Introduction à l'Extraction de Connaissances

Chapitre V

Clustering

Alexandre Saidi
Ecole Centrale de Lyon
Département Mathématiques-Informatique
UMR LIRIS - CNRS

Novembre 2017

Introduction

- **Clustering : une technique non supervisée**

  A considérer versus la classification supervisée où :
  - On connaît (et exploite) les classes par avance
  - On utilise ces ('vraies') classes pour calculer l'écart entre le modèle de prédiction et la connaissance a priori des classes.

- **Clustering** = regrouper les données en plusieurs groupes (= *clusters*)

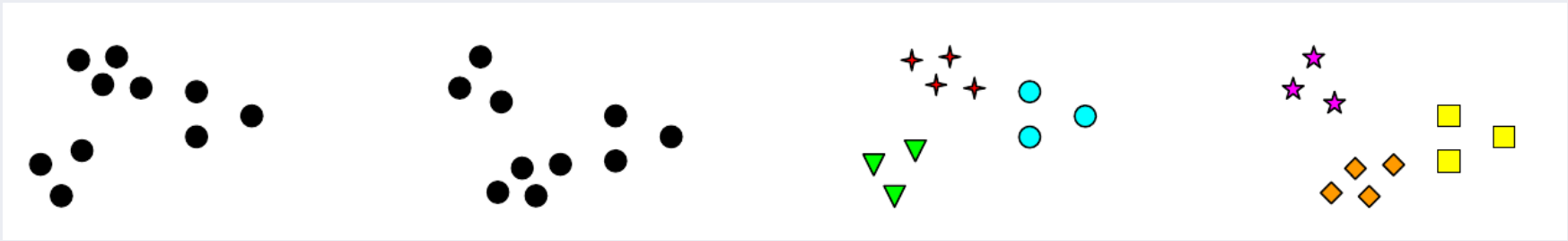
- Dans Clustering : les clusters (groupes) sont a priori inconnus.

- Chaque groupe doit être **homogène** et se distinguer des autres groupes.
  - Minimiser les *intra-distances*, maximiser les *inter-distances*
  - Il arrive d'avoir des groupes qui se recouvrent ➙ probabilités

☞ On peut prendre (ou pas) en compte la classe des observations :
  - Si on en tient compte : TOUS les attributs (classe comprise) sont classés
  - Si on écarte la classe : on l'utilise *a posteriori* pour vérifier si elles représentent les groupes (permet de calculer l'erreur).

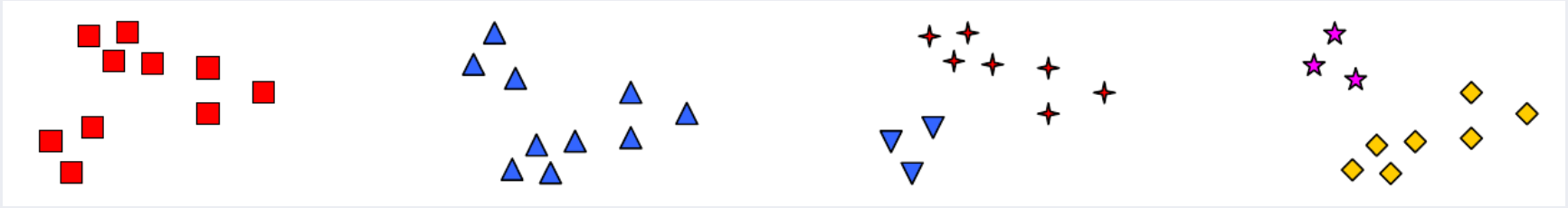
Introduction (suite)

La notion de Cluster est ambigüe :



Combien de clusters ?

Six clusters ?



Deux clusters ?

Quatre clusters ?

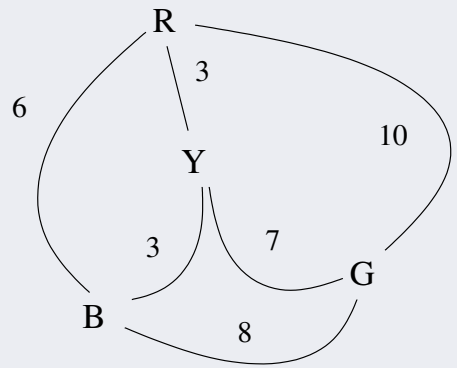
Mesures de similarité et distance

Quelques définitions

- Soit O un ensemble d'instances
- Une fonction $d : O \times O \rightarrow \mathbb{R}$ définit une **distance** si pour tout $x, y, z \in O$:
  - $d(x, y) = d(y, x)$
  - $d(x, y) = 0 \ ssi \ x = y$
  - $d(x, y) \geq 0$
  - $d(x, z) \leq d(x, y) + d(y, z)$

Exemple : Soit $O = \{R, B, G, Y\}$

d	R	B	G	Y
R	0	6	10	3
B	6	0	8	3
G	10	8	0	7
Y	3	3	7	0



Définitions : le centre d'un Cluster

- Soit $C \subseteq O$ (O ensemble de points / d'instances).

- Le *centre* (le **représentant**) de C peut avoir l'une des définitions suivantes :

  1. Un objet $o \in O$ pour lequel $\sum_{x \in C} d(o, x)$ est minimal

     ➛ on appelle o le **Centroïde** *(o n'est pas forcément observé dans* $\mathbf{C}$*)*

  2. Un objet $c \in C$ pour lequel $\sum_{x \in C} d(c, x)$ est minimal

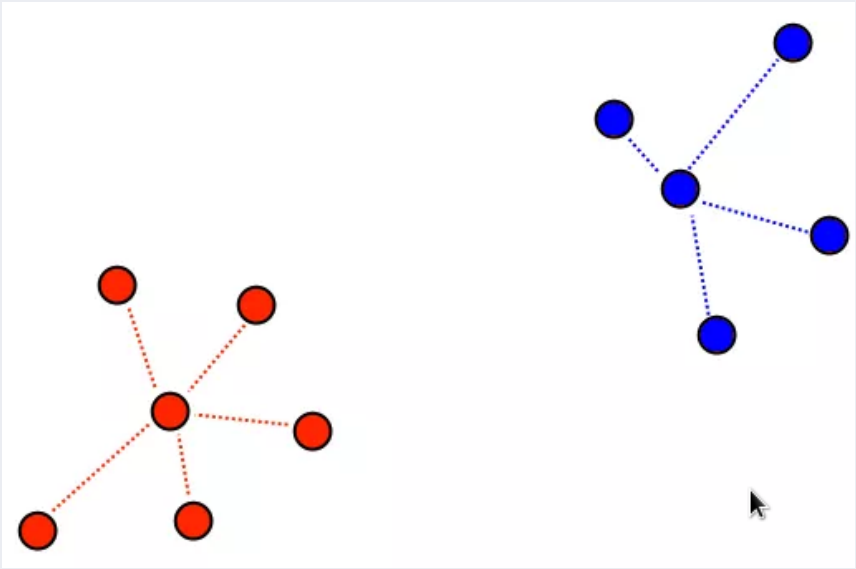
     ➛ on appelle c le **Médoïde** (ou *médianoïde*). *(cf. MST de c dans C)*

- Comme la différence entre la *médiane* et la *moyenne*

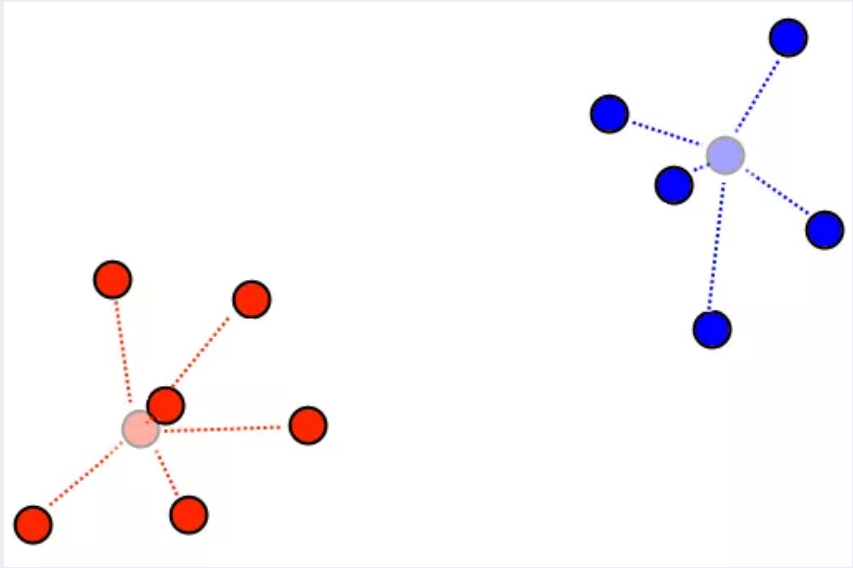
Définitions : le centre d'un Cluster (suite)

Différence entre le *Médoïde* et le *Centroïde* :

- le *Médoïde* est un point du cluster,
- le *Centroïde* (cf. moyenne) ne l'est pas forcément.



Médoïde : un point du nuage



Centroïde : une moyenne (pas forcément un point)

Définitions : le centre d'un Cluster (suite)

Exemple : Soit $O = \{Y, R, B, G\}$ et $C \subseteq O, C = \{R, B, G\}$ $(Y \notin C)$

➙ Distance de chaque point de C aux autres points de C (les lignes) :

d	R	B	G	Y
R	0	6	10	3
B	6	0	8	3
G	10	8	0	7
Y	3	3	7	0

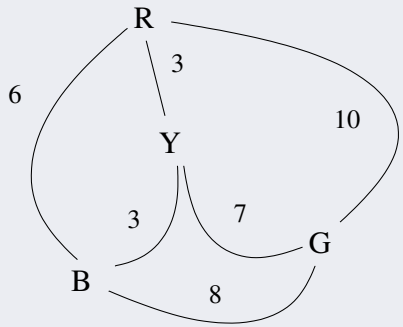
R : $d(R,R) + d(R,B) + d(R,G) = 16$
B : $d(B,R) + d(B,B) + d(B,G) = 14$
G : $d(G,R) + d(G,B) + d(G,G) = 18$

➙ Le Médoïde B est l'objet le plus *central* de C.

☞ N.B. : $d(Y,R) + d(Y,B) + d(Y,G) = 13$

➙ Y est le Médoïde de $O = \{R, B, G, Y\}$

- Pour $C = \{R, B, G\}$, on a $d(B,R) + d(B,G) = 6 + 8 = 14$: le centroïde serait à la distance $14/2 = 7$
  ➙ $Y \in O$, $Y \notin C$ pourrait être le centroïde de C.
- N.B. : ici la distance simple est appelée distance linéaire (voir ../..)



Distances classiques dans $\mathbb{R}^n$

Exemples de distances en $\mathbb{R}^n$

- Soient $\vec{x} = (x_1, x_2, ..., x_n)$ et $\vec{y} = (y_1, y_2, ..., y_n)$ deux points dans $\mathbb{R}^n$
  - Distance **Manhattan** (ou "city block") : $d_{Manh}(\vec{x}, \vec{y}) = \sum_{i=1}^{n} |x_i - y_i|$
  - Distance **Euclidienne** : $d_{Eucl}(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$ (Dite Norme/Distance *L2*)
  - Distance de **Minkowski** (plus générale) : Soit $q \in \mathbb{N}, q > 0$.

$$d_{Mink(q)}(\vec{x}, \vec{y}) = \sqrt[q]{\sum_{i=1}^{n} |x_i - y_i|^q}$$

- N.B. : $d_{Eucl}(\vec{x}, \vec{y}) = d_{Mink(2)}(\vec{x}, \vec{y})$
  $d_{Manh}(\vec{x}, \vec{y}) = d_{Mink(1)}(\vec{x}, \vec{y})$

Concept de cluster

• La méthode **Clustering** partitionne un ensemble $S \subseteq O$ en plusieurs clusters (groupes).

