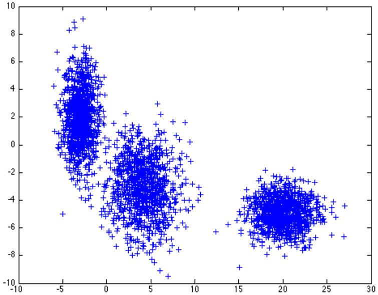
Clustering

Automatic set grouping of objects (clusters)

September 2021

Thomas Corpetti

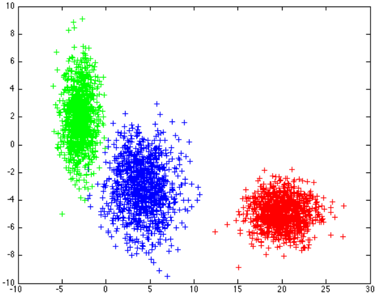
What is clustering ?



From a dataset : group homogeneous sets of data : clusters

- Group them by their similarity with respect to a model (generative methods)
- Separate them with respect to their dissimilarity (discriminative methods)

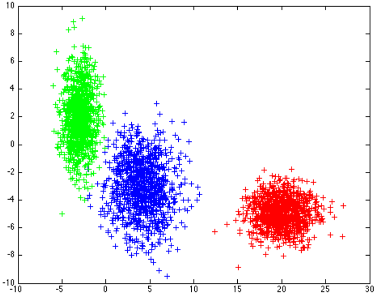
What is clustering ?



From a dataset : group homogeneous sets of data : clusters

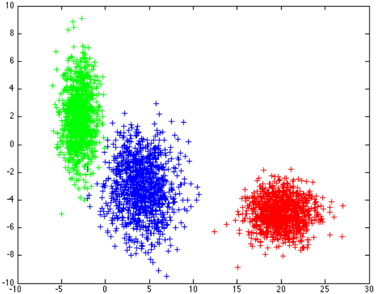
- Group them by their similarity with respect to a model (generative methods)
- Separate them with respect to their dissimilarity (discriminative methods)

What is a good grouping ?



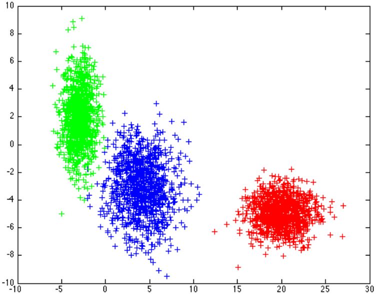
- Inside a cluster :

What is a good grouping ?



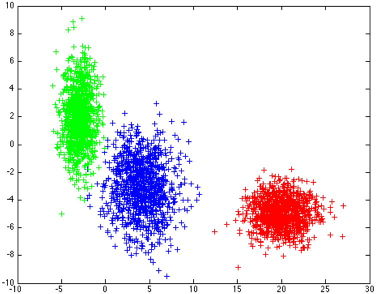
- Inside a cluster : high similarity

What is a good grouping ?



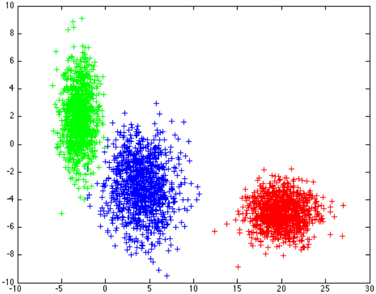
- Inside a cluster : high similarity
- Between clusters :

What is a good grouping ?



- Inside a cluster : high similarity
- Between clusters : low similarity (high dissimilarity)

What is a good grouping ?



- Inside a cluster : high similarity
- Between clusters : low similarity (high dissimilarity)

$\Longrightarrow$ Distance/metric of prime importance

Applications

Large range of applications

- Web : similar web-pages
- Social-networks : group of users
- Bio-informatics : identify species
- Marketing : types of clients
- Climatology : types of weather
- Image processing : homogeneous areas
- ...

