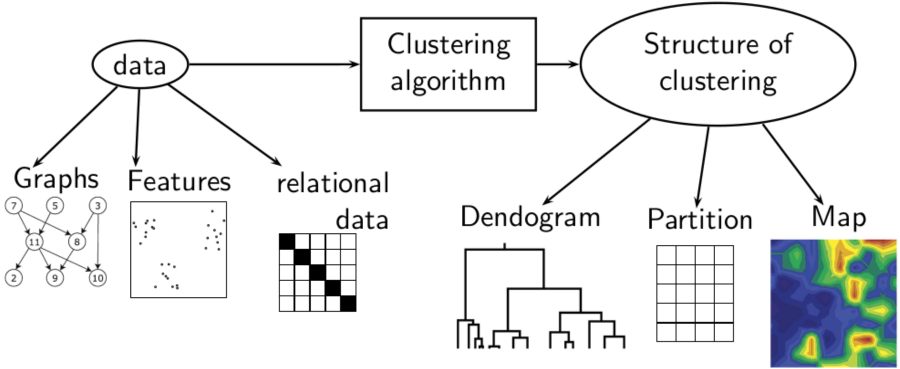
Global scheme



Clustering process

Clustering

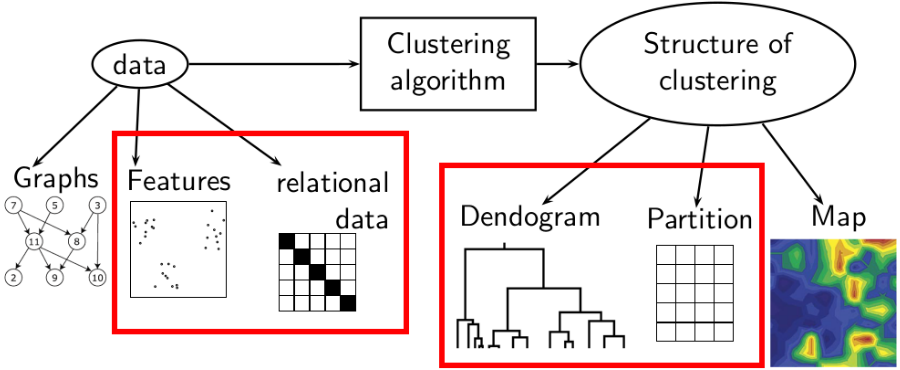
Grouping objects into cluster following a similarity notion



Clustering process

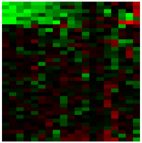
Clustering

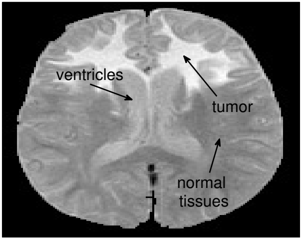
Grouping objects into cluster following a similarity notion



Applications

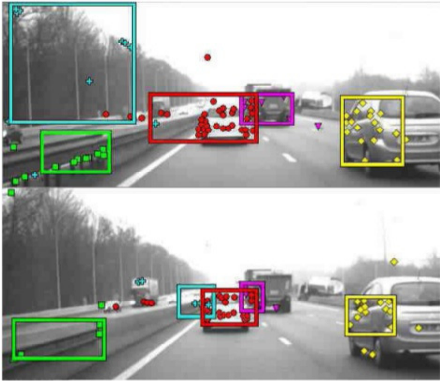
- Biology and bioinformatics
  - grouping genes with related expression patterns using DNA microarray [?]
  - clustering plants or animals
- Medicine [?]





Applications

- Detection of objects
  - robotics
  - video surveillance
  - automotive driving assistance [?]



Applications

- Geology
  - earth-quake and volcanoes studies
- Social network analysis
- Market research
  - Findind groups of customers
  - Predicting behavior of shopping
- World wide web [**?**]
  - documents engine search
  - images engine search

Problematic : the background knowledge

Clustering

Grouping objects into cluster following a similarity notion

Which similarity/dissimilarity definition should be chosen ?

Problematic : the background knowledge

Clustering

Grouping objects into cluster following a similarity notion

Which similarity/dissimilarity definition should be chosen ?

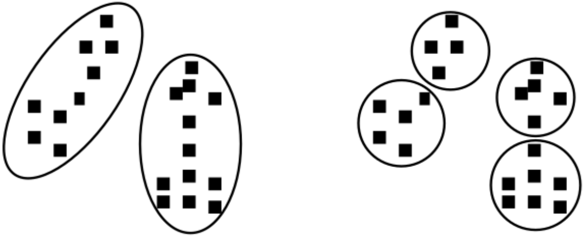
Distance measures

Two major family of distances :

- Euclidean distances
  - Mahalanobis distance : $d(\mathbf{x}, \mathbf{y}) = \sqrt{\mathbf{x} - \mathbf{y})^\top \Sigma (\mathbf{x} - \mathbf{y})}$
  - Manhattan distance : $d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{p} |x_i - y_i|$
- Non-Euclidean distances
  - Edit distance : measures difference between 2 strings
  - Jaccard index : measures dissimilarity between sample sets

Problematic : clustering or subclustering

Which clusters retains ?



Other problematics

- How to measure the correctness of a partition ?
- How to deal with noise in the feature vectors ?
- What to do if we have uncertain data ?
- How to detect outliers ?
- ...

Notations

Input notations

- $\mathbf{x}_i \in \{\mathbf{x}_1 \dots \mathbf{x}_N\}$ the set of objects with p attributes

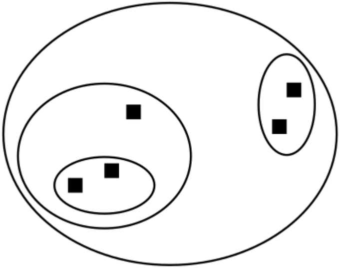
Clustering notations

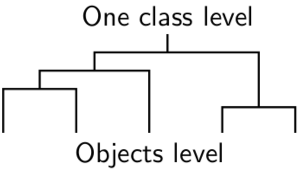
- $\omega_k \in \Omega = \{\omega_1 \dots \omega_c\}$ the set of clusters
- $n_1, n_2, \dots n_c$ the number of objects belonging to $\omega_1, \omega_2, \omega_c$

Outline

Hierarchical clustering

Produces a set of nested cluster organised as a dendrogram.





Two categories of hierarchical clustering

- Agglomerative methods
- Divise methods

Hierarchical clustering

Advantages

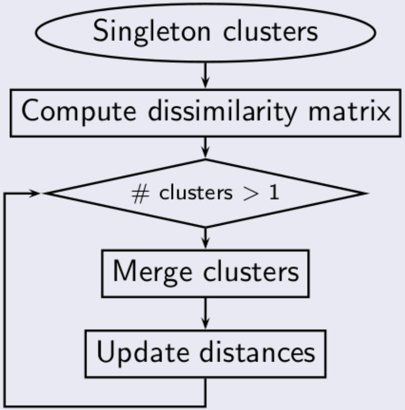
- No assumptions on the number of clusters
- Visualization of subclusterings
- Deterministic methods

Disadvantages

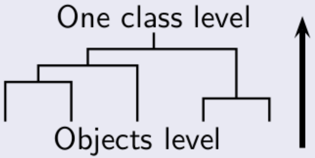
- Interpretation of the hierarchy can be complex
- Considerable amount of data implies large dendrogram

Agglomerative clustering

Basic scheme

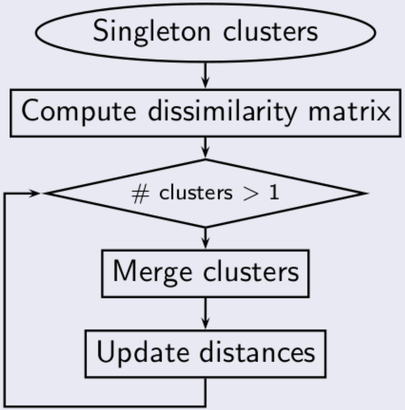


Bottom-up hierarchy

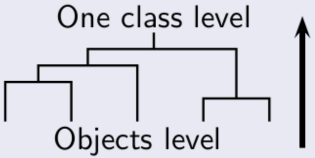


Agglomerative clustering

Basic scheme



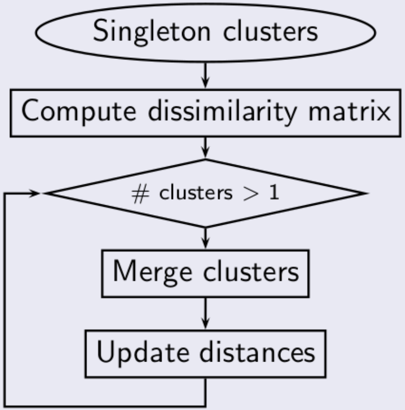
Bottom-up hierarchy



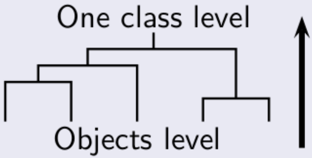
- Merge criterion : distance minimum

Agglomerative clustering

Basic scheme



Bottom-up hierarchy



- Merge criterion : distance minimum
- Update distances
  - single-link
  - complete-link
  - average-link

The single-link algorithm [? ?]

Distance calculation

Let c_i and c_j be two clusters

- The distance between c_i and c_j is the minimum distance between any object in c_i and any object in c_j

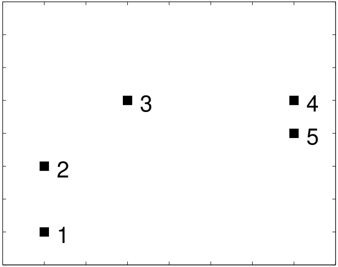
$\Rightarrow$ The distance is defined by the two most similar objects

Underlying idea

- Importance is given to regions where clusters are closed
- Overall structure of the cluster is negliged

$\Rightarrow$ local similarity-based clustering method

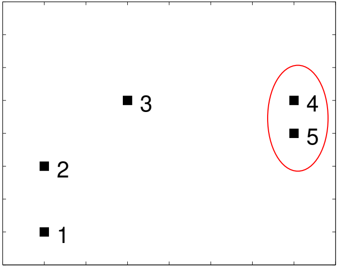
Single-link example



	1	2	3	4	5
1	0	1	2.2	3.6	3.4
2	1	0	1.4	3.2	3
3	2.2	1.4	0	2	2.1
4	3.6	3.2	2	0	0.5
5	3.4	3	2.1	0.5	0

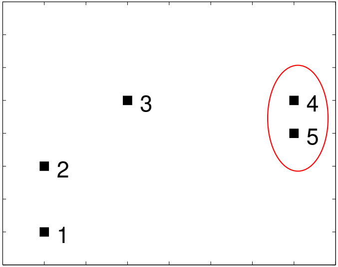
1 2 3 4 5

Single-link example



	1	2	3	4	5
1	0	1	2.2	3.6	3.4
2	1	0	1.4	3.2	3
3	2.2	1.4	0	2	2.1
4	3.6	3.2	2	0	0.5
5	3.4	3	2.1	0.5	0

Single-link example

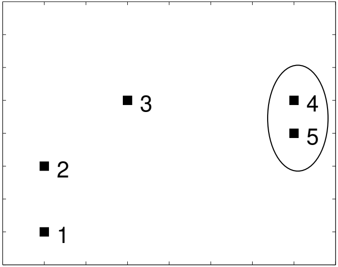


	1	2	3	4	5
1	0	1	2.2	3.6	3.4
2	1	0	1.4	3.2	3
3	2.2	1.4	0	2	2.1
4	3.6	3.2	2	0	0.5
5	3.4	3	2.1	0.5	0