• Plusieurs caractérisations (propriétés) du concept **cluster** existent :
Centrisme, Séparatisme et **Atteignabilité**

Ces propriétés caractérisent les qualités des clusters :

• Centrisme caractérise le centre d'un cluster : chaque objet est plus proche du centre de son cluster que du centre d'un autre,

• Atteignabilité : un cluster contient les éléments qui doivent en faire partie : chaque objet appartient au même cluster que son voisin le plus proche,

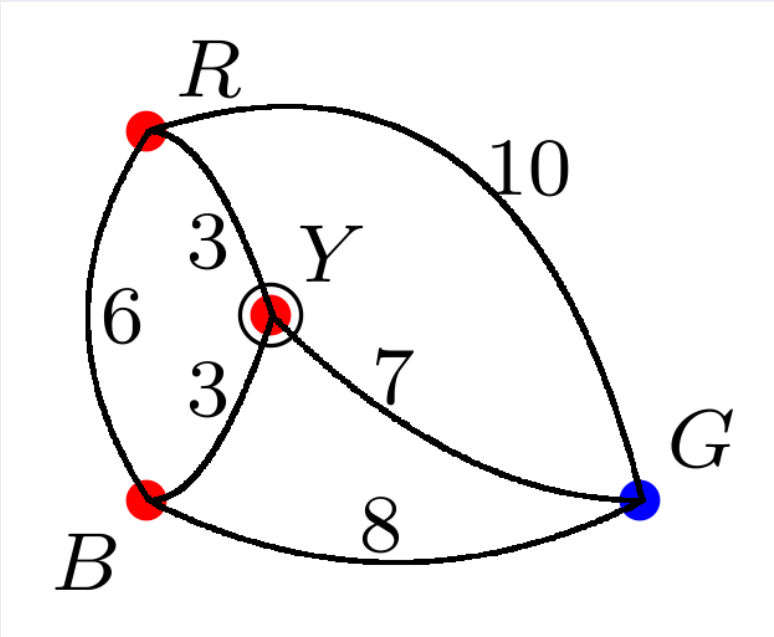
• Séparatisme : désigne deux clusters différents (dont les éléments sont séparés) : chaque objet est plus proche de tout objet de son cluster que de ceux d'un autre,

N.B. : les méthodes en particulier à base de *densité* (p. ex. DBScan, v. +loin) illustrent bien ces concepts.

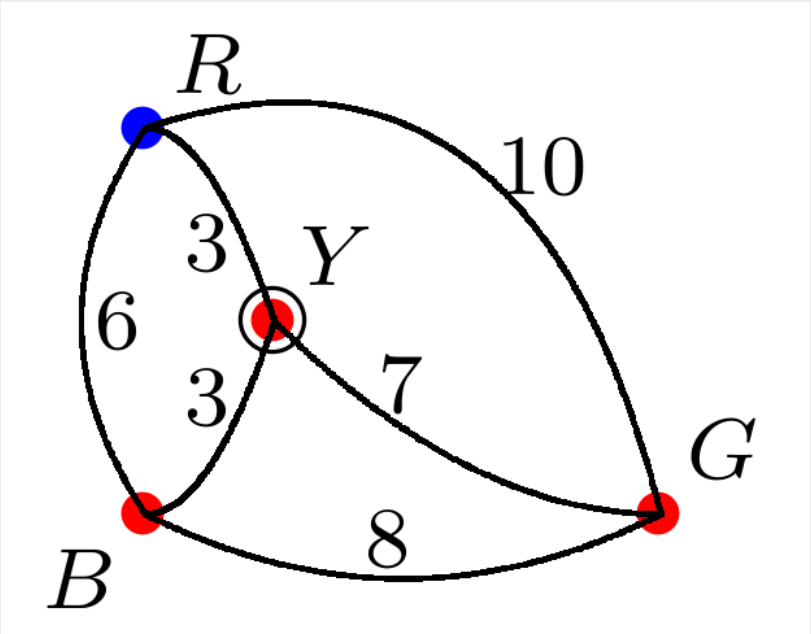
Concept de cluster (suite)

- **L'intérêt de ces 3 propriétés :**
  - Il y a plusieurs manières de partitionner :

➛ Que choisir (les 2 clusters "les rouges" et "le bleu" dans les 2 figures) ?



$\{R, B, Y\}$ et $\{G\}$ (meilleur clustering)



$\{G, B, Y\}$ et $\{R\}$ (moins bien ?), voir ../..

Addendum : : 3 priopriétes

① **Centrisme** : *liens entre un centre et les membres du cluster*

Soient l'ensemble d'instances O, $x \in O$ et deux clusters C, D différents :
- cluster C avec le centre c
- cluster D avec le centre d.

Si $x \in C$, alors $d(x, c) \leq d(x, d)$.

➙ Idéalement, x appartient au cluster qui lui est **le plus proche**.

> Le Centrisme souligne le rôle du centre c d'un cluster :
> ➙ $x \in C$ doit être plus proche du centre c que d'un autre centre.

- Connaissant les centres, on peut connaître les clusters.

② Le principe de **Séparabilité** (des clusters) est du même ordre :

- Un clustering bien formé sépare effectivement ce qui doit l'être.
- Donne le **diamètre** (*seuil* ▲) des clusters séparés (voir +loin).

➙ Tout objet à l'intérieur du diamètre d'un cluster doit en faire partie.

Addendum : : 3 priopriétes (suite)

③ **Atteignabilité** : concerne la *distance entre les éléments du cluster*

➛ est la distance qui sépare le point le plus éloigné du reste du cluster.

- Soit un ensemble d'objets $S \subseteq O$ et C un cluster qui contient l'objet $x \in S$.
  - Idée raisonnable : tout objet $y \in S$ *"proche de x"* appartient aussi à C :
    - ➛ Si $x \in C, y \in S$ et $d(x, y) \leq \blacklozenge$, alors $y \in C$. $\blacklozenge$ *est un seuil*
  - Que devrait être le seuil d'*Atteignabilité* ($\blacklozenge$) ? :
    - ➛ Elle peut être p. ex. $\blacklozenge = max\{d(x, y)|x, y \in C, y \neq x\}$ *(peu efficace, v. + loin)*
    - ➛ Sauf cas triviaux, ce seuil est calculé *a posteriori*

- Les conséquences d'atteignabilité $\blacklozenge$ (voir son calcul p. svte) :
  1. Si $d(x, y) \leq \blacklozenge$ alors x et y appartiennent au même cluster
  2. Si x et y appartiennent au même cluster et $d(y, z) \leq \blacklozenge$
     Alors x et z appartiennent au même cluster (transitive)
  3. Deux points appartiennent au même cluster seulement si les 2 règles ci-dessus l'imposent.

N.B. : Tout clustering qui satisfait *"séparatisme"* satisfait forcément *"atteignabilité"*.

➛ Voir les méthodes OPTICS, DBSCAN, COBWEB pour illustrer ces mesures.

Addendum : : 3 priopriétes (suite)

Calcul des seuils ♦ et ▲

- Seuil d'*Atteignabilité* (♦) d'un cluster C :

  Pour un ensemble d'objets $S \subseteq O$ et C un cluster qui contient l'objet $x \in S$,

  Tout objet $y \in S$ *"proche de x"* appartient aussi à C :

  ➛ Si $x \in C, y \in S$ et $d(x, y) \leq$ ♦, alors $y \in C$.

> Si pour $x \in C$, $dist\_au\_plus\_proche(x) = min\{d(x, v) | v \in C, v \neq x\}$.
> Alors la distance du plus éloigné dans le cluster C donne le seuil d'*Atteignabilité* :
> ♦ $= max\{dist\_au\_plus\_proche(w) | w \in C\}$.

♦ donne la distance du membre le plus éloigné du reste du cluster C.

- *Séparabilité* (▲ = le diamètre du cluster) : $\boxed{▲ = max\{d(v, w) | v, w \in C\}}$,

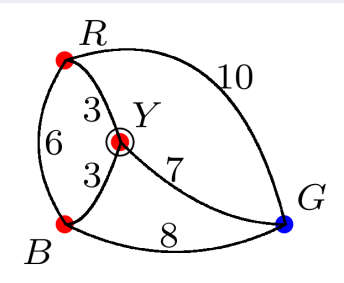
- Le *Centrisme* est au coeur des méthodes de clustring (dans la suite).

Rappel : Tout clustering qui satisfait *"séparatisme"* satisfait *"atteignabilité"*.

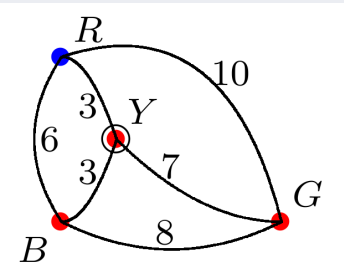
Addendum : : 3 priopriétes (suite)

Exemples d'illustration : Centrisme, séparabilité, Atteignabilité :

- Il y a plusieurs manières de partitionner {R, B, G, Y} en deux clusters.
- Par exemple, les clusters *rouge* et *bleu* (hyp : il faut 2 clusters) :



▲ = 6 , "Séparabilité" (diamètre) et "Centrisme".
➛ Centrisme : {R,B} plus proches de Y que de G
➛ Séparatisme : ▲ = max de distance de tout couple {R,B,Y} (rouges).
➛ les éléments de {R,Y,B} sont "proches" (vs. G)
→ Les clusters rouge et bleu bien séparés.



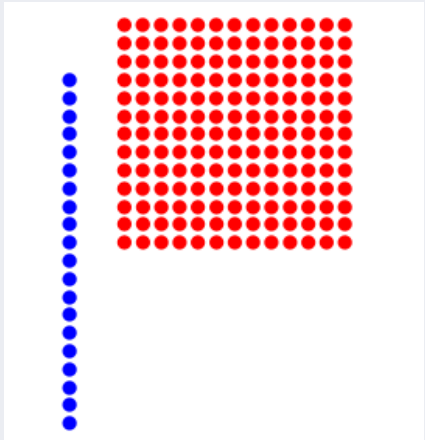
♦ = 7 , "Centrisme" mais pas "Atteignabilité".
➛ Centrisme : R est aussi loin de Y que le couple {G,B} de Y
➛ Atteignabilité non respectée : G est + éloigné dans {B,Y,G} que R.
→ Pour ♦ = 7, R est plus proche que G.

➛ **DONC** : le clustering du haut est meilleur que celui du bas.
➛ Importance des mesures d'évaluation (v. plus loin)

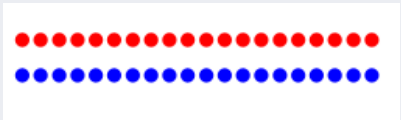
Addendum : : 3 priopriétes (suite)

Autres exemples :

• Satisfait l'atteignabilité mais pas le séparatisme ni le centrisme



• Satisfait le centrisme et l'atteignabilité mais pas le séparatisme.



• Satisfait le centrisme mais pas l'atteignabilité.