Main families of clustering

- Centroid : create several clusters and evaluate their quality depending on some centroids (k-means, k-medoids, PAM, ...)
- Hierarchical : group in a hierarchical way data (AGNES, DIANA, ...)
- Density : rely on the adequacy of data with respect to a certain density (DBSCAN)
- Model-based : one model for each cluster
- Spectral : based on a graph-representation of data

Main families of clustering

- Centroid : create several clusters and evaluate their quality depending on some centroids (k-means, k-medoids, PAM, ...)
- Hierarchical : group in a hierarchical way data (AGNES, DIANA, ...)
- Density : rely on the adequacy of data with respect to a certain density (DBSCAN)
- Model-based : one model for each cluster
- Spectral : based on a graph-representation of data

General ideas

Main families of clustering

- Centroid : create several clusters and evaluate their quality depending on some centroids (k-means, k-medoids, PAM, ...)
- Hierarchical : group in a hierarchical way data (AGNES, DIANA, ...)
- Density : rely on the adequacy of data with respect to a certain density (DBSCAN)
- Model-based : one model for each cluster
- Spectral : based on a graph-representation of data

Main families of clustering

- Centroid : create several clusters and evaluate their quality depending on some centroids (k-means, k-medoids, PAM, ...)
- Hierarchical : group in a hierarchical way data (AGNES, DIANA, ...)
- Density : rely on the adequacy of data with respect to a certain density (DBSCAN)
- Model-based : one model for each cluster
- Spectral : based on a graph-representation of data

Main families of clustering

- Centroid : create several clusters and evaluate their quality depending on some centroids (k-means, k-medoids, PAM, ...)
- Hierarchical : group in a hierarchical way data (AGNES, DIANA, ...)
- Density : rely on the adequacy of data with respect to a certain density (DBSCAN)
- Model-based : one model for each cluster
- Spectral : based on a graph-representation of data

Outline

1 Introduction

2 Partitioning
- K-means
- Evaluation / Computations / Characterisation of clusters
- K-medoids

3 Hierarchical Clustering
- Principes
- Agglomerative
- Divisive

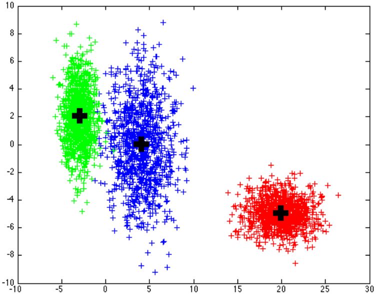
4 Density based clustering
- Principles

5 Model based clustering
- Principles

6 Spectral clustering (graph-based)
- Principles

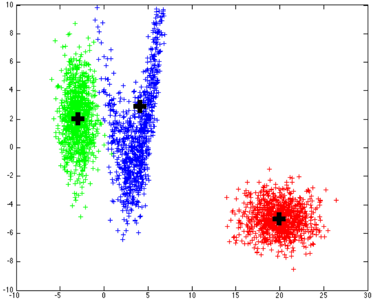
Centroid

- Construct k clusters from n objects
- Many criteria
  - k-means (MacQueen'67) : rely on the center to create clusters



Centroid

- Construct k clusters from n objects
- Many criteria
  - k-means (MacQueen'67) : rely on the center to create clusters
  - k-medoids or PAM (Partition around medoids) : rely on specific data to create clusters



Main idea

- Create k-partitions : each data is associated with the closest center of partition
- Also called quantification algorithm of Lloyd-Max

- We start from a **data matrix** X of dimension $N \times P$ (N points of dimension P)
- Le number of clusters k is a *hyperparameter*, supposed to be known

Main idea

- Create k-partitions : each data is associated with the closest center of partition
- Also called quantification algorithm of Lloyd-Max

- We start from a **data matrix** X of dimension $N \times P$ (N points of dimension P)
- Le number of clusters k is a *hyperparameter*, supposed to be known