Update distances

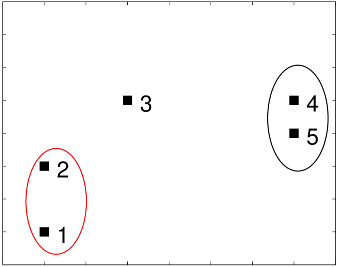
d((4,5),1) = min(d(4,1),d(5,1)) = 3.4
d((4,5),2) = min(d(4,2),d(5,2)) = 3
d((4,5),3) = min(d(4,3),d(5,3)) = 2

Single-link example



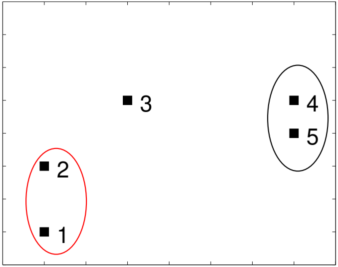
	1	2	3	(4,5)
1	0	1	2.2	3.4
2	1	0	1.4	3
3	2.2	1.4	0	2
(4,5)	3.4	3	2	0

Single-link example



	1	2	3	(4,5)
1	0	1	2.2	3.4
2	1	0	1.4	3
3	2.2	1.4	0	2
(4,5)	3.4	3	2	0

Single-link example

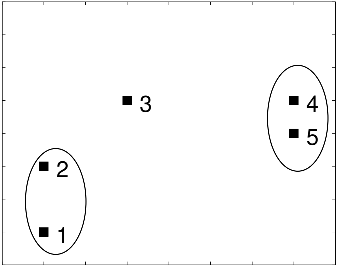


	1	2	3	(4,5)
1	0	1	2.2	3.4
2	1	0	1.4	3
3	2.2	1.4	0	2
(4,5)	3.4	3	2	0

Update distances

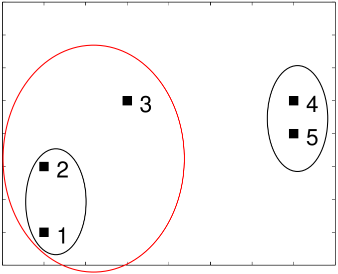
$$d((1,2),3) = \min(d(1,3),d(2,3)) = 1.4$$
$$d((1,2),(4,5)) = \min(d(1,(4,5)), d(2,(4,5))) = 3$$

Single-link example

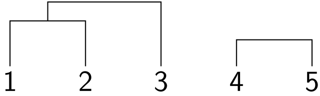


	(1,2)	3	(4,5)
(1,2)	0	1.4	3
3	1.4	0	2
(4,5)	3	2	0

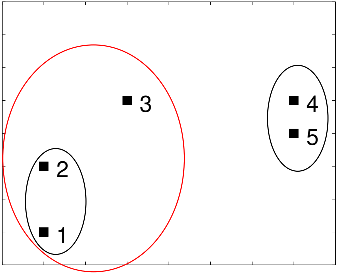
Single-link example

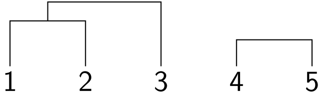


	(1,2)	3	(4,5)
(1,2)	0	1.4	3
3	1.4	0	2
(4,5)	3	2	0



Single-link example



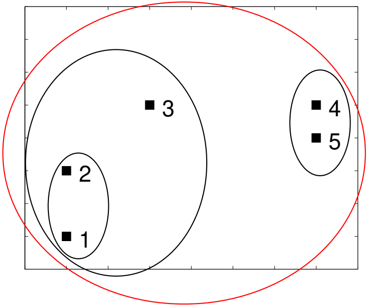


	(1,2)	3	(4,5)
(1,2)	0	1.4	3
3	1.4	0	2
(4,5)	3	2	0

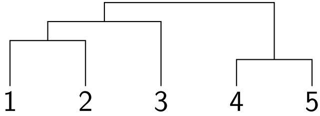
Update distances

d((1,2,3),(4,5)) $=$
min(d(3,(4,5)),d((1,2)(4,5))) $= 2$

Single-link example



	(1,2)	3	(4,5)
(1,2)	0	1.4	3
3	1.4	0	2
(4,5)	3	2	0

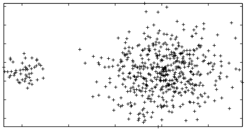


Update distances

d((1,2,3),(4,5)) =
min(d(3,(4,5)),d((1,2)(4,5))) = 2

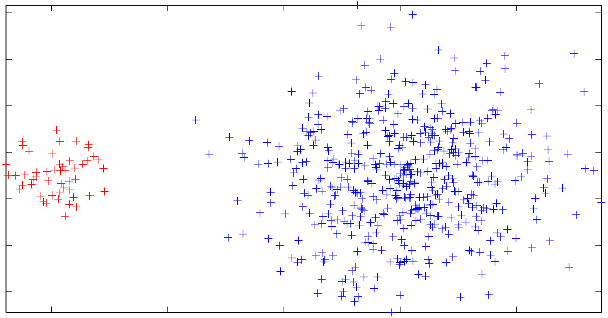
Strengths of the single-link clustering

Original data



Enable to find :

- non elliptical shaped groups
- unbalanced groups



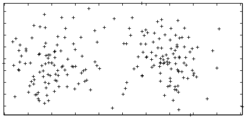
Single-link algorithm

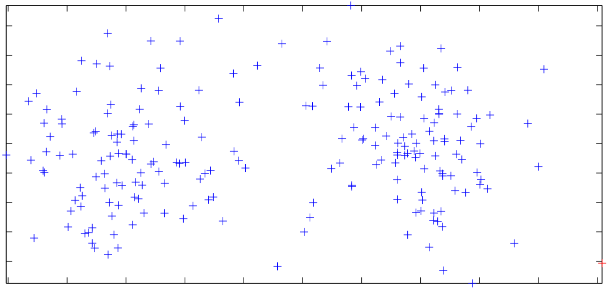
Limitations of the single-link clustering

Sensitive to

- noise
- outliers

Original data





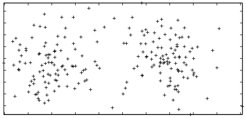
Single-link algorithm

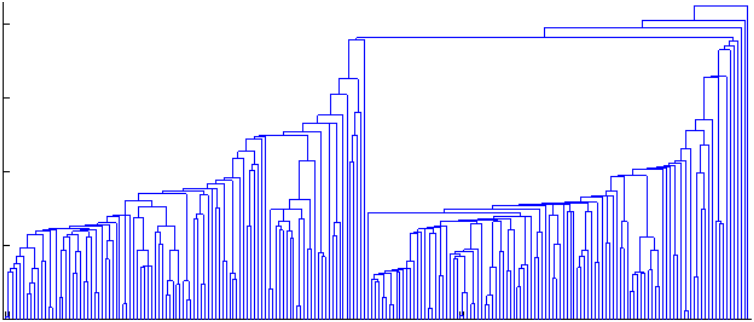
Limitations of the single-link clustering

Sensitive to

- noise
- outliers

Original data





dendrogram

The complete-link algorithm [?]

Distance calculation

Let c_i and c_j be two clusters

- The distance between c_i and c_j is the maximum distance between any object in c_i and any object in c_j

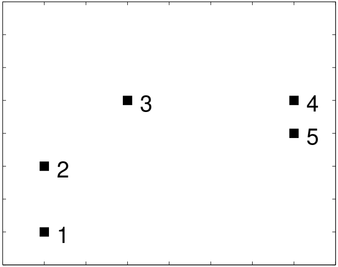
$\Rightarrow$ The distance is defined by the two most dissimilar objects

Underlying idea

- Merge cluster with the smallest diameter
- Importance is given to the cluster structure

$\Rightarrow$ global similarity-based clustering method

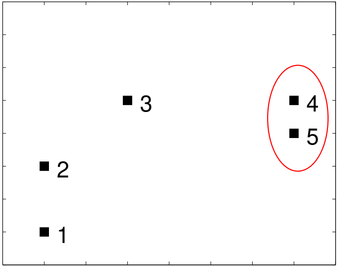
Complete-link example



	1	2	3	4	5
1	0	1	2.2	3.6	3.4
2	1	0	1.4	3.2	3
3	2.2	1.4	0	2	2.1
4	3.6	3.2	2	0	0.5
5	3.4	3	2.1	0.5	0

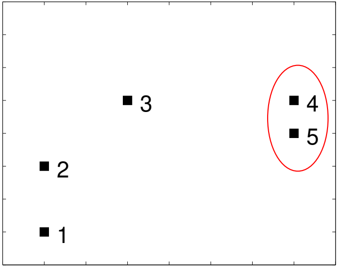
1 2 3 4 5

Complete-link example



	1	2	3	4	5
1	0	1	2.2	3.6	3.4
2	1	0	1.4	3.2	3
3	2.2	1.4	0	2	2.1
4	3.6	3.2	2	0	0.5
5	3.4	3	2.1	0.5	0

Complete-link example

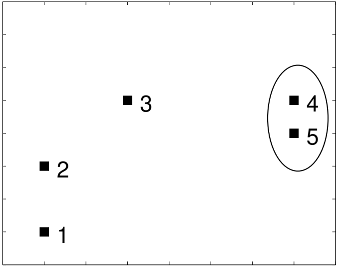


	1	2	3	4	5
1	0	1	2.2	3.6	3.4
2	1	0	1.4	3.2	3
3	2.2	1.4	0	2	2.1
4	3.6	3.2	2	0	0.5
5	3.4	3	2.1	0.5	0

Update distances

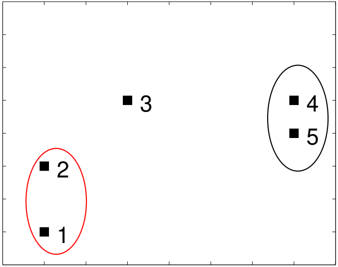
d((4,5),1)=max(d(4,1),d(5,1))=3.6
d((4,5),2)=max(d(4,2),d(5,2))=3.2
d((4,5),3)=max(d(4,3),d(5,3))=2.1

Complete-link example



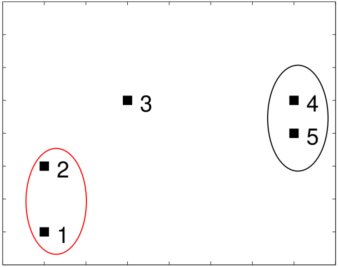
	1	2	3	(4,5)
1	0	1	2.2	3.6
2	1	0	1.4	3.2
3	2.2	1.4	0	2.1
(4,5)	3.6	3.2	2.1	0

Complete-link example



	1	2	3	(4,5)
1	0	1	2.2	3.6
2	1	0	1.4	3.2
3	2.2	1.4	0	2.1
(4,5)	3.6	3.2	2.1	0

Complete-link example

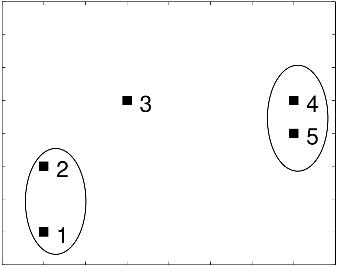


	1	2	3	(4,5)
1	0	1	2.2	3.6
2	1	0	1.4	3.2
3	2.2	1.4	0	2.1
(4,5)	3.6	3.2	2.1	0

Update distances

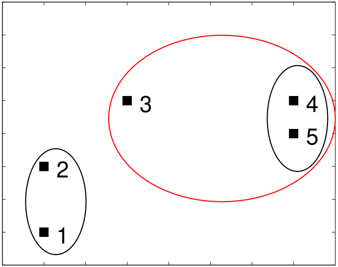
$$\begin{aligned} d((1,2),3) &= \max(d(1,3),d(2,3)){=}2.2 \\ d((1,2),(4,5)) &= \max(d(1,(4,5)), \\ &\qquad d(2,(4,5))){=}3.6 \end{aligned}$$