Addendum : : 3 priopriétes (suite)

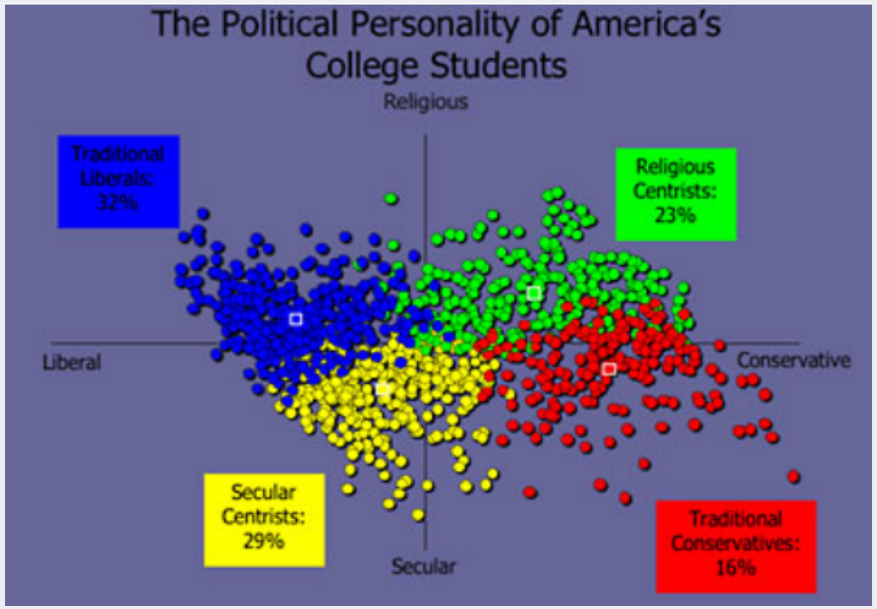
☞ Le Centrisme, la Séparabilité et l'Atteignabilité, c'est comme la "liberté" :

➛ *La mienne s'arrête où commence la tienne !*

La distance n'est pas forcément une distance Linéaire / Euclidienne "2d" !

➛ Dans la fig., plusieurs points pourraient être dans un autre cluster (si distance 2d)

➛ Dans les grandes dimensions, les distances tendent à se confondre



Méthodes de Clustering

Méthodes de Clustering (développées ci-après) :

- Méthodes de partitionnement
  - k-Means & k-Means Dichotomique
  - k-Medoids
  - Comparaison k-Means et k-Medoids
  - Expectation-Maximization

- Méthodes Hiérarchiques
  - Méthodes Agglomérative
    - ➛ Dendrogrammes

  - Méthodes Divisives
    - ➛ Cobweb

- Méthodes à bas de densité
  - ➛ DBSCAN

- Méthodes à bas de grilles
  - ➛ STING

- Grande échelles (Large Data)
  - ➛ CLARA et CLARANS
  - ➛ BIRCH, CURE, CLIQUE

Méthodes de Clustering (suite)

- Remarques sur WEKA et le Clustering.

  La plupart des méthodes sont implantées dont :

  - *EM* (Expectation Maximisation),
  - *SimpleKmean*
  - *Farthest-First.*
  - *CobWeb* : ascendant à base de la mesure *Category utility.*
    - ➛ A noter (sous Weka) :
      "Classes to Cluster Evaluation" tient compte des classes.
  - *DBScan* : clutering à base de densité
  - Autres variantes de *K-Means* : différents algorithmes
  - *etc.*

K-means

Principe général : clustering itératif des instances

- On souhaite partitionner un ensemble S en $k \geq 2$ clusters.
- Soient $C_1, C_2, ..., C_k$ des clusters avec les centres $c_1, c_2, ..., c_k$ respectivement.
- On définit la **dispersion intra-cluster** par $\sum_{i=1}^{k} \sum_{x \in C_k} d(c_k, x)$
- Le but est de trouver un clustering avec une dispersion intra-cluster minimale.

Exemple de clustering des couleurs (k=nb. couleurs) ➙

Algorithme de K-means

But : partitionner un ensemble S en k clusters.

Algorithme K-Means basique (à base de *Centroïdes*)

Données : D l'ensemble d'apprentissage
Résultats : Un ensemble de k clusters

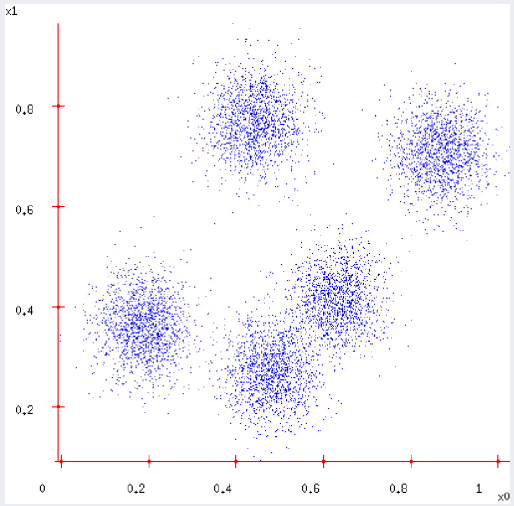
Choisir les Centroïdes (moyennes) $m_1, m_2, ..., m_k$ *(via la distance Euclidienne)*
Répéter
  Attribuer tout objet de S au Centroïde le plus proche
  Soient $C_1, C_2, ..., C_k$ *les ensembles d'objets attribués respectivement à* $m_1, m_2, ..., m_k$
  Ajuster les Centroïde par :
    $m1 \leftarrow$ *le Centroïde de* C_1
    $m2 \leftarrow$ *le Centroïde de* C_2
    ...
    $mk \leftarrow$ *le Centroïde de* C_k
Jusqu'à la stabilisation des clusters (que m_i *ne changent plus)*

- La qualité des résultats sera évaluée par $SSE = \sum_{i=1}^{k} \sum_{x \in C_i} [dist(m_i, x)]^2$ à minimiser

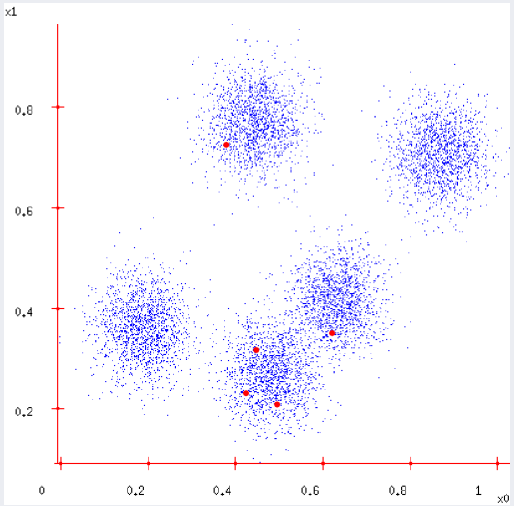
- Exemple de distance **Euclidienne** $dist =$

$$\sqrt{\left(a_1^{(1)} - a_1^{(2)}\right)^2 + \left(a_2^{(1)} - a_2^{(2)}\right)^2 + \ldots + \left(a_k^{(1)} - a_k^{(2)}\right)^2}$$

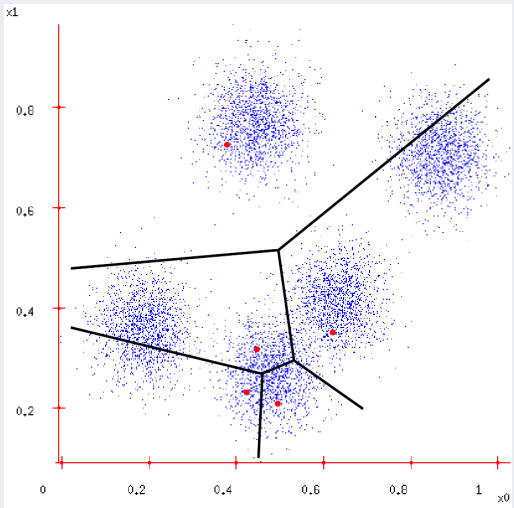
Illustration de K-means



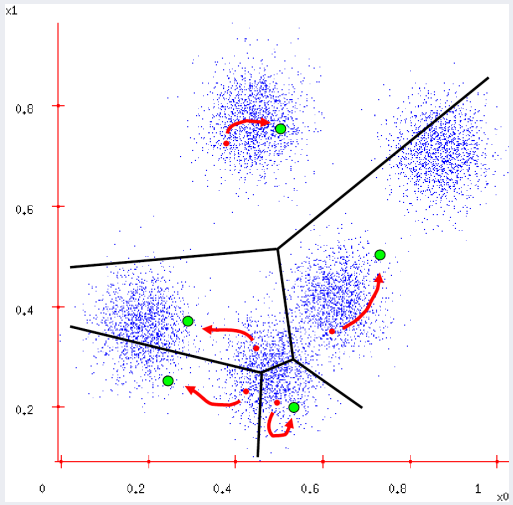
Points d'origine



Initialisation



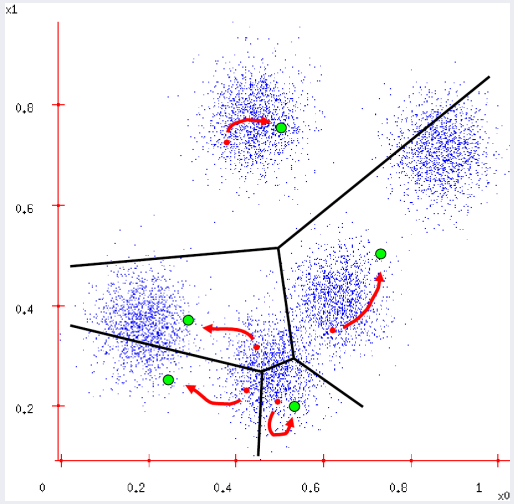
Étape 1



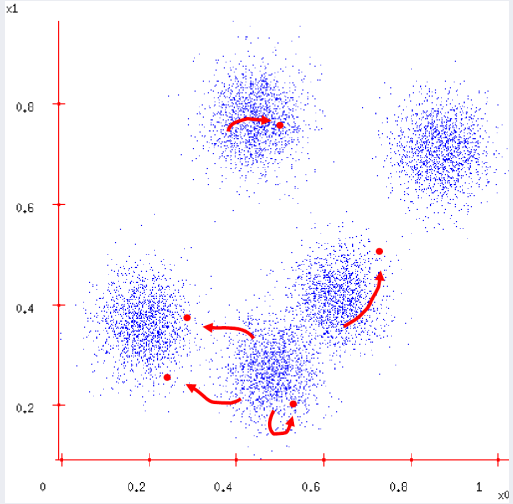
Étape 2

Illustration de K-means (suite)

Illustration de K-means (suite) :



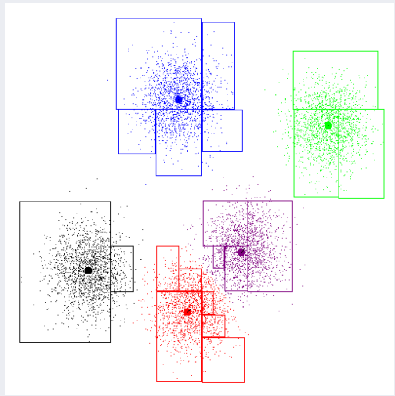
Etape 3



Etape 4 .....