K-means

Main idea

- Create k-partitions : each data is associated with the closest center of partition
- Also called quantification algorithm of Lloyd-Max

- We start from a **data matrix** X of dimension $N \times P$ (N points of dimension P)
- Le number of clusters k is a *hyperparameter*, supposed to be known

K-means

- Formalisation : find the partition $\mathcal{S}^* = \{S1, ..., S_k\}$ such that

$$\mathcal{S}^* = \arg\min_{\mathcal{S}} \sum_{i=1}^{k} \sum_{x_j \in S_i} \|x_j - \mu_i\|^2$$

with μ_i the average of points in S_i

- $\Longrightarrow$ Tricky optimization problem

K-means

- Formalisation : find the partition $\mathcal{S}^* = \{S1, ..., S_k\}$ such that

$$\mathcal{S}^* = \arg\min_{\mathcal{S}} \sum_{i=1}^{k} \sum_{x_j \in S_i} \|x_j - \mu_i\|^2$$

  with μ_i the average of points in S_i
- $\Longrightarrow$ Tricky optimization problem

Standard algorithm

1. Iteration $p = 0$: find k initial means $\mu_i^p, \forall i = 1...k$ (usually randomly)
2. For each iteration p, until convergence
   1. Each point x_j is assigned to partition S_i^p if its distance with center μ_i^p is minimal :

$$S_i^p = \left\{ x_j : \|x_j - \mu_i^p\| \leqslant \|x_j - \mu_{i^*}^p\|,\ \forall i^* = 1...k \right\}$$

   2. Update the mean of each partition

$$\mu_{i^*}^{p+1} = \frac{1}{\mathrm{card}(S_i^p)} \sum_{x_j \in S_i^p} x_j$$

   3. $p = p + 1$

Note : the convergence can be low $\Longrightarrow$ find some heuristics

Standard algorithm

1. Iteration $p=0$: find k initial means $\mu_i^p, \forall i = 1...k$ (usually randomly)
2. For each iteration p, until convergence
    1. Each point x_j is assigned to partition S_i^p if its distance with center μ_i^p is minimal :

    $$S_i^p = \left\{ x_j : \|x_j - \mu_i^p\| \leqslant \|x_j - \mu_{i^*}^p\|, \ \forall i^* = 1...k \right\}$$

    2. Update the mean of each partition

    $$\mu_{i^*}^{p+1} = \frac{1}{\text{card}(S_i^p)} \sum_{x_j \in S_i^p} x_j$$

    3. $p = p+1$

Note : the convergence can be low $\Longrightarrow$ find some heuristics

K-means : some notes

Pros

- Cost-function always decreasing along iterations
- Simple and efficient

Coins

- Discontinuous data (what is a "centroid" in this case) ?
- How to fix the number of clusters ?

Variations, alternative approaches

- Monte-Carlo
- Fix k with cross validation

In each case : centers are not guarantee to be part of the dataset

K-means : some notes

Pros

- Cost-function always decreasing along iterations
- Simple and efficient

Coins

- Discontinuous data (what is a "centroid" in this case) ?
- How to fix the number of clusters ?

Variations, alternative approaches

- Monte-Carlo
- Fix k with cross validation

In each case : centers are not guarantee to be part of the dataset

K-means : some notes

Pros

- Cost-function always decreasing along iterations
- Simple and efficient

Coins

- Discontinuous data (what is a “centroid” in this case) ?
- How to fix the number of clusters ?

Variations, alternative approaches

- Monte-Carlo
- Fix k with cross validation

In each case : centers are not guarantee to be part of the dataset

K-means : some notes

Pros

- Cost-function always decreasing along iterations
- Simple and efficient

Coins

- Discontinuous data (what is a "centroid" in this case) ?
- How to fix the number of clusters ?