Complete-link example

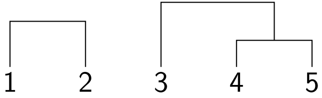


	(1,2)	3	(4,5)
(1,2)	0	2.2	3.6
3	2.2	0	2.1
(4,5)	3.6	2.1	0

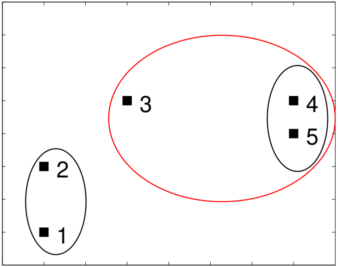
Complete-link example



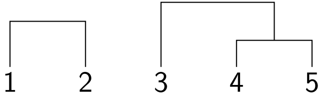
	(1,2)	3	(4,5)
(1,2)	0	2.2	3.6
3	2.2	0	2.1
(4,5)	3.6	2.1	0



Complete-link example



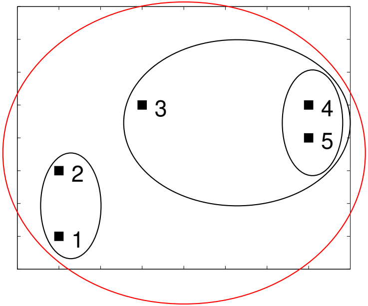
	(1,2)	3	(4,5)
(1,2)	0	2.2	3.6
3	2.2	0	2.1
(4,5)	3.6	2.1	0



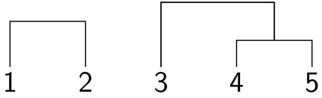
Update distances

d((1,2),(3,4,5))=
max(d((1,2),3),d((1,2),(4,5)))=3.6

Complete-link example



	(1,2)	3	(4,5)
(1,2)	0	2.2	3.6
3	2.2	0	2.1
(4,5)	3.6	2.1	0

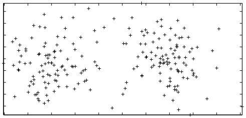


Update distances

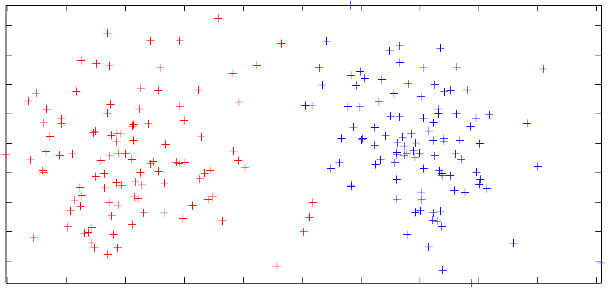
d((1,2),(3,4,5))=
max(d((1,2),3),d((1,2),(4,5)))=3.6

Strengths of the complete-link clustering

Original data



- Enable to find compact shaped cluster
- Less sensitive to noise



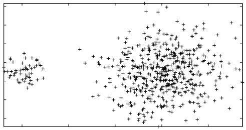
Complete-link algorithm

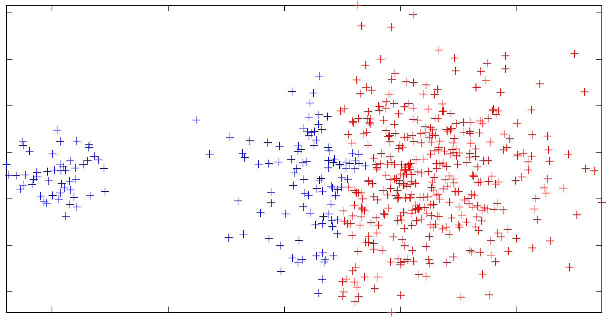
Limitations of the complete-link clustering

Sensitive to

- unbalanced cluster

Original data





Complete-link algorithm

Average-link algorithm [?]

Distance calculation

Let c_i and c_j be two clusters

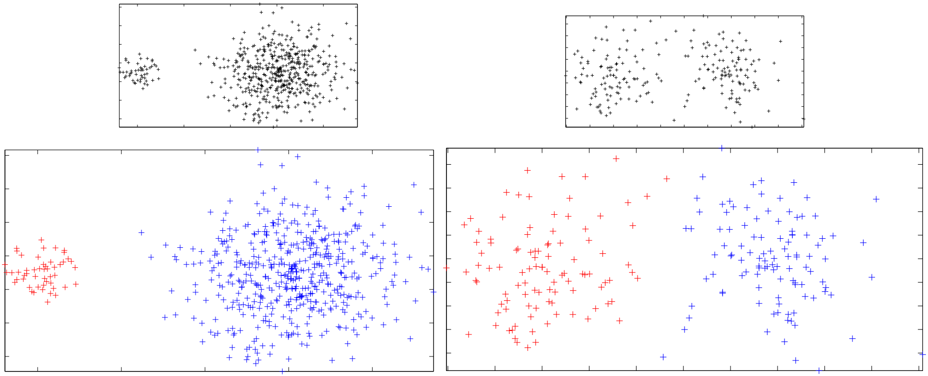
- The distance between c_i and c_j is the average distance between any object in c_i and any object in c_j

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

Underlying idea

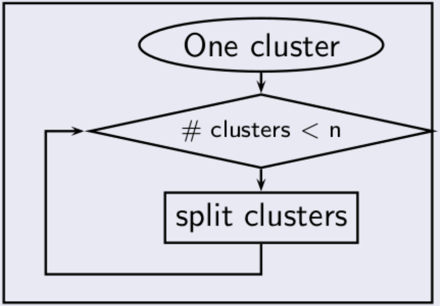
Reducing drawbacks associated to single and complete link

Strengths of the average-link clustering

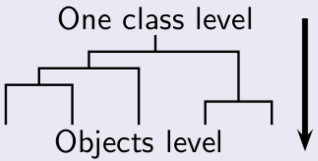


Divise clustering

Basic scheme

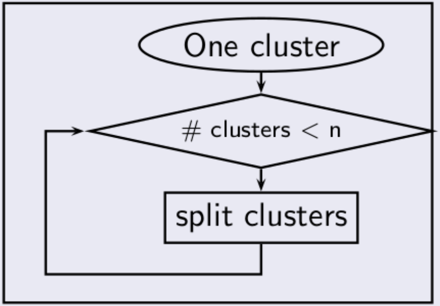


Top-down hierarchy

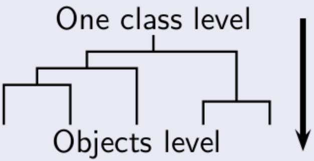


Divise clustering

Basic scheme



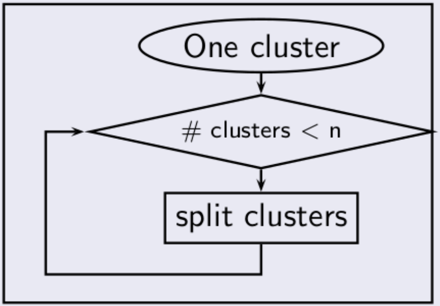
Top-down hierarchy



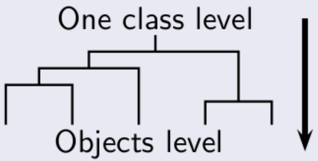
- Split criterion :
  - using one/several attributes for a specific split
  - intercluster distances

Divise clustering

Basic scheme



Top-down hierarchy



- Split criterion :
  - using one/several attributes for a specific split
  - intercluster distances

⇒ computationally intensive

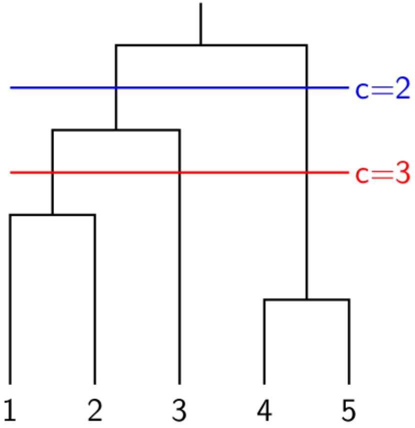
⇒ less widely used than agglomerative methods

The cut of dendrogram

Getting a crisp partition

Goal : Find the α-cut that

- select c groups
- find balanced clusters
- minimize a clustering validation measure
- select the maximum distance between two merges
- ...



Outline

Partitional clustering

Produces a hard or fuzzy partition.

Types of partition

- Hard partition
  - Each object $\mathbf{x}_i$ belongs to an exclusive class ω_k
  - $p_{ik} = \{0, 1\}$, $\sum p_{ik} = 1$
- Fuzzy partition
  - $\mathbf{x}_i$ has a degree of membership for each class ω_k
  - $u_{ik} \in [0, 1]$, $\sum_{k=1}^{c} u_{ik} = 1$

Density-based clustering methods

Basic idea

Clusters are dense regions in the data space.

Important notions

- The density
- The connectivity between objects

Advantages

Non-parametric methods, i.e. no assumption about :

- the number of clusters
- their distribution

The DBSCAN algorithm

Neighborhood definition

The neighborhood of an object $\mathbf{x}_i$ is :

$$N_\varepsilon(\mathbf{x}_i) = \{\mathbf{x}_j \quad s.t. \quad d_{ij} \leq \varepsilon\}$$

Ddr definition

$\mathbf{x}_j$ is directly density-reachable from $\mathbf{x}_i$ if

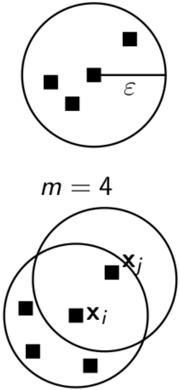
$\rightarrow$ $\mathbf{x}_j \in N_\varepsilon(\mathbf{x}_i)$

$\rightarrow$ $|N_\varepsilon(\mathbf{x}_i)| \geq m$

Consequence

Carefull ! The definition is non symmetric :

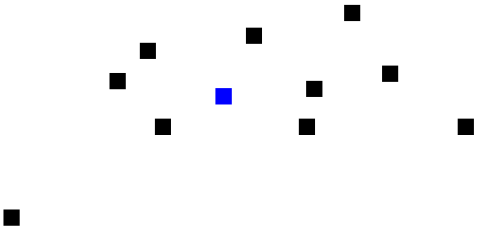
$$\mathbf{x}_j \text{ is ddr from } \mathbf{x}_i \nRightarrow \mathbf{x}_i \text{ is ddr from } \mathbf{x}_j$$



The DBSCAN algorithm

Expansion rule

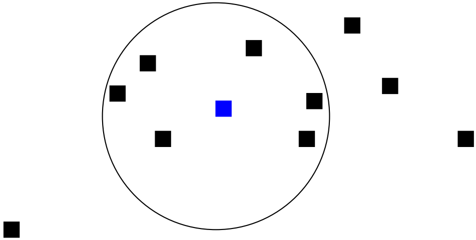
If $\mathbf{x}_i \in \omega_k$ and $\mathbf{x}_j$ is ddr from $\mathbf{x}_i$ then $\mathbf{x}_j \in \omega_k$