Démarche :

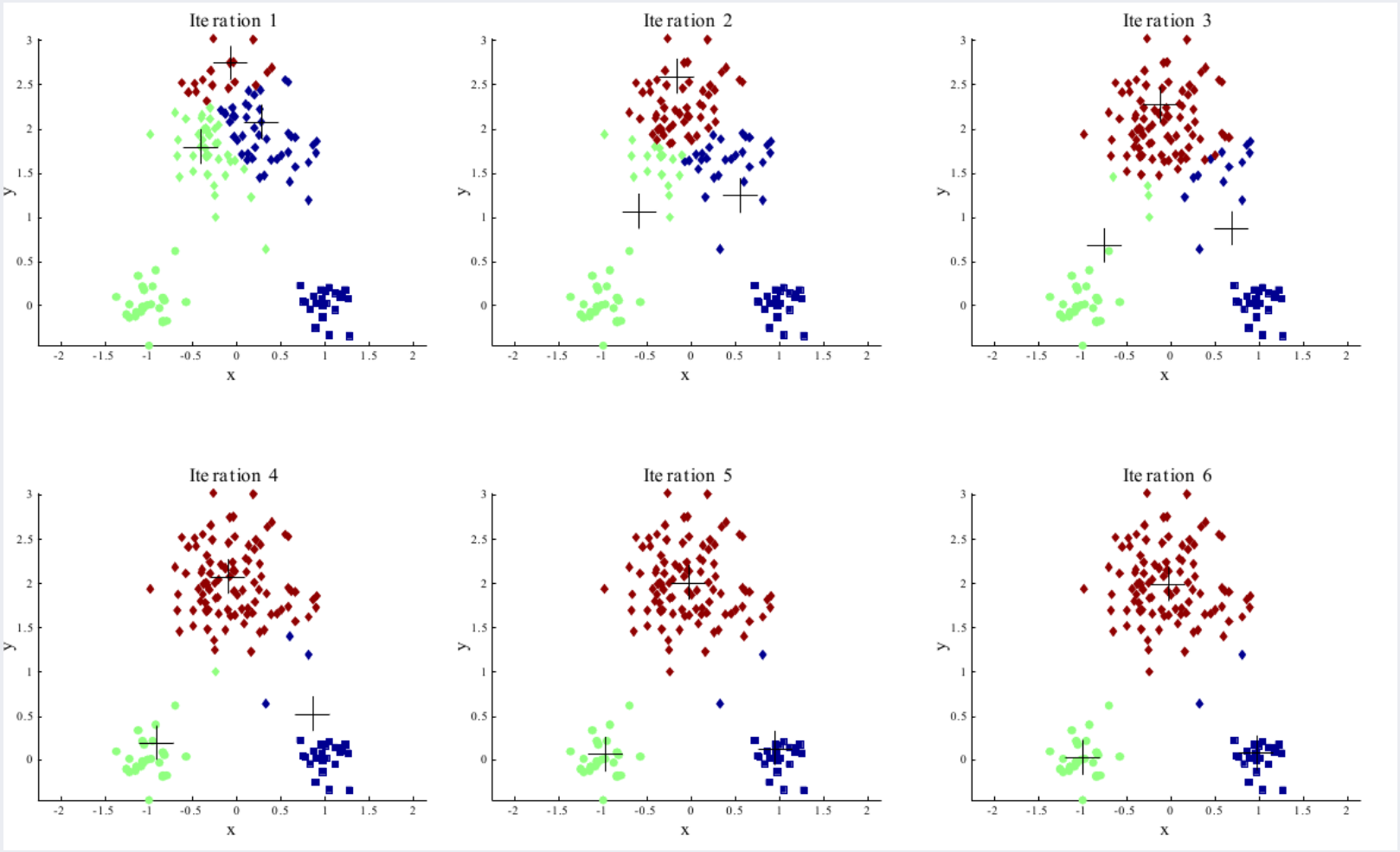
- On fixe $k = 5$ et les centroïdes aléatoires initiaux ;
- Chaque instance "trouve" le Centre le plus "proche" ;
- Chaque Centre "trouve" le Centroïde des points qui lui sont devenus "proches"
- Et s'y déplace !
- .... Jusqu'à ne plus se déplacer !



Etape finale

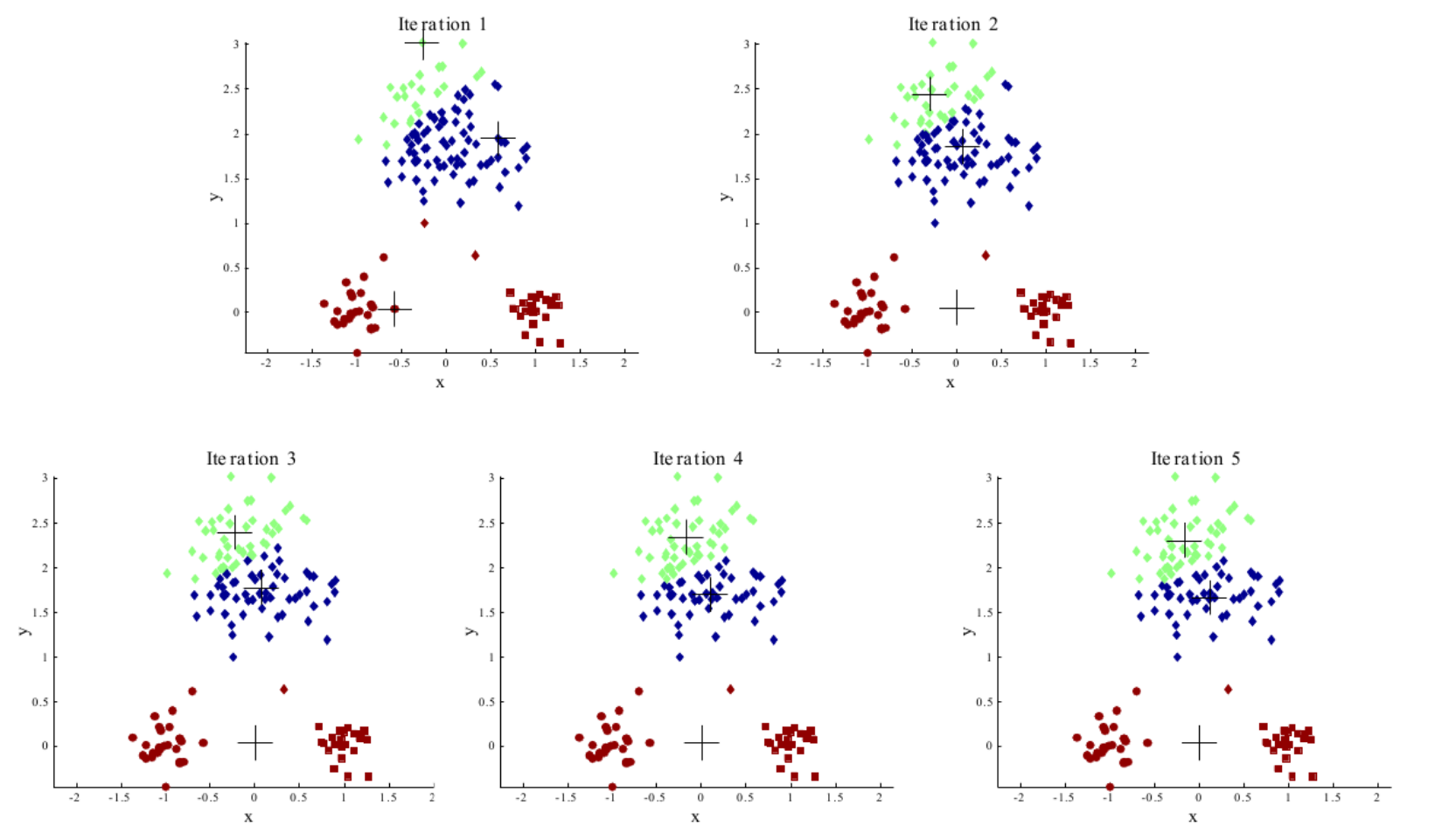
K-means : importance des choix initiaux

Le choix des centres initiaux est important : ici, bon choix des 3 médoïdes



K-means : importance des choix initiaux (suite)

Exemple (suite) : ici, mauvais choix initiaux des 3 médoïdes



K-means : Recherche des voisins

- On peut prouver que pour un cluster (par l'algorithme *K-means*) :

  La sommes au carré des distances aux centroïdes (calculés) est minimale

☞ Mais c'est une propriété **locale** : pas de garanti d'optimum global.

- Une techniques habituelle pour atténuer ce problème :
  - exécuter l'algorithme *K-means* avec des choix différents pour k
  - puis choisir les meilleures distances globales

Amélioration / accélération du calcul des distances :

- Sachant qu'on doit considérer TOUTES les instances à chaque itération
  ➛ (cf. classer individuellement les instances de test, cf. dans IBL)

  Trouver le centroïde le plus proche n'est pas différent de PPV dans IBL
  ➛ On peut utiliser les mêmes techniques (KD / Ball-trees) ⋯ ⇝

K-means : Recherche des voisins (suite)

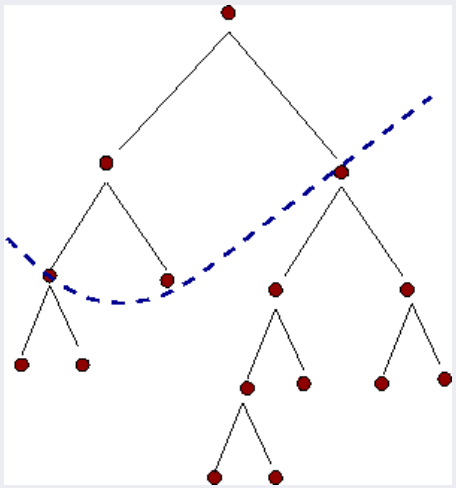
Optimisation de la recherche des Centroïdes dans k-means

Idée : une hiérarchie locale permet d'éviter le parcours de toute la BD.

➛ *Les instances sous le trait en pointillé n'ont pas besoin d'être comparées*

Démarche (d'amélioration des calculs de distances / médoïdes) :

- Construire un KD-Tree (ou Ball-Tree) pour toutes les instances (arbre inchangé jusqu'à la fin du traitement)
- Pour trouver les points les plus proches des Centroïdes (dans toute la BD) :
  - descendre l'arbre depuis la racine et rouver le ($\simeq$) Centroïde
  - les instances au niveau des feuilles tombent dans une région dont le centre est à un niveau supérieur de la même branche



- Pour calculer les Centroïdes, on maintient le cumul des distances pour les instances traitées dans un cluster + le nombre d'instances

  ➛ A la fin, on divise le cumul par le nombre pour trouver le centroïde

K-means : Recherche des voisins (suite)

K-means et les caractéristiques des Clusters

- K-means garantit "centrisme" mais pas "atteignabilité" .

$b\bullet$ $\qquad$ $d\bullet$

$a\bullet$ $\qquad$ $c\bullet$

Rappel "atteignabilité" (un ensemble de points S, un cluster C) :
pour l'instance $x \in C$ ($C \subseteq S$), $\forall y \in S$ "proche" de $x \Rightarrow (y \in C)$

- Le résultat si l'on démarre avec les centroids a, b ➛
- Une meilleure solution serait avec $\{\{a, b\}, \{c, d\}\}$.
- Si plusieurs *K-means* lancés, on utilise les critères d'évaluation de K-means :
  ➛ *max* des distances inter-cluster et *min* des distances intra-cluster ../..