Variations, alternative approaches

- Monte-Carlo
- Fix k with cross validation

In each case : centers are not guarantee to be part of the dataset

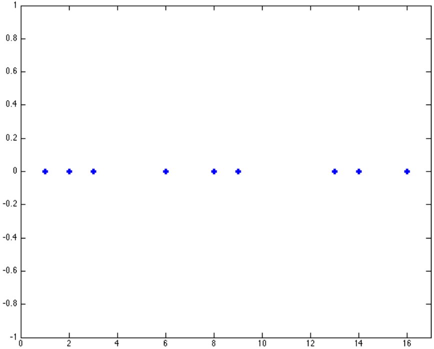
K-means : exercice

Apply a k-mean with 3 clusters for the $1D$ dataset :

$$P = \{1, 2, 3, 6, 8, 9, 13, 14, 16\}$$

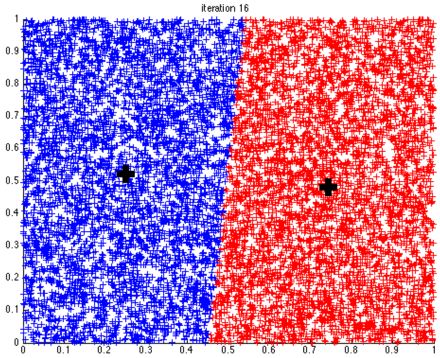
With initial means

$$\mu_1^0 = 1, \mu_2^0 = 2, \mu_3^0 = 3$$



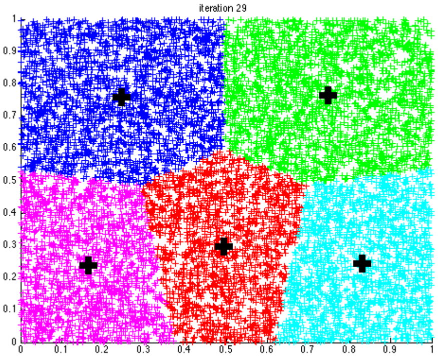
K-means : illustrations

Uniform random set : 2 partitions



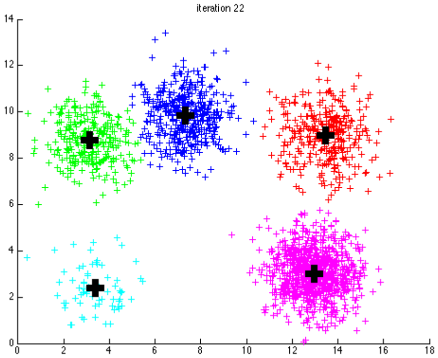
K-means : illustrations

Uniform random set : 5 partitions



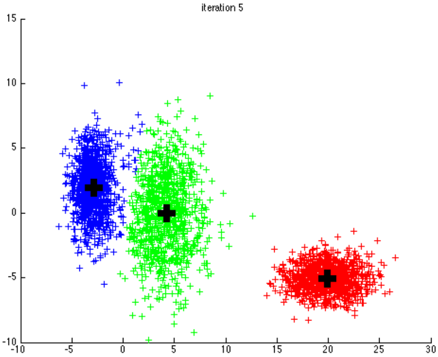
K-means : illustrations

Point cloud with 5 clusters



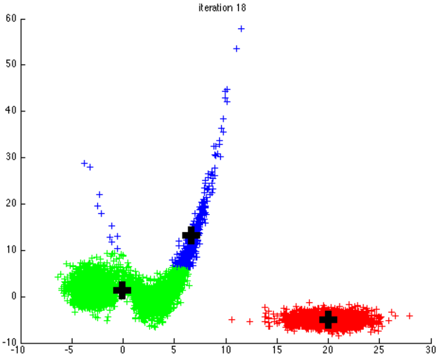
K-means : illustrations

Point cloud with 3 clusters



K-means : illustrations

Point cloud with 3 clusters



What about images

create a point cloud with R, G, B coef, and may be spatial coordinates

R	G	B	X	Y
212	80	12	180	341
222	34	49	180	342
190	23	423	180	343
209	12	190	180	344
...	...	...	...	...
...	...	...	...	...