The DBSCAN algorithm

Expansion rule

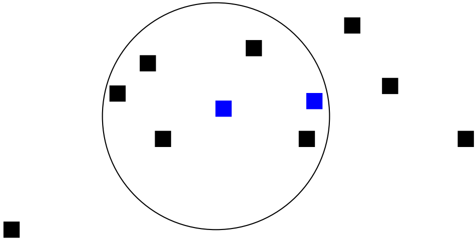
If $\mathbf{x}_i \in \omega_k$ and $\mathbf{x}_j$ is ddr from $\mathbf{x}_i$ then $\mathbf{x}_j \in \omega_k$



The DBSCAN algorithm

Expansion rule

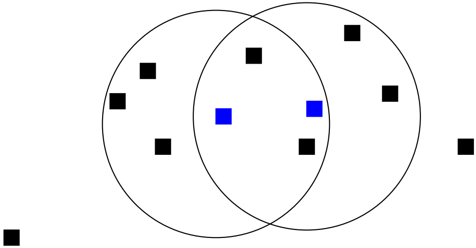
If $\mathbf{x}_i \in \omega_k$ and $\mathbf{x}_j$ is ddr from $\mathbf{x}_i$ then $\mathbf{x}_j \in \omega_k$



The DBSCAN algorithm

Expansion rule

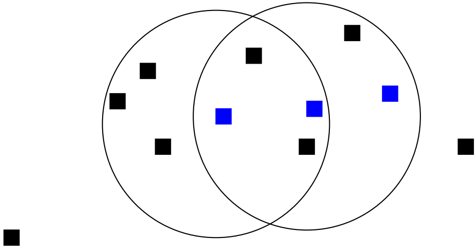
If $\mathbf{x}_i \in \omega_k$ and $\mathbf{x}_j$ is ddr from $\mathbf{x}_i$ then $\mathbf{x}_j \in \omega_k$



The DBSCAN algorithm

Expansion rule

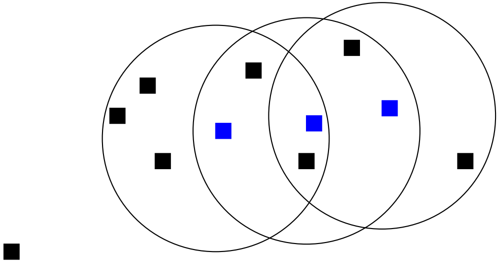
If $\mathbf{x}_i \in \omega_k$ and $\mathbf{x}_j$ is ddr from $\mathbf{x}_i$ then $\mathbf{x}_j \in \omega_k$



The DBSCAN algorithm

Expansion rule

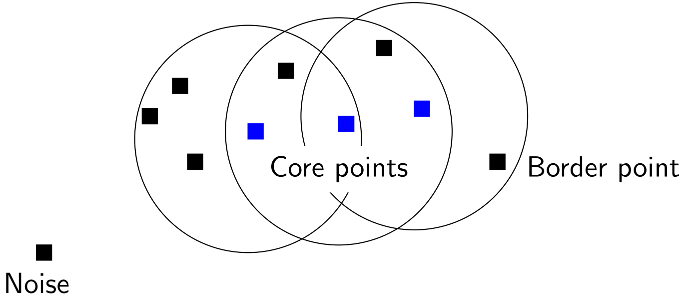
If $\mathbf{x}_i \in \omega_k$ and $\mathbf{x}_j$ is ddr from $\mathbf{x}_i$ then $\mathbf{x}_j \in \omega_k$



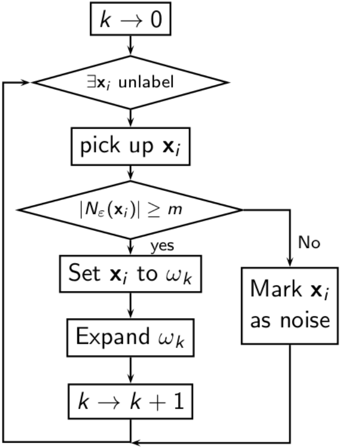
The DBSCAN algorithm

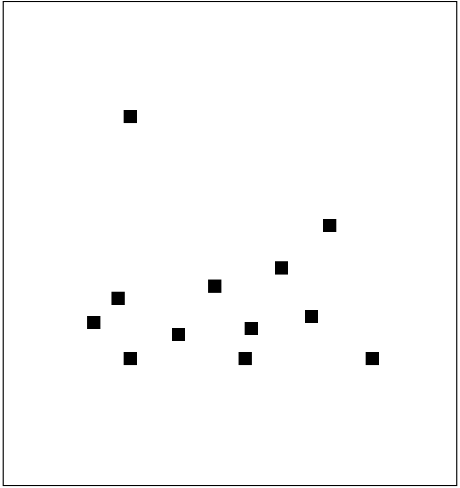
Expansion rule

If $\mathbf{x}_i \in \omega_k$ and $\mathbf{x}_j$ is ddr from $\mathbf{x}_i$ then $\mathbf{x}_j \in \omega_k$