Distorsion (K-means)

Méthode de base du calcul de l'erreur de Clustering

- Soient les instances $\overrightarrow{x_i}, i = 1..n$ dans un espace $\mathbb{R}^d$ (d attributs) :
- $Dom(k)$: le domaine constitué des instances du cluster k
- K clusters, et une fonction d'encodage $encoder : \mathbb{R}^d \rightarrow [1..K]$

  ➞ *encoder* affecte le cluster le plus proche d'une instance, pas forcément SON cluster calculé !
- Une fonction de décodage $decoder : [1..K] \rightarrow \mathbb{R}^d$ (cluster vers l'espace $\mathbb{R}^d$)

☞ *encoder* idéal : $decoder(encoder(x_i)) = x_i$ où $x_i \in Dom(encoder(x_i))$

- On définit la **Distorsion** (comparer à SSE) :

$$Distorsion = \sum_{i=1}^{n} [x_i - decoder(encoder(x_i))]^2$$

☞ Si $\forall x_i \in C_j$ dont le centre est m_j, la moyenne $decoder[x_i] = m_j$,
Alors *Distorsion* ≡ l'erreur SSE.

Propriétés des Centroïdes

(1) Quelque soit le cluster final de x_i, $encoder(x_i)$ associe x_i au Centroïde m_j le plus proche $(\exists j \in [1..K], j = encoder(x_i))$:

$$m_{encoder(x_i)} = \underset{m_j \in \{m_1, m_2, \ldots, m_k\}}{\operatorname{argmin}} [x_i - m_j]^2$$

m_j tel que la distorsion soit **min.**

➞ m_j est le Centroïde du cluster associé à x_i tel que SSE soit minimale.

On associe à x_i un centroïde (proche) qui n'est pas forcément celui de son cluster calculé (p. ex. par *K-means*) !

(2) Il en découle :

La dérivée partielle de la *Distorsion* par rapport à chaque Centroïde doit être nulle.

Rappel : $Distorsion = \sum_{i=1}^{n} [x_i - m_{encoder(x_i)}]^2$

../..

Propriétés des Centroïdes (suite)

Que faut-il pour minimiser la Distorsion .

$$Distorsion = \sum_{i=1}^{n} [x_i - m_{encoder(x_i)}]^2 = \sum_{j=1}^{k} \sum_{x_i \in dom(m_j)} [x_i - m_j]^2$$

➙ $dom(m_j)$ représente les instance du cluster C_j , $|dom|$ = la taille du domaine dom

$$\frac{\partial Distorsion}{\partial m_j} = \frac{\partial}{\partial m_j} \sum_{x_i \in dom(m_j)} [x_i - m_j]^2$$
$$= -2 \sum_{x_i \in dom(m_j)} [x_i - m_j] = 0$$ pour être un minimum

D'où $$\sum_{x_i \in dom(m_j)} [x_i - m_j] = \sum_{x_i \in dom(m_j)} (x_i) - |dom(m_j)|.m_j = 0$$

➙ $$\boxed{m_j = \frac{1}{|dom(m_j)|} \sum_{x_i \in dom(m_j)} x_i}$$ la *moyenne*

➙ Pour minimiser la Distorsion, tout Centroïde C_j doit être au centre (moyenne) de son domaine $dom(m_j)$

Principes de K-means

Rappel des 2 propriétés des Centroïdes :

1. Quelque soit le cluster final de x_i, $encoder(x_i)$ associe x_i au Centroïde le plus proche
2. $m_j = \frac{1}{|dom(m_j)|} \sum_{x_i \in dom(m_j)} x_i$

• Ces deux propriétés donnent le **principe même de K-Means** :
1 ➛ Attacher $x_i, i = 1..n$ au Centroïde le plus proche $m_j, j = 1..k$
2 ➛ Recalculer le nouveau m_j au centre du $dom(m_j)$

• On peut prouver que cette stratégie termine dans un état (stable) où plus aucune des deux propriétés ne change l'état du système.

Indication : nombre d'instances fini
- ➛ Nombre de configuration (optima) fini.
- ➛ Au pire, on examinera toutes les configurations.

Remarque : *K-Means* est une (variante de la) méthode *EM* simple.

Evaluation de K-means

- La mesure la plus commune : **la somme des erreurs au carré** (SSE) :

$$SSE = \sum_{i=1}^{k} \sum_{x \in C_i} [dist(m_i, x)]^2$$

x : une instance dans le cluster C_i, m_i : le point représentatif du claster C_i
➙ On a montré que m_i correspond au centre (moyenne) du cluster C_i

- Étant donné deux *clusterings*, on peut choisir celui avec SSE minimale.

- Une technique utilisée pour réduire SSE est d'augmenter k (le nombre de clusters)

Néanmoins : un "bon" clustering avec un plus petit k peut avoir une moindre SSE qu'un mauvais clustering avec un k plus élevé.

Le hasard des choix

• S'il y a k vrais clusters (de qualité), alors la chance de choisir le centroïde de chaque (bon) cluster est très faible.

◦ Cette chance est minime pour un grand (vrai) k
◦ Si les clusters sont de la même taille t, alors :

$$p = \frac{nbr\ de\ manières\ de\ choisir\ un\ centroïde\ pour\ chaque\ cluster}{nbr\ de\ manières\ de\ choisir\ k\ centroïdes} = \frac{k!t^k}{(k.t)^k} = \frac{k!}{k^k}$$

Exemple : pour $k = 10$,

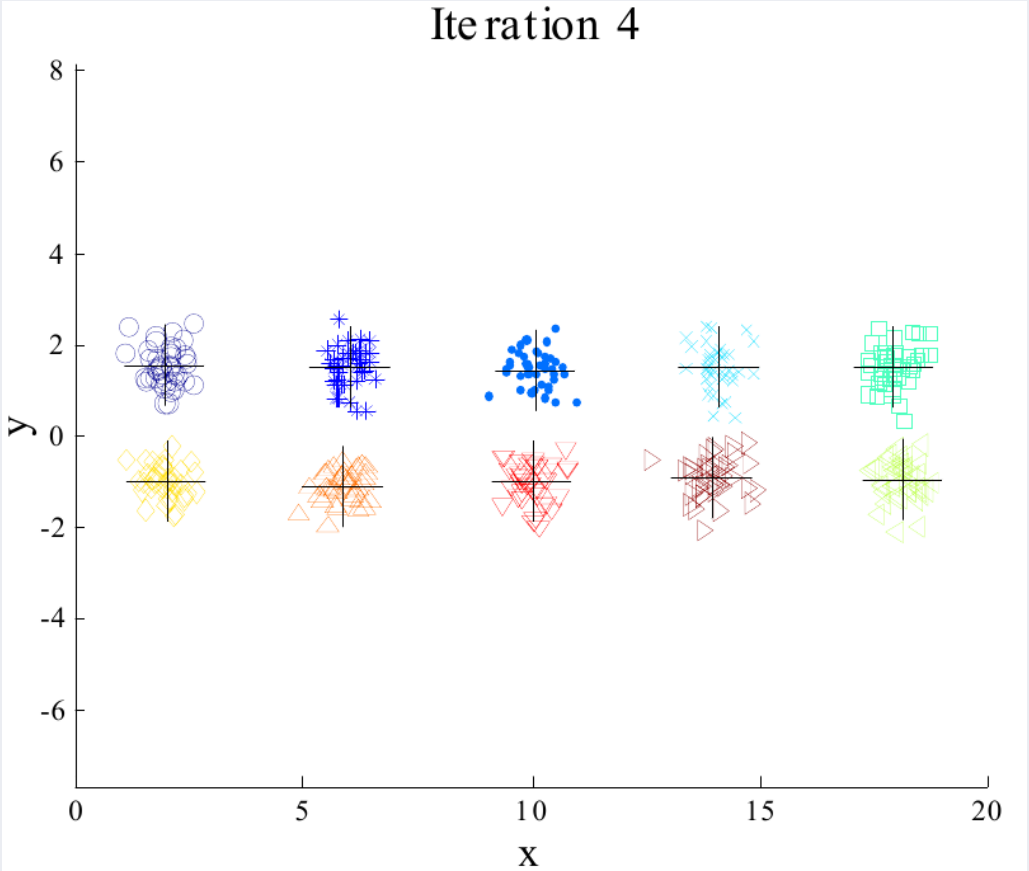
➙ la probabilité de bien choisir les centroïdes $= \frac{10!}{10^{10}} = 0.00036$

• Parfois, les centroïdes initiaux s'auto réajustent "bien", parfois non !

• Voyons un exemple avec 5 paires de clusters $\cdots \rightsquigarrow$

Le hasard des choix (suite)

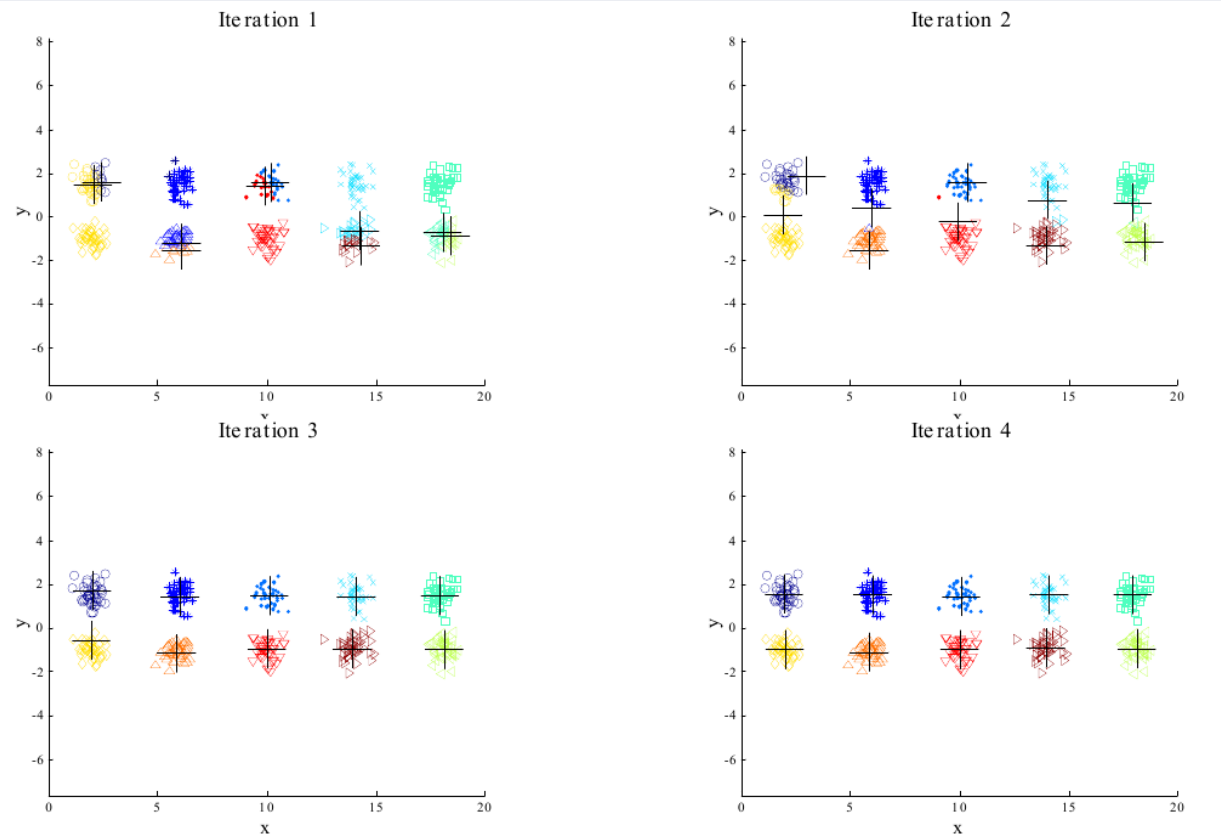
Exemple-1 (résultat d'un clustering idéal avec 5 pairs de clusters) :



Cas idéal obtenu avec les 10 clusters bien formés.