To impose spatial consistency

- one can add X, Y coordinates ;
- one can also filter images

What about images

create a point cloud with R, G, B coef, and may be spatial coordinates

R	G	B	X	Y
212	80	12	180	341
222	34	49	180	342
190	23	423	180	343
209	12	190	180	344
...	...	...	...	...
...	...	...	...	...

To impose spatial consistency

- one can add X, Y coordinates ;
- one can also filter images

Computation on clusters

How to characterise a cluster ? How to compare partitions ?

- Characterisation :
  - Inertia inside a cluster S_i composed of points $x_1, ..., x_M$ of M :

$$\mathcal{I}_i = \sum_{k=1}^{M} p(x_k) d(x_k, \mu_i)$$

with

- $d(.,.)$: a distance function ;
- μ_i : center of the cluster ;
- $p(x_k)$: probability of point x_k

Note : If all points have similar probability, $p(x_k) = 1/M$

Computation on clusters

How to characterise a cluster ?

- Characterisation :
  - Inertia inside a cluster S_i composed of points $x_1, ..., x_M$ of M :

$$\mathcal{I}_i = \sum_{k=1}^{M} p(x_k) d(x_k, \mu_i)$$

    - $d(.,.)$: a distance function ;
    - μ_i : center of the cluster ;
    - $p(x_k)$: probability of point x_k

  Note : If all points have similar probability, $p(x_k) = 1/M$

Computation on clusters

- Characterisation :
  - "Intra-class" intertia (of all clusters S) :

$$\mathcal{I}ntra = \sum_{k=1}^{|S|} \mathcal{I}_k$$

$\Longrightarrow$ sum of all cluster intertia

  - "Inter-class" intertia (between all clusters S) :

$$\mathcal{I}nter = \sum_{k=1}^{|S|} \frac{1}{|\text{card}(S_k)|} d(\mu_k, \mu)$$

with

- μ : global average of points

Computation on clusters

- Characterisation :
  - "Intra-class" intertia (of all clusters S) :

$$\mathcal{I}ntra = \sum_{k=1}^{|S|} \mathcal{I}_k$$

$\Longrightarrow$ sum of all cluster intertia

  - "Inter-class" intertia (between all clusters S) :

$$\mathcal{I}nter = \sum_{k=1}^{|S|} \frac{1}{|\text{card}(S_k)|} d(\mu_k, \mu)$$

with

- μ : global average of points

How to compare partitions

- Let's take all pair of points (x_m, x_n) and let's have a look of their partitions with two methods. Four possibilities :

Class with method 1 \ Class with method 2	same	different
same	a	b
different	c	d

- Rand index :

$$R = \frac{a+d}{a+b+c+d} = \frac{a+d}{C_2^N}$$

How to compare partitions

- Let's take all pair of points (x_m, x_n) and let's have a look of their partitions with two methods. Four possibilities :

Class with method 1 \ Class with method 2	same	different
same	a	b
different	c	d

- Rand index :

$$R = \frac{a+d}{a+b+c+d} = \frac{a+d}{C_2^N}$$

Exercise

- We have a set of 5 points $\{x_1, x_2, x_3, x_4, x_5\}$, and 2 clustering methods gave the following partitions :

  - $S_1 = \{1, 1, 2, 2, 2\}$
  - $S_2 = \{1, 2, 2, 1, 2\}$

- What is the Rand index ?