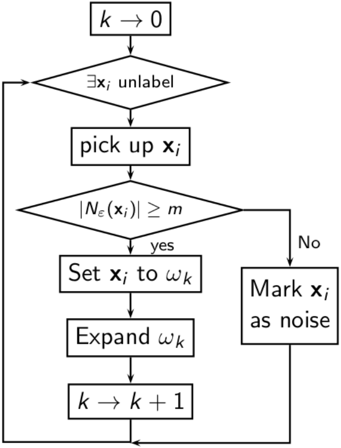


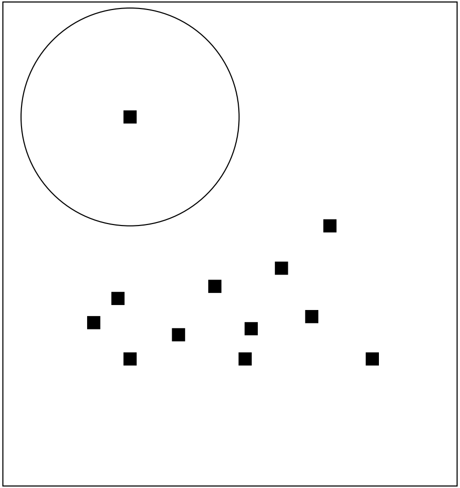
The DBSCAN algorithm



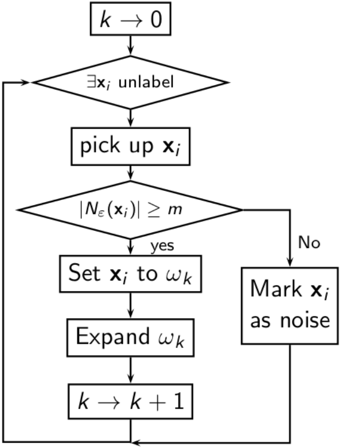


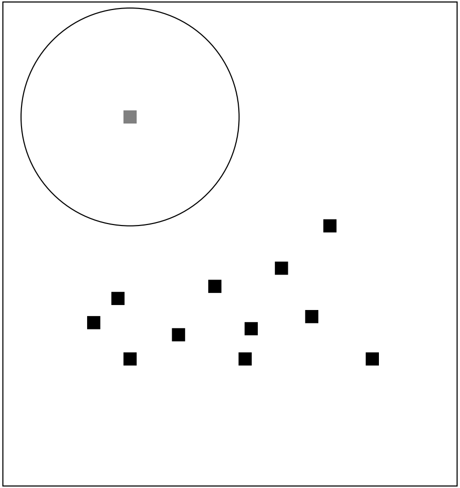
The DBSCAN algorithm



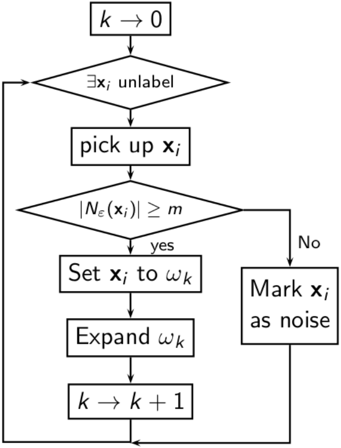


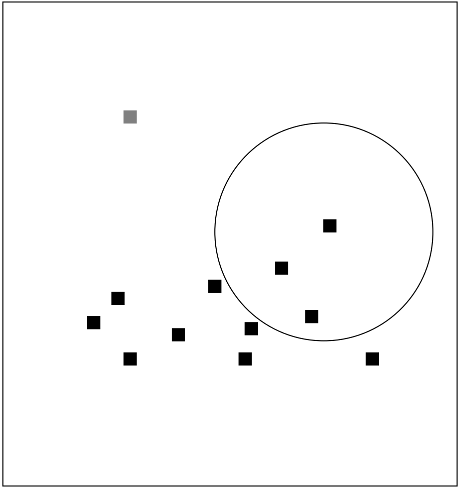
The DBSCAN algorithm



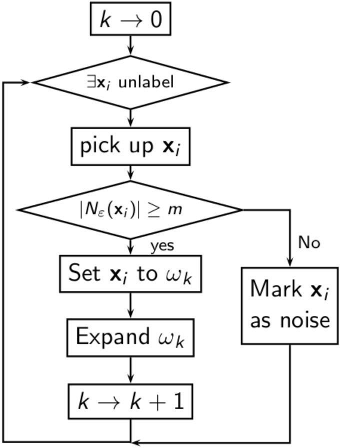


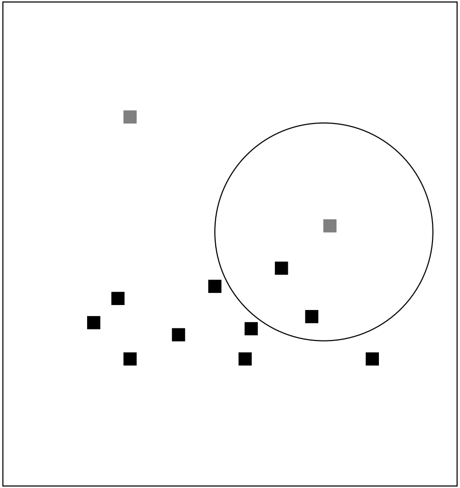
The DBSCAN algorithm



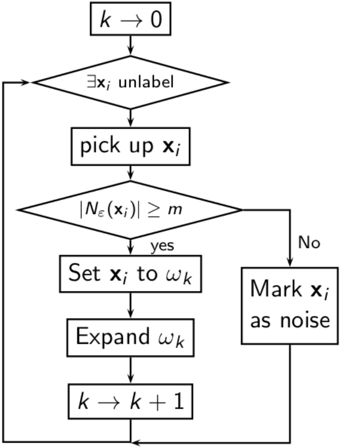


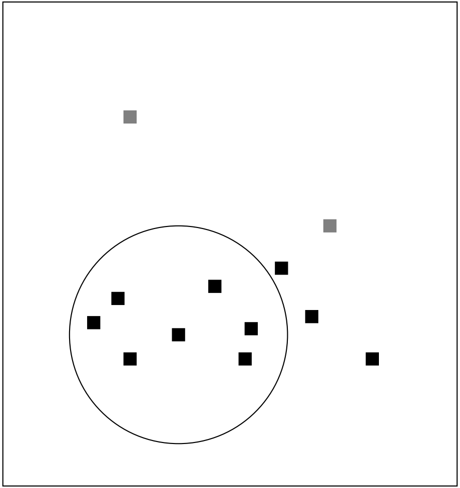
The DBSCAN algorithm



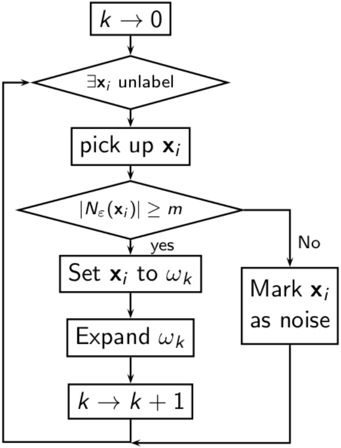


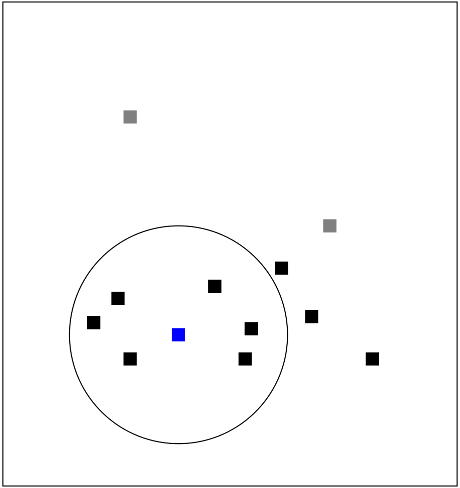
The DBSCAN algorithm



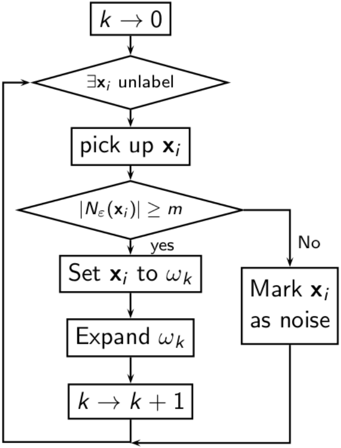


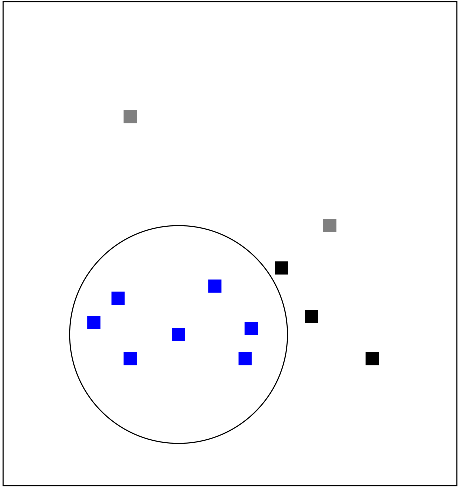
The DBSCAN algorithm



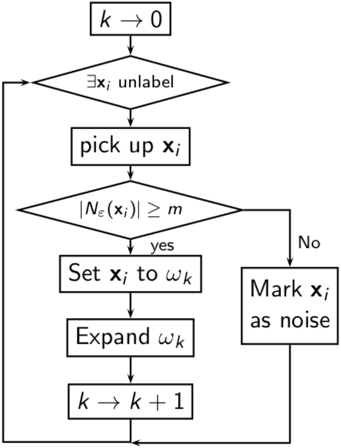


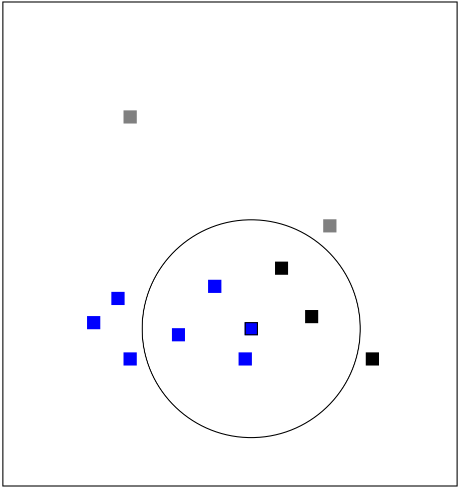
The DBSCAN algorithm



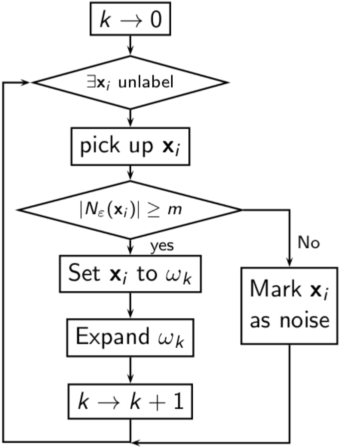


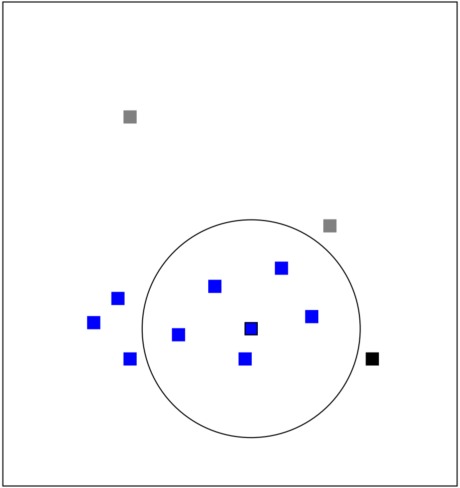
The DBSCAN algorithm



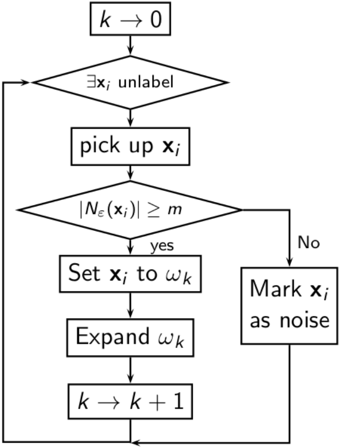


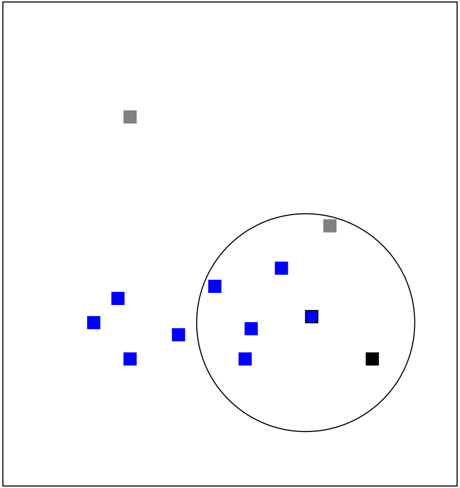
The DBSCAN algorithm



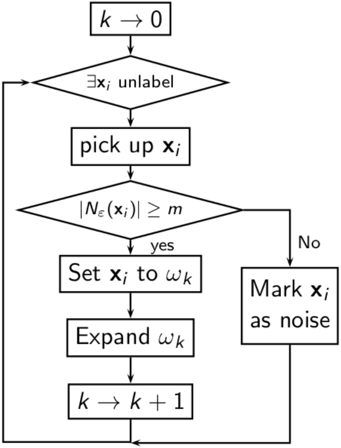


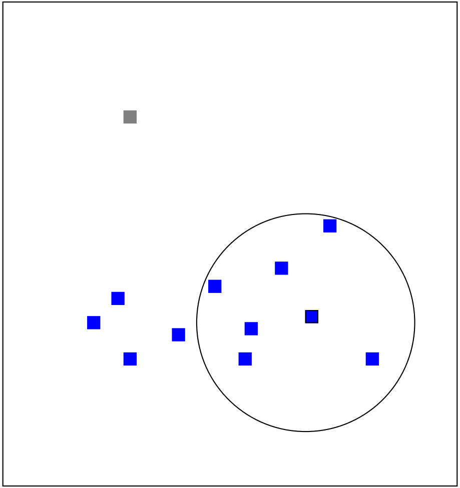
The DBSCAN algorithm



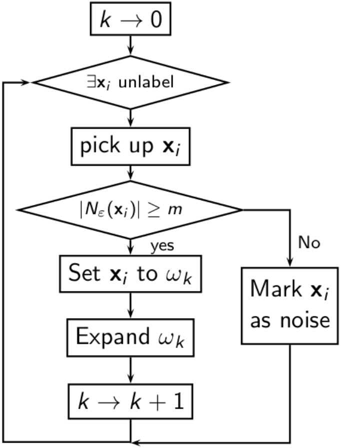


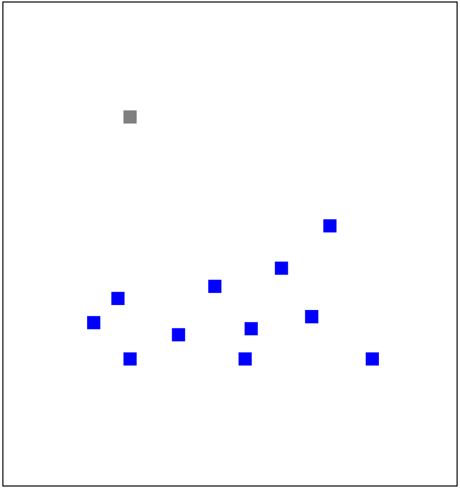
The DBSCAN algorithm





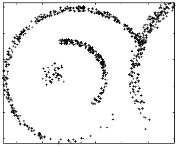
The DBSCAN algorithm





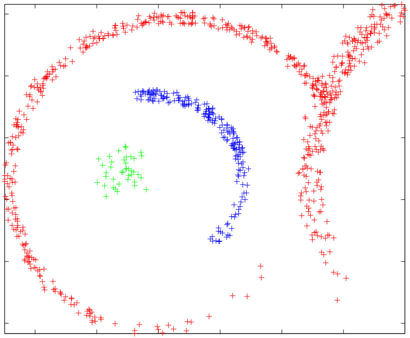
Strengths of DBSCAN

Original data



Enable to :

- find arbitrary shapes and unbalanced groups
- deal with noise



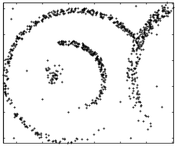
DBSCAN (m=5, $\varepsilon = 3$)

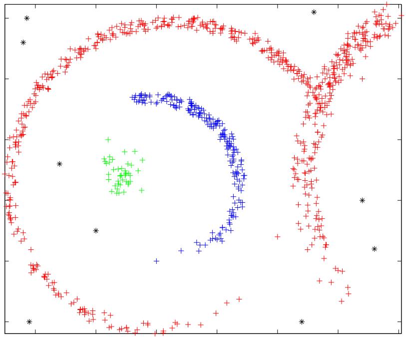
Strengths of DBSCAN

Enable to :

- find arbitrary shapes and unbalanced groups
- deal with noise

Original data





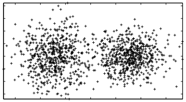
DBSCAN (m=5, $\varepsilon = 3$)

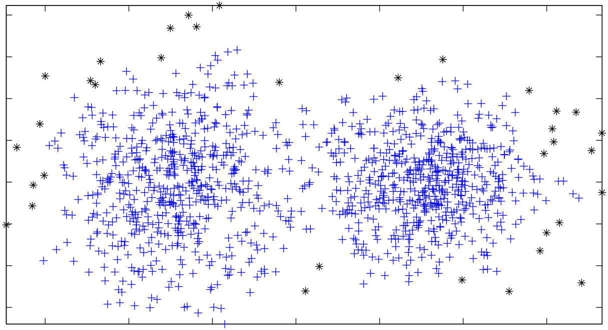
Limitations of DBSCAN

Unable to :

- split overlapped cluster
- cluster with large difference densities

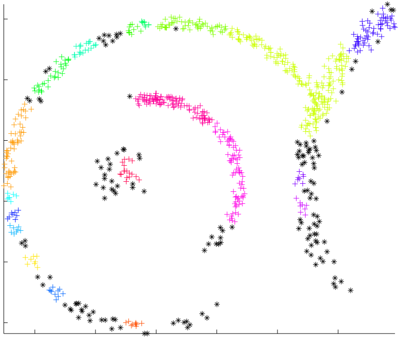
Original data



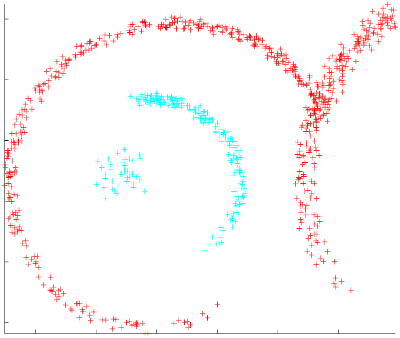


DBSCAN (m=5, $\varepsilon = 0.5$)

DBSCAN : sensitivity to parameters



$\varepsilon = 0.5$



$\varepsilon = 4$

The parameters determination

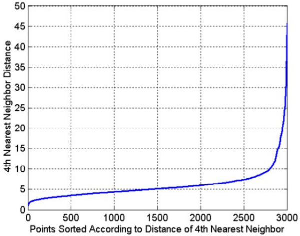
Basic idea

For one object $\mathbf{x}_i$ in a cluster :

- Neighbors are roughly at the same distance of $\mathbf{x}_i$
- Outliers have more distance with $\mathbf{x}_i$ than other neighbors

Suppose m fixed.