Le hasard des choix (suite)

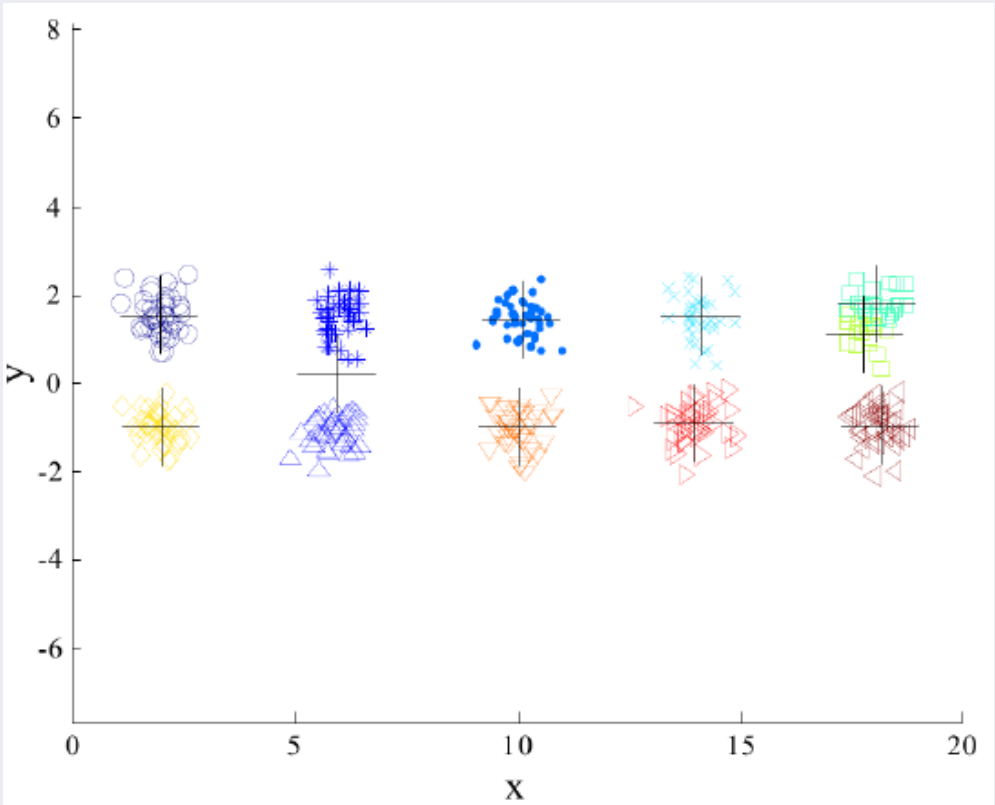
Détails des itérations de la même BD. :



Détails de l'Ex-1 : ici, *K-means* à placé initialement chaque couple de centroïdes dans une seule de chacune des paires de grappes.

Le hasard des choix (suite)

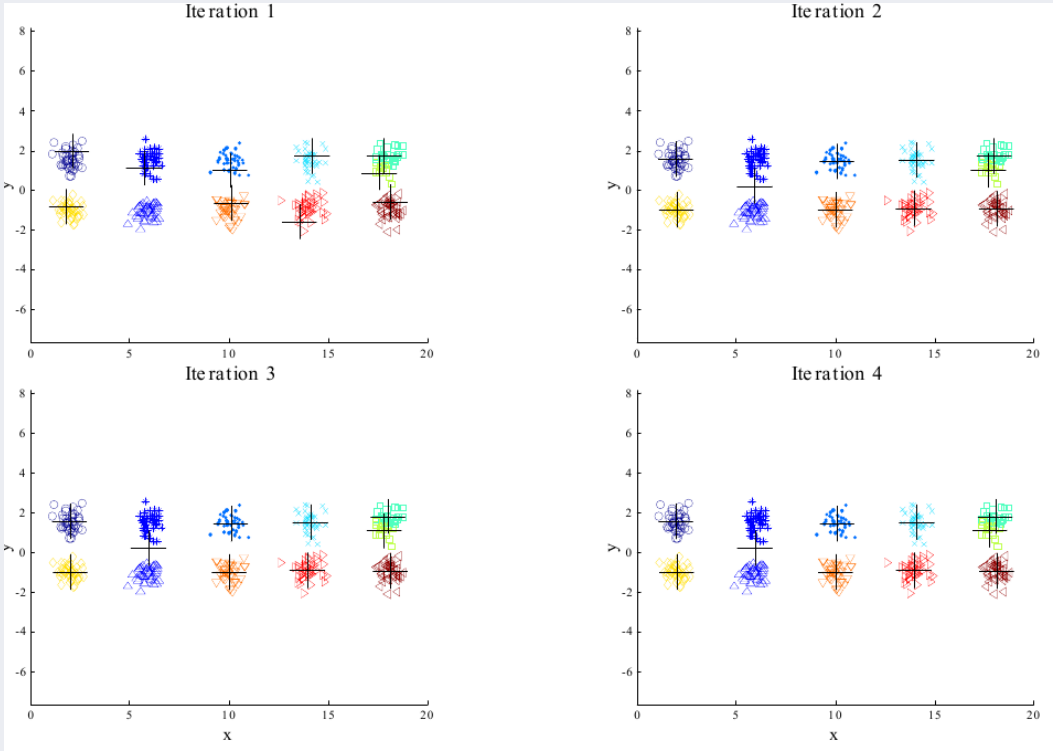
Exemple-2 : résultats obtenu avec 10 clusters "mal" formés



Résultats obtenu avec 10 clusters "mal" formés (voir détails ../..).

Le hasard des choix (suite)

Détails des itérations de l'exemple-2



Détails Ex-2 : certaines paires de grappes avec 3 centroïdes initiaux, d'autres avec un.

Solutions au problème des centroïdes initiaux de K-means

- Plusieurs exécutions (de K-means) avec des choix différents
  ➙ Peut aider à choisir un meilleur clustering
- Choisir plus que k centroïdes ($k' > k$) initiaux puis choisir k parmi eux
- **Échantillonner** une partie des données et utiliser un clustering Hiérarchique (v. plus loin) pour choisir les centroïdes initiaux
  ➛ Bisecting K-means (dichotomique) : Une alternative
- Sélectionner les centroïdes initiaux les plus dispersées *(cf. Farthest First)*
- Et enfin ... Post traiter (voir plus loin)

K-means : problème des clusters vides

- Le K-means basique peut produire des clusters vides
- On tente d'insérer au moins une instance dans les clusters vides :
  - ➛ Choisir le point qui contribue le plus au SSE et en faire un singleton (au niveau des instances)
  - ➛ Choisir un point d'un cluster ayant le maximum de SSE (au niveau cluster)
  - ➛ ...
- Si plusieurs clusters vides, on peut répéter ces opérations

K-means : rectification incrémentale des centroïdes

Rappel de l'algorithme basique (K = nbr de clusters fixé) :

1. Tirage aléatoire de K centres initiaux
2. Répéter jusqu'à plus de changement
   Affecter chacune des instances restantes au cluster le plus proche
   Recalculer la moyenne de chaque cluster

• Dans cet algorithme, les centroïdes sont mis à jour à la fin d'une itération (après l'affectation de tous les points d'un cluster donné)

Une approche alternative :

• Mise-à-jour du centroïde après l'ajout de chaque point :
  ◦ Chaque point ajouté peut modifier de zéro à deux centroïdes
    ➛ Ce point ne bouge pas ou quitte un cluster pour aller vers un autre
  ◦ Méthode plus coûteuse
  ◦ Évite la création de clusters vides
  ◦ Pose un problème : introduit une dépendance à l'ordre des ajouts de points

➛ On utilise des pondérations pour atténuer son impact.

Pre-traitement et Post-traitement des clusters

- **Pré-traitement des données**
  - Normaliser les données
  - Éliminer les outliers (et les anomalies)
- **Post-traitement**
  - Éliminer les petits clusters (qui peuvent représenter des outliers)
    ➙ les fusionner
  - Découper les clusters avec une SSE élevée (dits *"loose clusters"*)
  - Fusionner les clusters "proches" avec une SSE basse.
- Ces étapes peuvent être également appliquées pendant le clustering
  ➙ (cf. algorithme ISODATA)

Inadaptation de K-means à certaines données

- K-means est mal adapté quand les (bons) clusters sont différents en :
  - Tailles
  - Densités
  - Formes des nuages des points non réguliers

- K-means a également des insuffisances quand les données contiennent des outliers (*ex calibur*).

- Illustrations ../..

Inadaptation de K-means à certaines données (suite)

K-means : problème de la taille des clusters

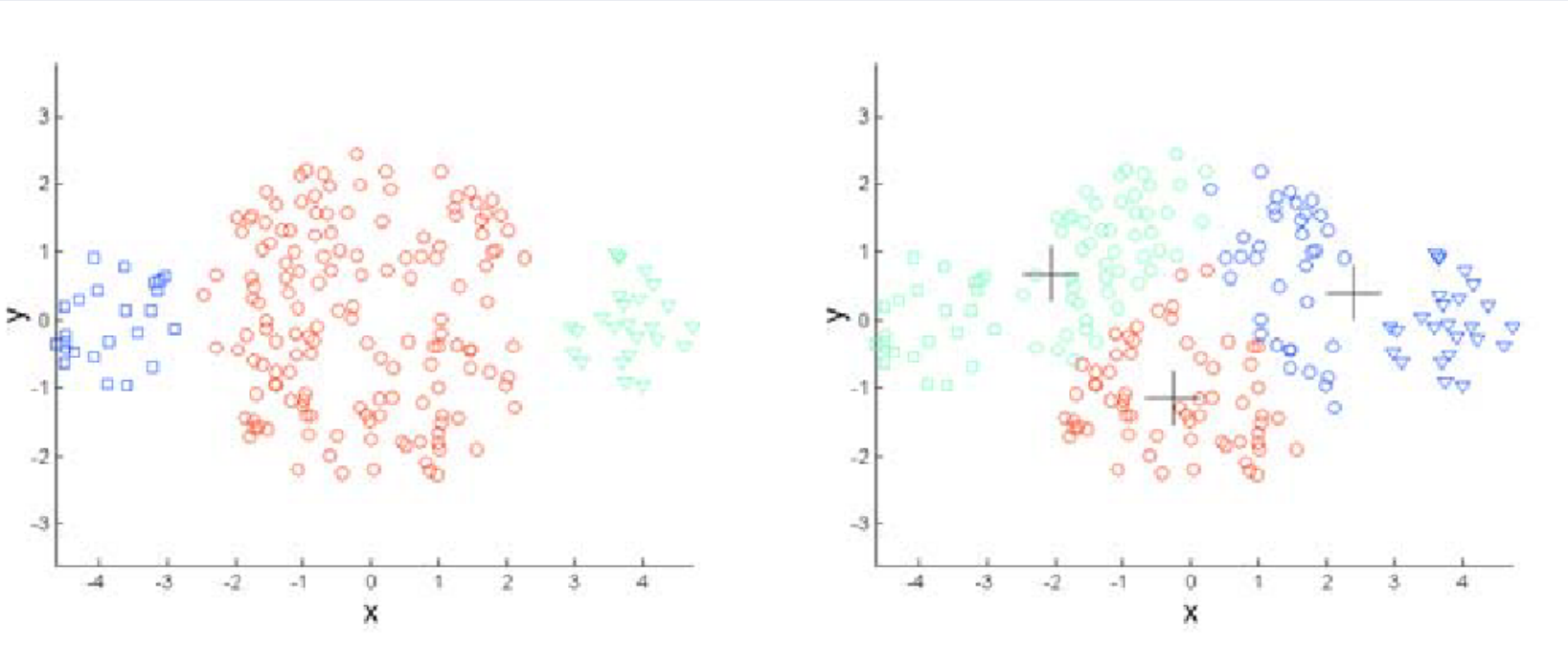


FIGURE 1: K-means : problème de la taille des clusters

Points d'origine (bcp de rouges) K-means (3 Clusters)