K-medoids

General ideas

- Find most representative centroids (medoids) in the cluster
- Principle : iteratively replace medoids if the global distance is reduced

- More robust with respect to outliers

K-medoids

General ideas

- Find most representative centroids (medoids) in the cluster
- Principle : iteratively replace medoids if the global distance is reduced

- More robust with respect to outliers

K-medoids

Standard algorithm

1. Iteration $p = 0$: select k initial centroids $\mu_i^p, \forall i = 1...k$ (for example randomly inside the dataset)
2. For each iteration p, until convergence
   1. Each point is assigned to partition S_i^p if its distance with medoid μ_i^p is minimal :

      $$S_i^p = \left\{ x_j : \|x_j - \mu_i^p\| \leqslant \|x_j - \mu_{i^*}^p\|, \ \forall i^* = 1...k \right\}$$

      Note : exactly the same than k-means
   2. For each class i, choose randomly some non-medoids (or all) μ_j^p
      - Compute the replace cost of μ_i^p by μ_j^p

      $\Longrightarrow$ If the cost is lower, μ_i^p is replaced by μ_j^p

Note : the convergence can be low $\Longrightarrow$ find some heuristics

K-medoids

Standard algorithm

1. Iteration $p = 0$: select k initial centroids $\mu_i^p, \forall i = 1...k$ (for example randomly inside the dataset)
2. For each iteration p, until convergence
   1. Each point is assigned to partition S_i^p if its distance with medoid μ_i^p is minimal :

      $$S_i^p = \left\{ x_j : \|x_j - \mu_i^p\| \leqslant \|x_j - \mu_{i^*}^p\|, \ \forall i^* = 1...k \right\}$$

      Note : exactly the same than k-means
   2. For each class i, choose randomly some non-medoids (or all) μ_j^p
      - Compute the replace cost of μ_i^p by μ_j^p

      $\Longrightarrow$ If the cost is lower, μ_i^p is replaced by μ_j^p

Note : the convergence can be low $\Longrightarrow$ find some heuristics

K-medoids : some notes

Pros

- Cost-function always decreasing along iterations
- Simple and efficient
- The cluster centroids (medoids) are interpretable

Coins

- How to fix the number of clusters ?

Variations, alternative approaches

- Monte-Carlo
- Fix k with cross validation

K-medoids : some notes

Pros

- Cost-function always decreasing along iterations
- Simple and efficient
- The cluster centroids (medoids) are interpretable

Coins

- How to fix the number of clusters ?

Variations, alternative approaches

- Monte-Carlo
- Fix k with cross validation

K-medoids : some notes

Pros

- Cost-function always decreasing along iterations
- Simple and efficient
- The cluster centroids (medoids) are interpretable

Coins

- How to fix the number of clusters ?

Variations, alternative approaches

- Monte-Carlo
- Fix k with cross validation

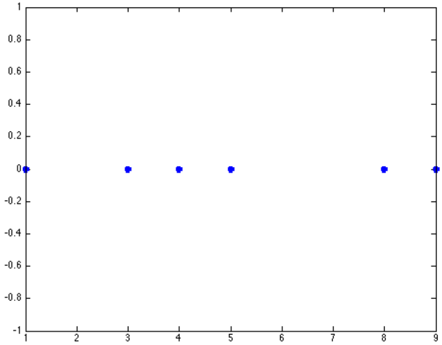
K-medoids : exercise

Apply the K-medoids method with 2 clusters on the $1D$ dataset :

$$P = \{1, 3, 4, 5, 8, 9\}$$

With initial medoids

$$\mu_1^0 = 1, \mu_2^0 = 8$$



Outline

1 Introduction

2 Partitioning
- K-means
- Evaluation / Computations / Characterisation of clusters
- K-medoids

3 Hierarchical Clustering
- Principes
- Agglomerative
- Divisive

4 Density based clustering
- Principles

5 Model based clustering
- Principles

6 Spectral clustering (graph-based)
- Principles

Hierarchical Clustering

Two main types :

1. Clustering agglomerative (bottom-up) : all points are in distincts clusters that are merged based on some criteria
2. Clustering divisive : all points are in a single cluster which is split depending on some criteria

$\Longrightarrow$ Important question : what is the distance between two clusters. Is it :

- The smallest distance between points of each clusters
- The largest distance between points of each clusters
- The distance between the mean of each cluster
- The average between all pairwise distances
- ...