To set ε, plot sorted distance of every points to its 4^{th} nearest neighbor :



Geometrical clustering methods

Basic idea

A cluster ω_k is represented by a centroid $\mathbf{v}_k$
$\Rightarrow$ The number of cluster is known

Geometrical clustering methods

Basic idea

A cluster ω_k is represented by a centroid $\mathbf{v}_k$

$\Rightarrow$ The number of cluster is known

Notations

$\mathbf{V} = \{\mathbf{v}_1 \dots \mathbf{v}_c\}$

$d_{ik} = d(\mathbf{x}_i, \mathbf{v}_j) = \|\mathbf{x}_i - \mathbf{v}_k\|$

k-means and its variants

k-means [?]

Objective function

min J_{KM} s.t.

$$J_{KM} = \sum_{i=1}^{N} \sum_{\mathbf{x}_i \in \omega_k}^{C} \|\mathbf{x}_i - \mathbf{v}_k\|^2$$

Optimization

NP-Hard $\Rightarrow$ minimization using an iterative procedure :

$$\min J_{KM} \text{ w.t.r to clusters} \quad \rightleftarrows \quad \min J_{KM}(\mathbf{V})$$

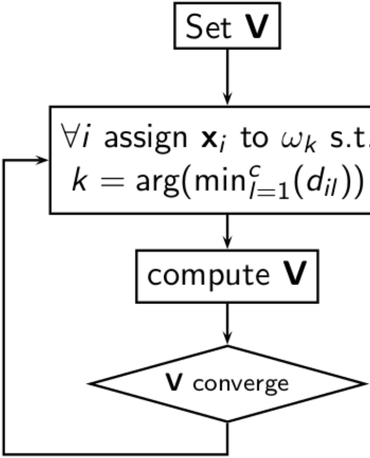
Advantage

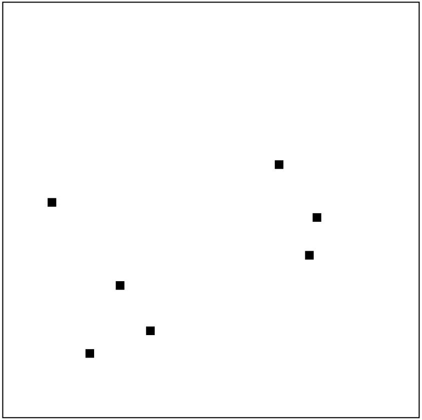
Fast

Disadvantage

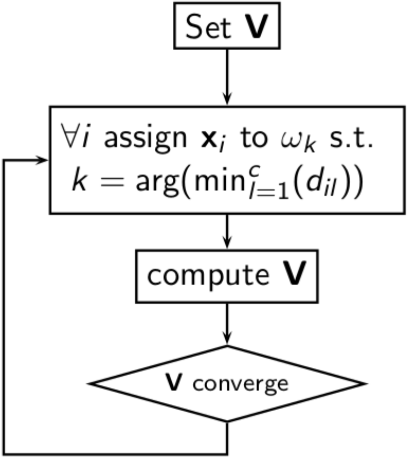
Risk of local minimum

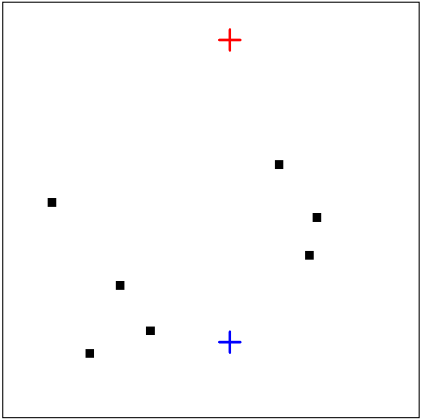
Optimization



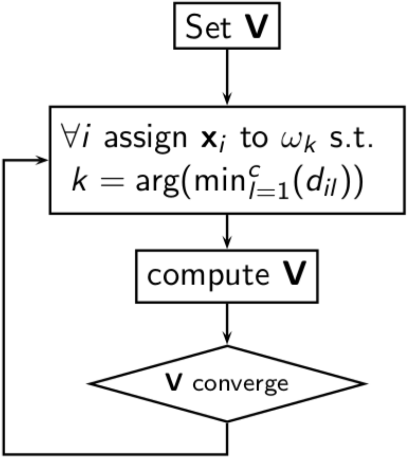


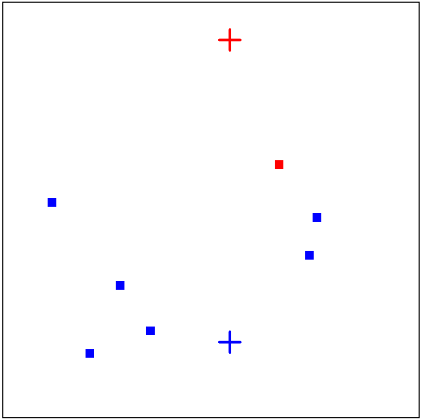
Optimization



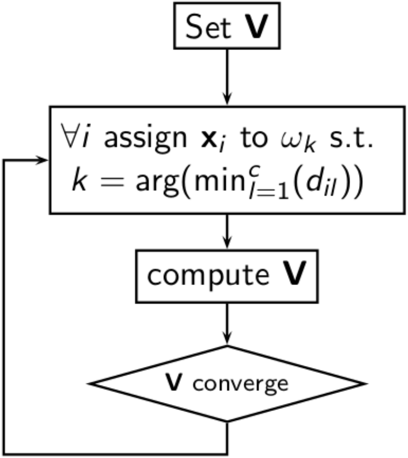


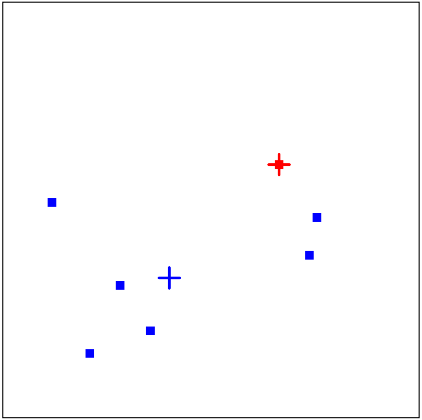
Optimization



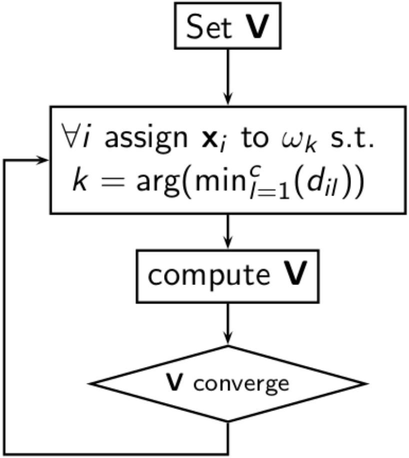


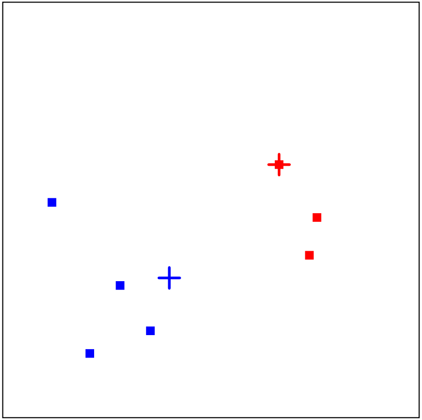
Optimization



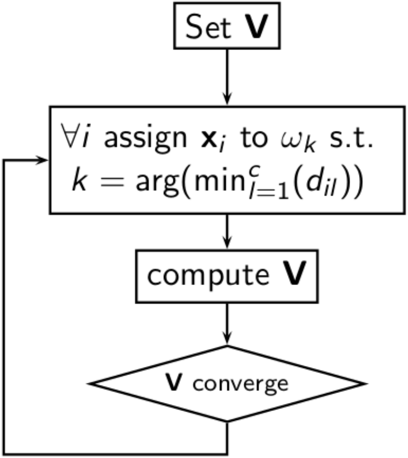


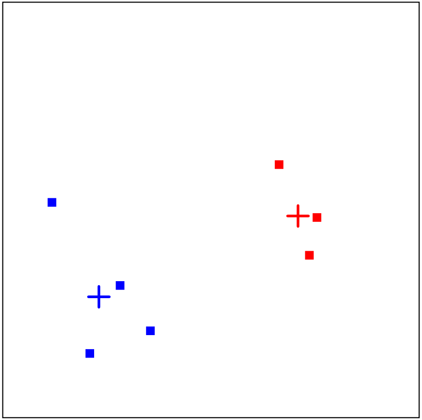
Optimization





Optimization



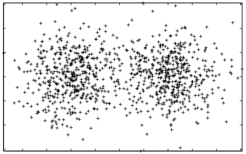


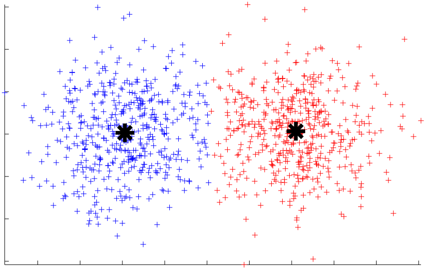
Strengths of *k*-means

Enable to deal with :

- globular shapes
- overlapped cluster

Original data

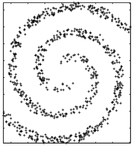




k-means (k=2)

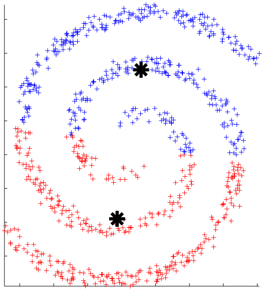
Limitations of *k*-means

Original data



Sensitive to :

- non geometrical shapes
- unbalanced cluster



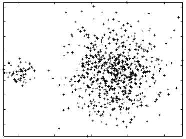
k-means (k=2)

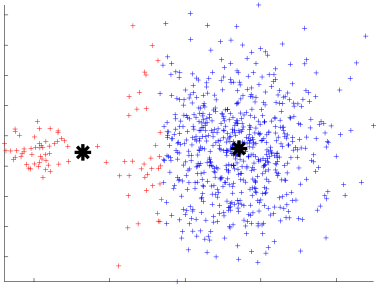
Limitations of *k*-means

Sensitive to :