(observer les points rouges d'origine)

Inadaptation de K-means à certaines données (suite)

K-means : problème de la densité des clusters

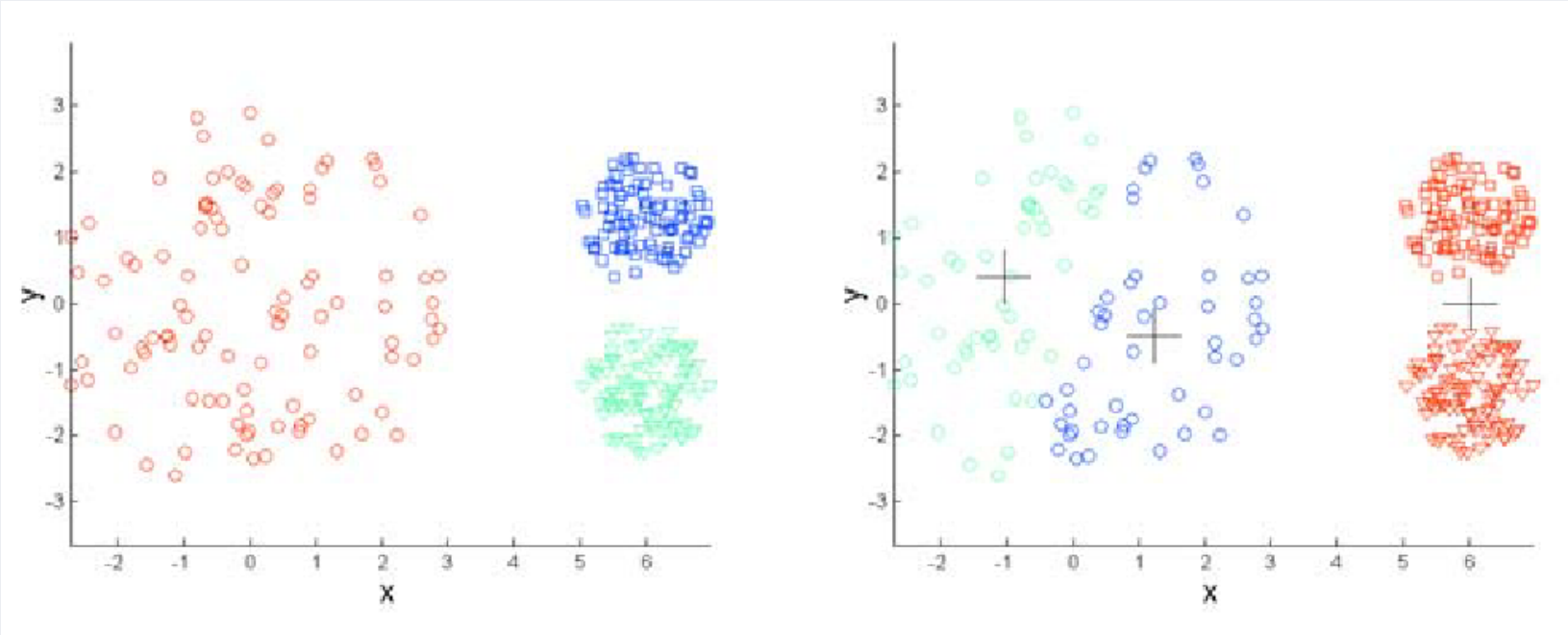


FIGURE 2: K-means : problème de la densité des clusters

Points d'origine (bleu dense) K-means (3 Clusters)

- *K-means* ignore les densités locales

Inadaptation de K-means à certaines données (suite)

K-means : problème de la forme des points

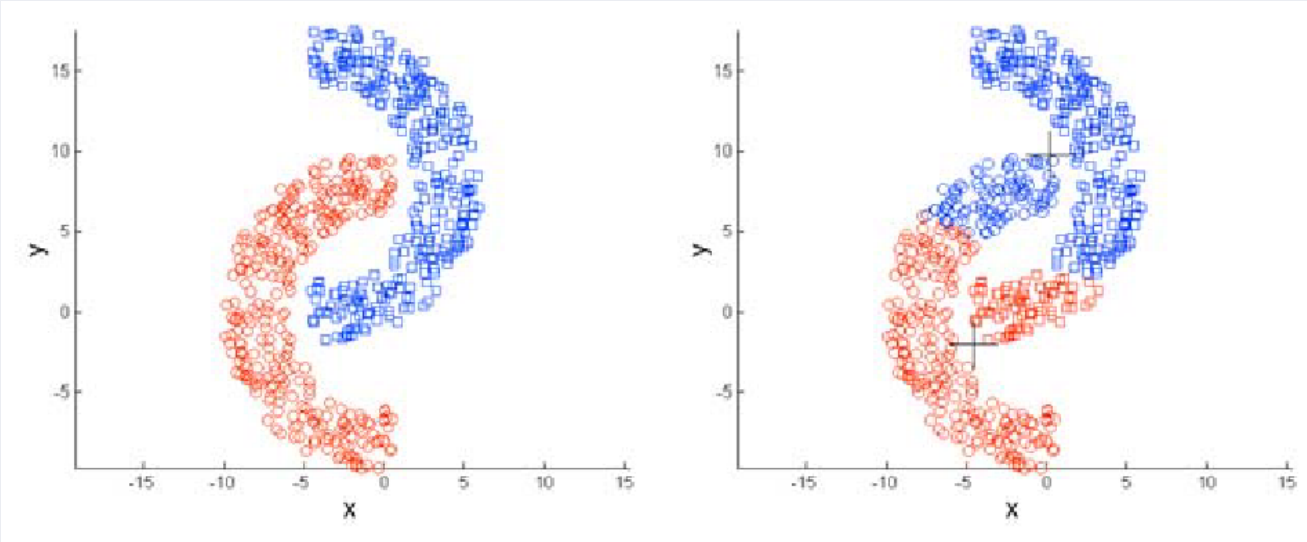


FIGURE 3: K-means : problème de la forme des points

Points d'origine K-means (2 Clusters)

- Une solution à ces problèmes est d'utiliser plusieurs (beaucoup) de clusters.

  ➛ Trouver des clusters "partiels", mais il faut les fusionner ensuite.

- Voir également des méthodes de Clustering différentes (cf. DB-Scan)

Variante : K-means Dichotomique

- Une variante de K-means qui produit un clustering par partitionnement (ou Hiérarchique)

- Le principe : partitionnement *Dichotomique* des clusters jusqu'à l'obtention de k clusters.

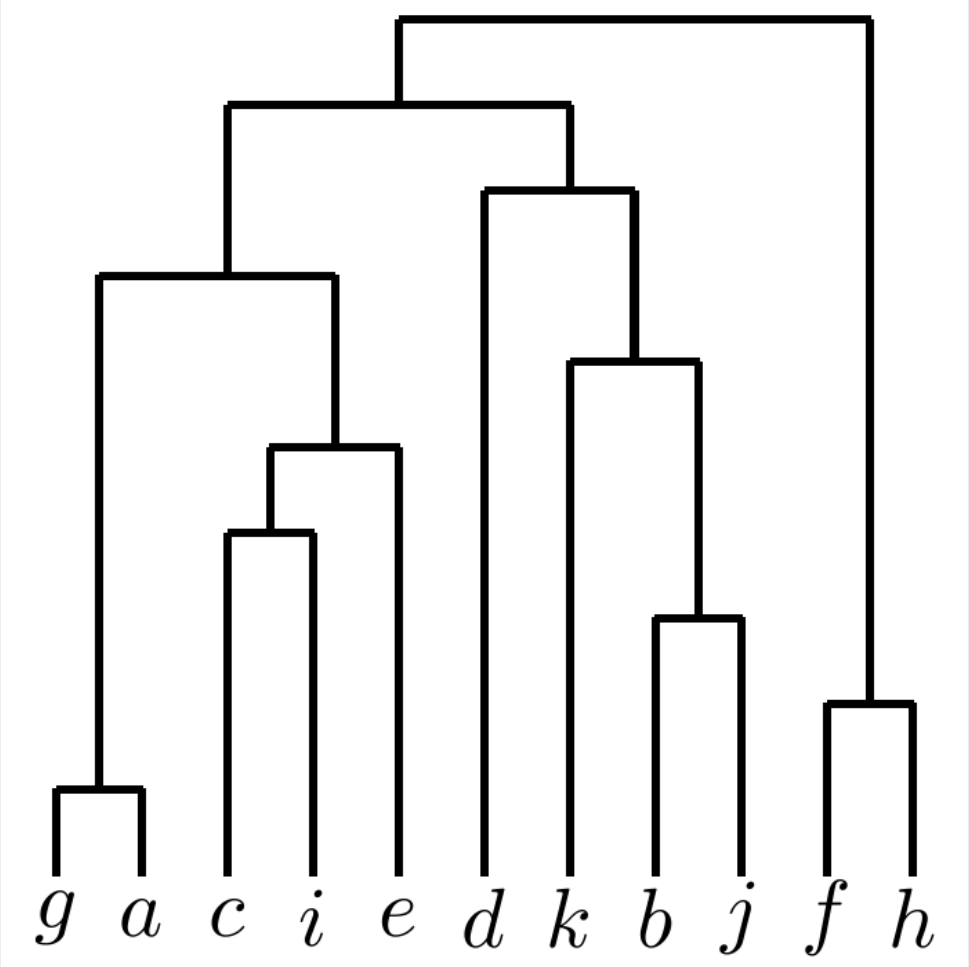
Algorithme K-Means *Dichotomique* (*Bisecting K-means*) :

Donnees : S l'ensemble d'apprentissage ; M : nombre d'itérations
Résultat : Un ensemble de k clusters
Initialiser la liste LC des clusters par un seul cluster contenant tous les points
Répéter :
    Choisir un cluster C de la liste des clusters
    Pour i de 1 à M (M : $nbr.d'itérations$ *fixé en entrée*)
        Découper C en deux clusters en utilisant K-means basique (et minimiser SSE)
        → Le cluster C produit 2 meilleurs clusters (après M tentatives)
    Ajouter à LC les deux clusters de la bisection (avec un minimum de SSE)
Jusqu'à avoir k clusters

LC contient k clusters

➙ Si $M = 1$, *Dichotomique* basique.