Hierarchical Clustering

Two main types :

1. Clustering agglomerative (bottom-up) : all points are in distincts clusters that are merged based on some criteria
2. Clustering divisive : all points are in a single cluster which is split depending on some criteria

$\Longrightarrow$ Important question : what is the distance between two clusters. Is it :

- The smallest distance between points of each clusters
- The largest distance between points of each clusters
- The distance between the mean of each cluster
- The average between all pairwise distances
- ...

Hierarchical Agglomerative Clustering

Starting point : distance matrix

- From a set of N points of dimension P, the (symmetric) distance matrix is :

$$M = \begin{bmatrix} 0 & D(1,2) & . & D(1,N) \\ D(2,1) & 0 & . & D(2,N) \\ . & . & 0 & . \\ D(N,1) & D(N,2) & . & 0 \end{bmatrix}$$

Algorithm

1. Group all distances lower than a given threshold together
2. Recompute a distance matrix with the new dataset

Stop criteria to define

Main family : AGNES (AGglomerative NESting)

Hierarchical Agglomerative Clustering

Starting point : distance matrix

- From a set of N points of dimension P, the (symmetric) distance matrix is :

$$M = \begin{bmatrix} 0 & D(1,2) & . & D(1,N) \\ D(2,1) & 0 & . & D(2,N) \\ . & . & 0 & . \\ D(N,1) & D(N,2) & . & 0 \end{bmatrix}$$

Algorithm

1. Group all distances lower than a given threshold together
2. Recompute a distance matrix with the new dataset

Stop criteria to define

Main family : AGNES (AGglomerative NESting)

Hierarchical Agglomerative Clustering

Starting point : distance matrix

- From a set of N points of dimension P, the (symmetric) distance matrix is :

$$M = \begin{bmatrix} 0 & D(1,2) & . & D(1,N) \\ D(2,1) & 0 & . & D(2,N) \\ . & . & 0 & . \\ D(N,1) & D(N,2) & . & 0 \end{bmatrix}$$

Algorithm

1. Group all distances lower than a given threshold together
2. Recompute a distance matrix with the new dataset

Stop criteria to define

Main family : AGNES (AGglomerative NESting)

Hierarchical Divisive Clustering

Algorithm

1. Start from a large cluster that embeds all data
2. Divise it if not consistent

How to divise it ? $\Longrightarrow$ Less used algorithm

Main family : DIANA (DIvise ANAlysis)

Hierarchical Divisive Clustering

Algorithm

1. Start from a large cluster that embeds all data
2. Divise it if not consistent

How to divise it ? $\Longrightarrow$ Less used algorithm

Main family : DIANA (DIvise ANAlysis)

Outline

1. Introduction
2. Partitioning
   - K-means
   - Evaluation / Computations / Characterisation of clusters
   - K-medoids
3. Hierarchical Clustering
   - Principes
   - Agglomerative
   - Divisive
4. Density based clustering
   - Principles
5. Model based clustering
   - Principles
6. Spectral clustering (graph-based)
   - Principles

Density based clustering

Basic assumptions

1. A cluster is a “dense” area
2. Data are composed of various clusters separated by less-dense areas

Principles

1. Two points are potentially in the same cluster if the are separated by a distance less than a radius fixed by the user
2. Each cluster has a minimal size

Main problem : in case of too much noise : all points in the same cluster

Pros : no need to compute means

Density based clustering

Basic assumptions

1. A cluster is a "dense" area
2. Data are composed of various clusters separated by less-dense areas

Principles

1. Two points are potentially in the same cluster if the are separated by a distance less than a radius fixed by the user
2. Each cluster has a minimal size

Main problem : in case of too much noise : all points in the same cluster

Pros : no need to compute means

Density based clustering

Basic assumptions

1. A cluster is a “dense” area
2. Data are composed of various clusters separated by less-dense areas

Principles

1. Two points are potentially in the same cluster if the are separated by a distance less than a radius fixed by the user
2. Each cluster has a minimal size