- non geometrical shapes
- unbalanced cluster

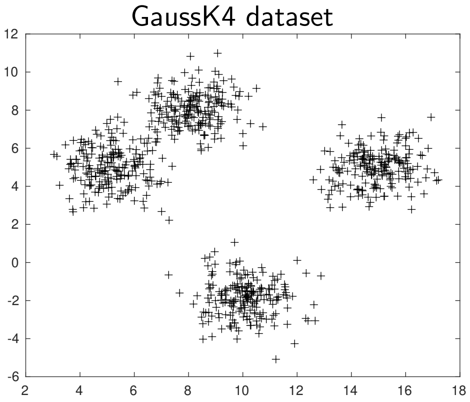
Original data





k-means (k=2)

Determination of the number of clusters



Basic idea

For c=1 to 10

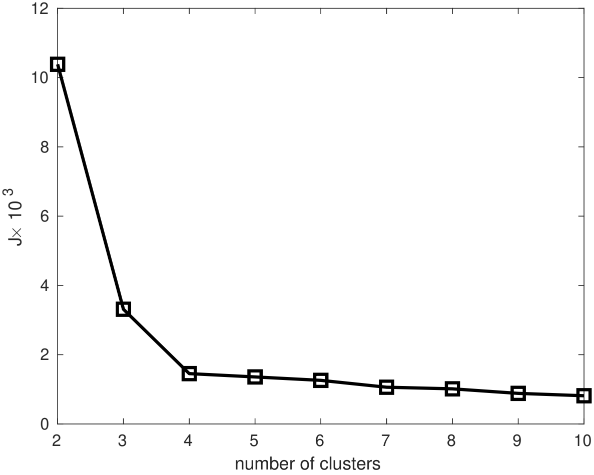
- run kmeans
- evaluate the partition

Plot evaluation measure vs number of clusters

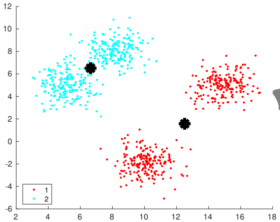
Possible measures

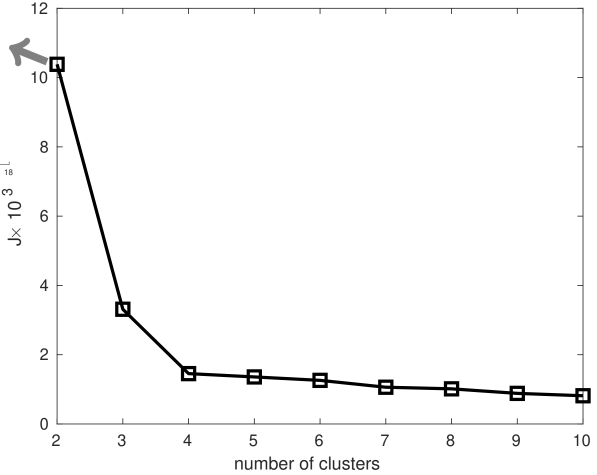
- Gap Statistic
- AIC / BIC criterions
- Silhouette coefficients

Determination of the number of clusters

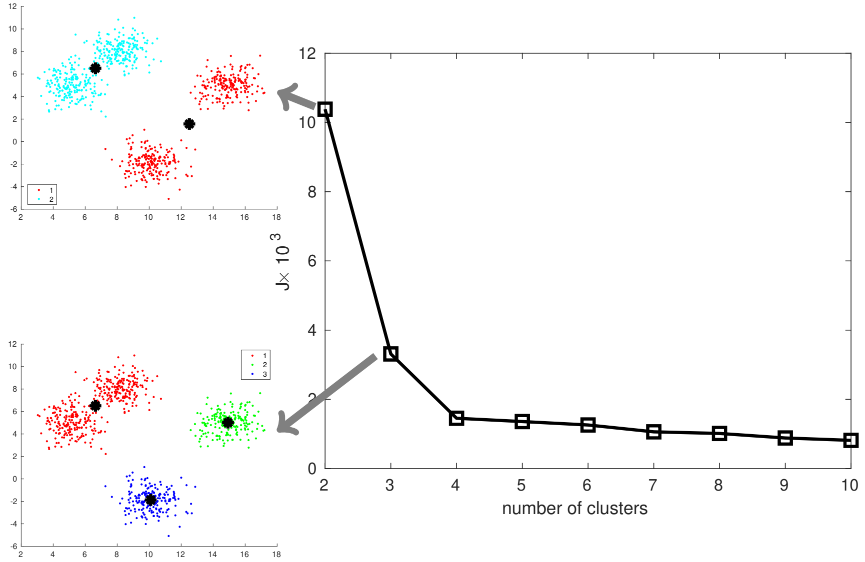


Determination of the number of clusters

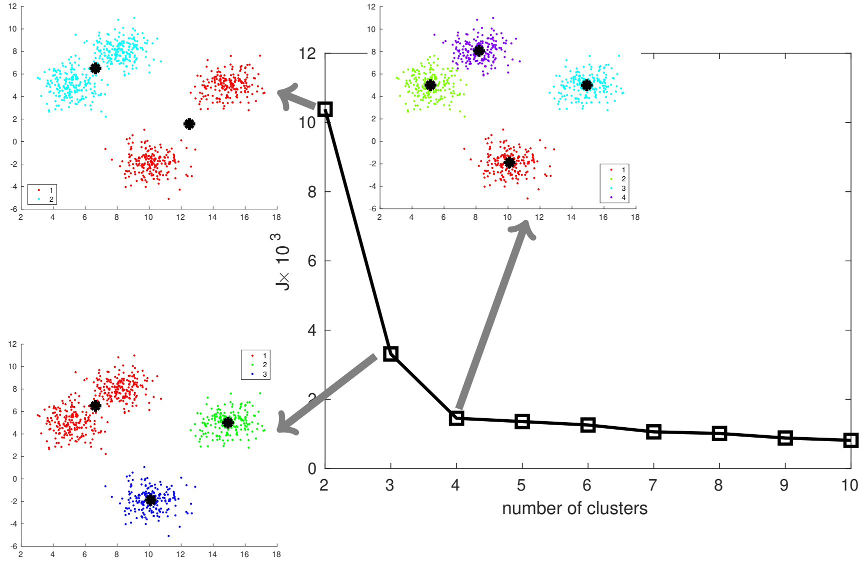




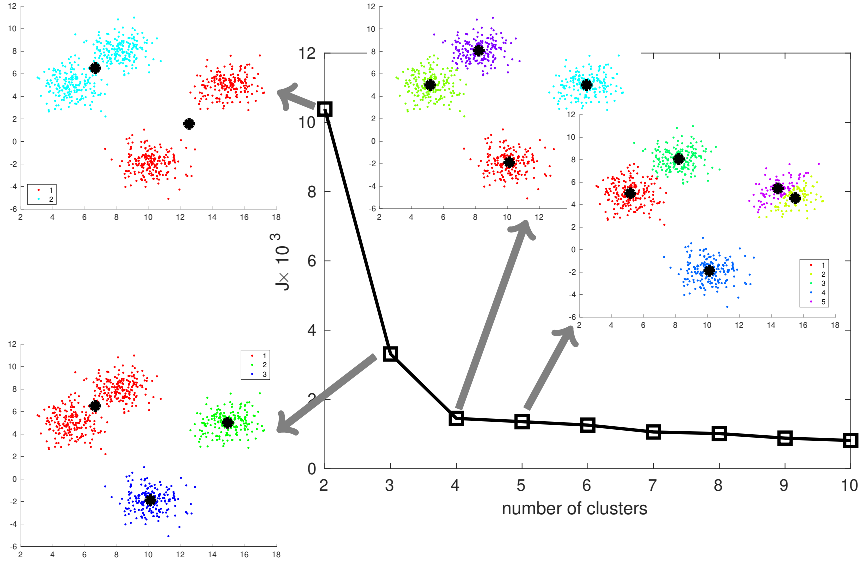
Determination of the number of clusters



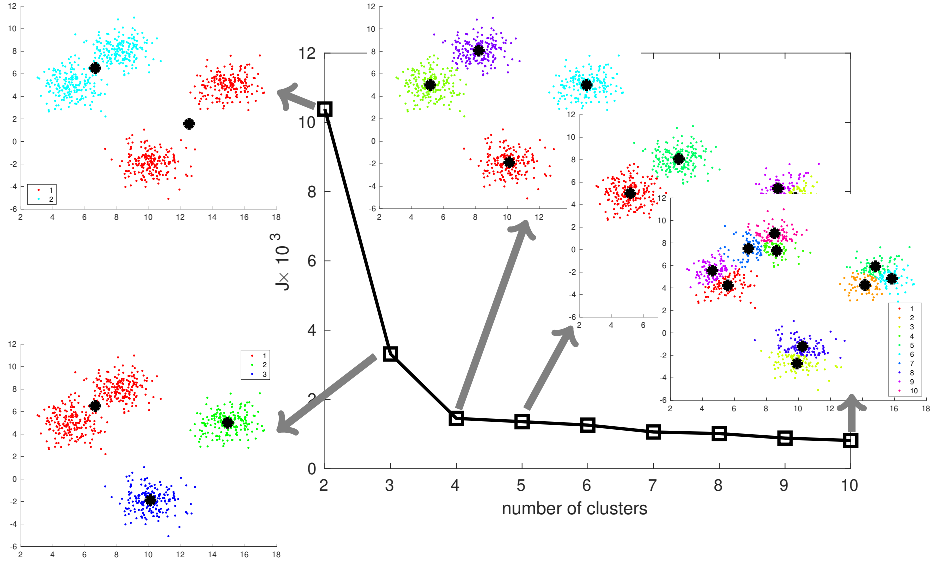
Determination of the number of clusters



Determination of the number of clusters

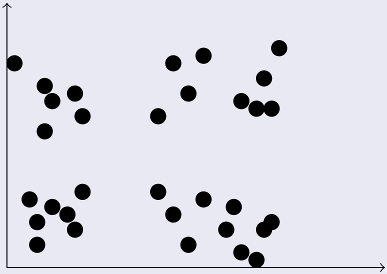


Determination of the number of clusters



The choice of initial centroids

The farthest-first method [?]

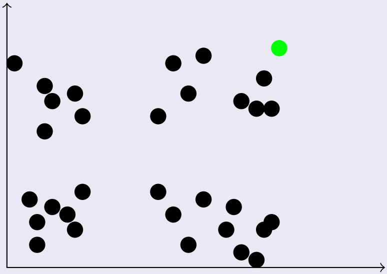


Other methods

- Nearest neighbor density
- Agglomerative hierarchical clustering

The choice of initial centroids

The farthest-first method [?]

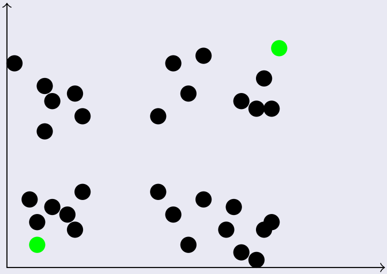


Other methods

- Nearest neighbor density
- Agglomerative hierarchical clustering

The choice of initial centroids

The farthest-first method [?]

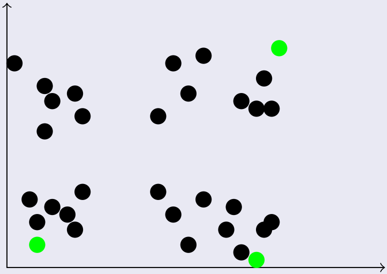


Other methods

- Nearest neighbor density
- Agglomerative hierarchical clustering

The choice of initial centroids

The farthest-first method [?]