Main problem : in case of too much noise : all points in the same cluster

Pros : no need to compute means

Density based clustering

Basic assumptions

1. A cluster is a “dense” area
2. Data are composed of various clusters separated by less-dense areas

Principles

1. Two points are potentially in the same cluster if the are separated by a distance less than a radius fixed by the user
2. Each cluster has a minimal size

Main problem : in case of too much noise : all points in the same cluster

Pros : no need to compute means

Outline

1 Introduction
2 Partitioning
- K-means
- Evaluation / Computations / Characterisation of clusters
- K-medoids

3 Hierarchical Clustering
- Principes
- Agglomerative
- Divisive

4 Density based clustering
- Principles

5 Model based clustering
- Principles

6 Spectral clustering (graph-based)
- Principles

Model based clustering

Basic assumptions

1. We know the number k of clusters
2. Each er has a model

Principles

- Each point is associated with the most likely model

Main difficulty : get such models

Model based clustering

Basic assumptions

1. We know the number k of clusters
2. Each er has a model

Principles

- Each point is associated with the most likely model

Main difficulty : get such models

Model based clustering

Basic assumptions

1. We know the number k of clusters
2. Each er has a model

Principles

- Each point is associated with the most likely model

Main difficulty : get such models

Outline

1 Introduction

2 Partitioning
- K-means
- Evaluation / Computations / Characterisation of clusters
- K-medoids

3 Hierarchical Clustering
- Principes
- Agglomerative
- Divisive

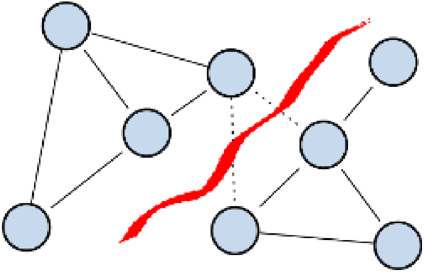
4 Density based clustering
- Principles

5 Model based clustering
- Principles

6 Spectral clustering (graph-based)
- Principles

Spectral clustering

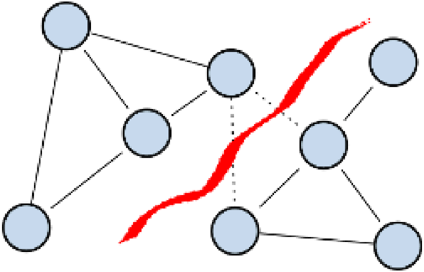
Data are represented on graphes



- The distance between each vertices is computed based on the similarity between points
- Perform “cut” on graphes

Spectral clustering

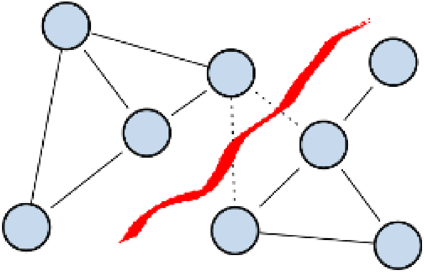
Data are represented on graphes



- The distance between each vertices is computed based on the similarity between points
- Perform "cut" on graphes

Spectral clustering

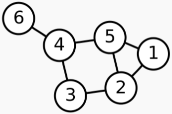
Data are represented on graphes



- The distance between each vertices is computed based on the similarity between points
- Perform "cut" on graphes

Graph theory

source : Wikipedia

Graphe	Représentation par une matrice d'adjacence	Représentation par une matrice laplacienne (non normalisée)
	$\begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 2 & -1 & 0 & 0 & -1 & 0 \\ -1 & 3 & -1 & 0 & -1 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & -1 \\ -1 & -1 & 0 & -1 & 3 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \end{pmatrix}$

- Based on Adjency and Laplacian matrix, one can characterise the structure of data