Other methods

- Nearest neighbor density
- Agglomerative hierarchical clustering

Variants of k-means

Different choice of centroids

- k-Medoids
- k-Medians
- k-Modes

Automatically handling c

- Competitive Agglomeration
- Bisecting k-means

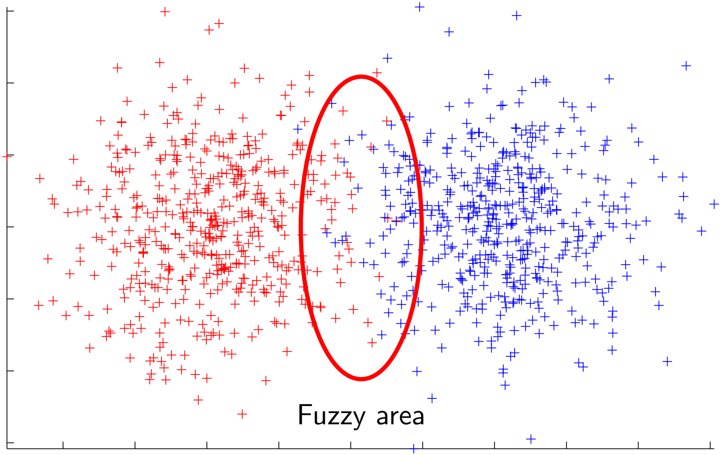
Returning non crisp partition

- Fuzzy c-means
- Evidential c-means

Algorithms returning fuzzy partition

Goal

Express uncertainty about the clustering result



Fuzzy c-means (FCM) [? ?]

Geometrical model

Each object x_i has a degree of membership in each cluster ω_k : u_{ik}

Aternate optimization

$$opt(u_{ik}) \rightleftarrows opt(\mathbf{v}_k)$$

Objective function

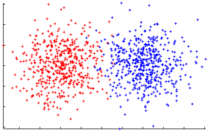
$$J_{FCM} = \sum_{i=1}^{N} \sum_{k=1}^{C} u_{ik}^{\beta} d_{ik}^{2}$$

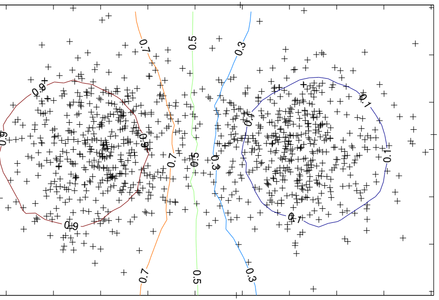
Subject to

$$\sum_{k=1}^{C} u_{ik} = 1 \text{ and } u_{ik} \geq 0 \quad \forall i, k$$

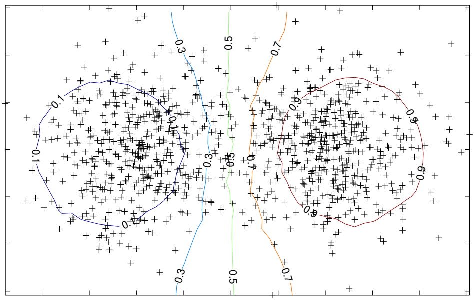
Fuzzy c-means

Original data





ω_1



ω_2

Variants of FCM

The Noise Clustering algorithm [?]

Add a noise cluster ω_* associated to a fixed δ :

$$J_{NC}(U, V) = \sum_{i=1}^{N} \sum_{k=1}^{C} u_{ik}^{\beta} d_{ik}^{2} + \sum_{k=1}^{C} u_{i*}^{\beta} \delta$$

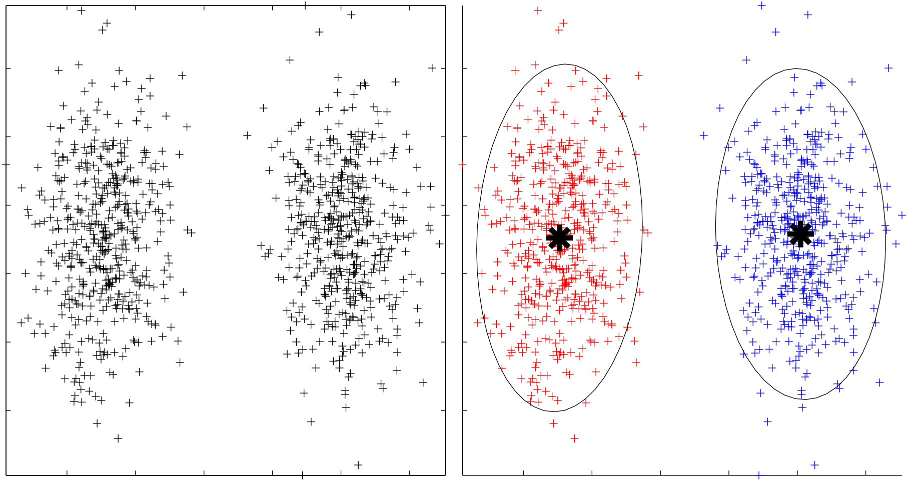
FCM with Mahalanobis distance [?]

Add a Mahalanobis distance between each $\mathbf{x}_i$ and ω_k :

$$d_{ik}^2 = (\mathbf{x}_i - \mathbf{v}_k)^\top S_k (\mathbf{x}_i - \mathbf{v}_k)$$

Minimize $J_{GK}(U, V, S)$ s.t. $|S_k| = 1 \quad \forall k = 1, C$

GK clustering [?]



original data GK algorithm

Outline

Measures

Two types of clustering validation measures :

External measures

External information are available (e.g. class label)

$\Rightarrow$ evaluation of the behavior of a clustering algorithm

$\Rightarrow$ expert assessment on few objects

Internal measures

No background knowledge

$\Rightarrow$ most of the real world applications

Internal measures

Depend on the output structure.

Goals

- Comparaison with different clustering algorithms
- Parameters determination for a specific algorithm
- Evaluation of the uncertainty of the clustering result

Internal measures

Depend on the output structure.

Goals

- Comparaison with different clustering algorithms
- Parameters determination for a specific algorithm
- Evaluation of the uncertainty of the clustering result

Example

Validity index for

- Crisp partitions
- Fuzzy partitions
- Evidential partitions

Measures for crisp partitions

Based on the combination of two criteria :

Compactness

Separation

Measures for crisp partitions

Based on the combination of two criteria :

Compactness

- Variance cluster

  $\Rightarrow$ low value $\equiv$ good density

Separation

Measures for crisp partitions

Based on the combination of two criteria :

Compactness

- Variance cluster

  $\Rightarrow$ low value $\equiv$ good density
- Intra-cluster distance
  - max or avg center-based distance
  - max or avg pairwise distance

$$d_{intra}(\omega_k) = \max_{x_i, x_j \in \omega_k} d_{ij}$$

$\Rightarrow$ d_{intra} low $\equiv$ good compactness

Separation

Measures for crisp partitions

Based on the combination of two criteria :

Compactness

- Variance cluster

  $\Rightarrow$ low value $\equiv$ good density
- Intra-cluster distance
  - max or avg center-based distance
  - max or avg pairwise distance

$$d_{intra}(\omega_k) = \max_{x_i, x_j \in \omega_k} d_{ij}$$

$\Rightarrow$ d_{intra} low $\equiv$ good compactness

Separation

- Inter-cluster distance
  - min or avg center-based distance
  - min or avg pairwise distance

$$d_{inter}(\omega_k, \omega_l) = \min_{\substack{x_i \in \omega_k, \\ x_j \in \omega_l}} d_{ij}$$

$\Rightarrow$ d_{inter} high $\equiv$ large separation

Dunn's indice

Definition

$$D = \min_{\omega_k} \left[\min_{\omega_l \neq \omega_k} \left(\frac{d_{inter}(\omega_k, \omega_l)}{\max\limits_{k=1\ldots c} d_{intra}(\omega_k)} \right) \right]$$

D should be maximized

Properties

Robust to :

- Various density
- Unequal size of cluster

Sensitive to :

- Noise
- Subclusters
- Arbitrary shapes

Silhouette index

Definition

$$S = \frac{1}{c} \sum_{\omega_k} \frac{1}{n_k} \sum_{x_i \in \omega_k} \frac{d_{ma}(x_i) - d_{intra}(x_i)}{\max(d_{ma}(x_i), d_{intra}(x_i))}$$

s.t. $d_{ma}(x_i) = \min_{l \neq k} d_{avg}(x_i, \omega_l)$

S should be maximized

Properties

Enable to handle :

- Noise
- Various density
- Unequal size of cluster

Affected by :

- Subclusters
- Arbitrary shapes

Other measures

RMSSD (Root Mean Square Standard Deviation)

- Consider only the compactness of a cluster
- Close to k-means objective function

Index

- A combination between Compactness and Separation
- Handle subclusters

CVNN

- Based on nearest neighbor
- Handle arbitrary shapes of cluster

And still several other measures [**?**].

Measures for fuzzy partitions

Only based on fuzzy memberships

Measure the amount of overlap between clusters

The partition coefficient

$$PC = \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{c} u_{ik}^2$$

Should be maximized

The partition entropy

$$PE = \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{c} u_{ik}^2 \log(u_{ik})$$

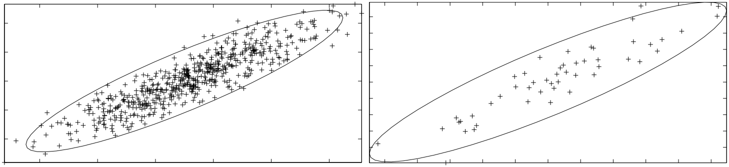
Should be minimized

Measures mainly used to determine c in FCM.

Measures for fuzzy partitions

Basic idea

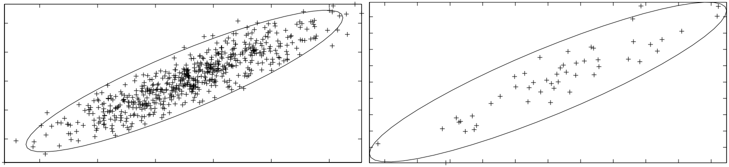
High concentration of points in a small spatial volume
$\Rightarrow$ measures compactness



Measures for fuzzy partitions

Basic idea

High concentration of points in a small spatial volume
$\Rightarrow$ measures compactness



Fuzzy HyperVolume

$$FHV = \sum_{k=1}^{c} \det(F_k)^{1/2}, \text{ s.t. } F_k = \frac{\sum_{i=1}^{n} u_{ik}^{\beta}(\mathbf{x}_i - \mathbf{v}_k)(\mathbf{x}_i - \mathbf{v}_k)^T}{\sum_{i=1}^{n} u_{ik}^{\beta}}$$

is the fuzzy covariance matrix of ω_k

FHV should be lowered

Outline

Conclusion

Most popular clustering algorithm

- hierarchical clustering
- k-means
- dbscan

Conclusion

Most popular clustering algorithm

- hierarchical clustering
- k-means
- dbscan

Advantages

Simple and fast

Disadvantages

Parameters to set (similarity notion, number of cluster, etc.)
$\Rightarrow$ Little background knowledge is a necessity !

Conclusion

Which clustering algorithm chose ?

Guideline

Dataset characteristics	Algorithms
unbalanced groups	single-link, DBSCAN
subclusters	hierarchical clustering
arbitrary shapes	DBSCAN
elliptic shapes	k-means, GK
overlapped cluster	k-means and variants

Conclusion

Other dataset characteristics

- High-dimensional data (p high)
- Big data (n high)
- categorical data
- uncertain data

Other interesting clustering techniques

- ensemble clustering
- constrained clustering

Conclusion

Clustering for special data

- documents
- multimedia
- time-series
- biological
- network

References I