Exemple Dichotomique



Variante : Clustering K-médoïdes

- Rappel : les médoïdes sont des instances de la BD (vs. Centroïdes = moyennes).
- K-médoïdes Clustering est une variante de clustering itératif (PAM) :

Algo. K-médoïdes itératif (ou *PAM : Partitioning Around Medoids*)

Données : S l'ensemble d'apprentissage
Résultat : Un ensemble de k clusters

Choisir les médoïdes aléatoires $m_1, m_2, ..., m_k \in S$

Répéter
- Chercher $m_j \in \{m_1, m_2, ..., m_k\}$ et un point aléatoire $p \in S - \{m_1, m_2, ..., m_k\}$
  tels que remplacer m_j par p améliore le clustering (e.g. l'erreur SSE)
- Remplacer m_j par p (remplacement si $cout(m_j, p) < 0$, c.à.d. amélioration)

Jusqu'à la Stabilisation des clusters

On arrête si aucun point p n'améliore l'erreur.

- Le m_j remplacé par un autre point p si ce remplacement est "meilleur".
- Algorithme plus couteux
- Résultats meilleurs de *K-means*

Variante : Clustering K-médoïdes (suite)

Différentes stratégies de K-médoïdes itératif

K-medoïde clustering : la stratégie **Farthest First**

- Pour le choix initial des m_i :
  Pour un objet o et un ensemble S d'objets, on définit
  $d(o, S) := min\{d(o, p) \mid p \in S\}$:

  - Choisir un point m_1 .
  - Choisir m_2 le point le plus éloigné de m_1 .
  - Choisir m_3 le point le plus éloigné de $\{m_1, m_2\}$.
  - Choisir m_4 le point le plus éloigné de $\{m_1, m_2, m_3\}$.
  - ...
  - Choisir m_k le point le plus éloigné de $\{m_1, m_2, \cdots m_{k-1}\}$.

Variante : Clustering K-médoïdes (suite)

Comparaison des méthodes *Kmeans* et *Kmedoids*

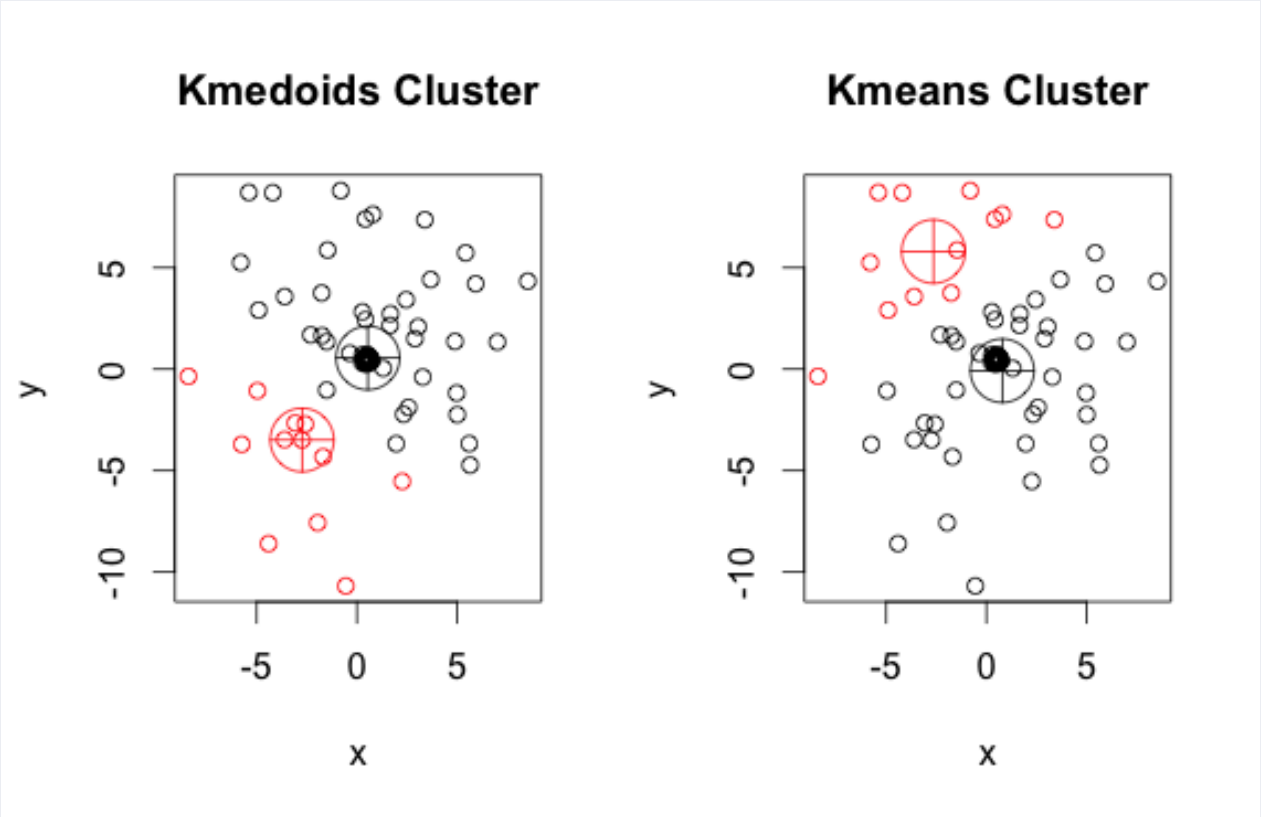


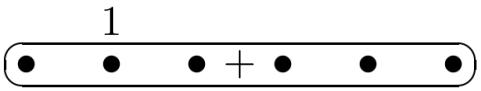
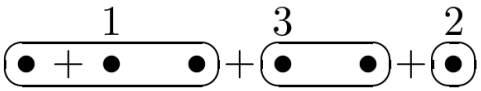
FIGURE 4: A gauche, les objets centraux sont des observations
A droite : pas forcément.

Variante : Clustering K-médoïdes (suite)

Exemple (6 points, k=3 avec les médoïdes m_1, puis m_2 puis m_3)

Rayon maximal (vs les centroïdes m_i)

(après le choix de m_i)

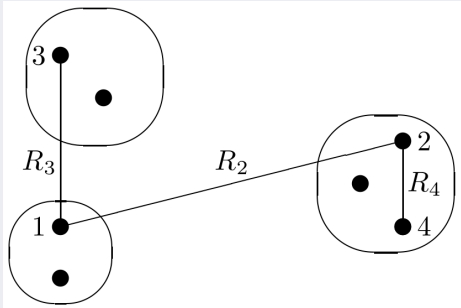
	Farthest First	Optimal(la croix +)
	4	2.5
	2	1
	1	0.5

- On choisit m_1 = le point no 1 (les numéros commencent à 0)
- m_2 = le plus éloigné de m_1 = le point 2
- m_3 = le plus éloigné de $\{m_1, m_2\}$ = le point 3

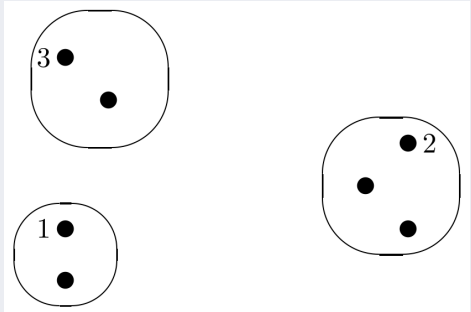
Variante : Clustering K-médoïdes (suite)

Propriétés de Farthest-First (pour $k = 3$) :

Exemple :
- Le premier point (1) est choisi au hasard.
- Le point (2) le plus éloigné de (1).
- Le point (3) le plus éloigné de {(1),(2)}



Farthest First : garanti des performances :
- Ici, le rayon de chaque cluster est $\leq R_4$
  ➛ Atteignabilité
- Meilleurs résultats que *K-means* de base
  (voir plus loin *k-means++*)



☞ Le choix de m_1 et m_2 les plus éloignés ➛ Risque de choisir des "outliers".

Variante de Farthest-First : K-means++

- Similaire à Farthest-First avec une stratégie de choix des médoïdes initiaux
- Évite le défaut de *Farthest-First* en choisissant des médoïdes éloignés les uns des autres tout en évitant de favoriser les outliers.
- Donne de meilleurs résultats (en erreur SSE)

- **Principe** (pour K clusters et un ensemble X de taille $\geq K$) :

  X peut être un (sous) ensemble des instances : un échantillon aléatoire.

  - Choisir d'abord le premier médoïdes m_1 aléatoirement
  - Ensuite, pour les autres médoïdes $x = m_k \in X, k = 2..K$:
    Choisir le prochain m_k avec une proba proportionnelle à $D(x)^2$ où

    $D(x)$ = le minimum de distance à un médoïde existant : $D(x) = min_{k'<K}\ dist(x, m_{k'})$

  ➛ Cette stratégie évite les outliers qui ont par définition une valeur élevée de $D(x)$ mais ne sont pas proches des autres points.

- Méthode disponible dans Weka (voir BE3).

De K-means à la méthode probabiliste EM

Relations entre K-means, Mixture Gaussienne et EM :

- K-emans est basé sur une méthode *Gaussienne multi-dimensionnelle* (avec dimensions indépendantes) utilisant des moyennes partagées (les centroïdes) et avec une variance identique (au seins d'un cluster, cf. SSE).
  - Les estimations s'obtiennent par une maximisation de la vraisemblance.
  - La probabilité (a priori) utilisée est une loi *uniforme*.
- La différence entre K-means et EM standard est qu'en recalculant les centroïdes, K-means re-pondère les espérances (*expectations*) telles que le centroïde le plus proche obtienne une espérance de 1.0 et tous les autres 0.0.

  ➙ Cette approche est également appelée la stratégie de maximisation d'espérance où "le gagnant ramasse tout" (*EM-winner-take-all*) dans la littérature sur EM.
- Voir plus loin la méthode EM.

Clustering Probabiliste

Rappels sur K-means :

- K arbitraire (nombre de clusters)
- *Écart type* (variance) artificiel pour les clusters (cf. SSE)
- Risque de voir chaque instance devenir un cluster tout seul
- Il faudra est que le résultat représente au moins un minimum local
  ➛ On sait que l'idéal est un minimum global
- Est-ce que la répétition de l'algorithme Clustering (et choix du meilleur) n'annule pas ce minimum local ?
- Ne détruit-il pas l'effet incrémental de l'algorithme (sur lequel on s'appuie) ?
- L'aspect Hiérarchique reflète-il le choix des meilleurs (bons) clusters ?
- etc.

Clustering Probabiliste (suite)

Et d'une manière plus générale :

- Difficulté inhérente aux algorithmes incrémentaux (itératifs) :
  - dépendance à l'ordre de prise en compte des instances ;
  - les éventuelles opérations de fusion et division peuvent-elles annuler l'effet des choix initiaux (centroïdes initiaux, k, ...) ?

L'idée du Clustering Probabiliste :

- D'un point de vu statistique, la tâche de Clustering est de trouver un ensemble de clusters les plus **vraisemblables**, étant donné une BD.

  ➙ et donc une meilleure espérance a priori (= *prior expectation*).

- En l'absence d'assez d'évidences (sur la décision de placer une instance définitivement dans tel ou tel cluster), on peut plutôt considérer

  la probabilité pour une instance d'être dans tel ou tel cluster

➙ C'est l'objectif du clustering probabiliste (dans sa réalisation *statistique* !).

Modèle de Mélange (Mixture) Finie

Clustering Statistique :

- La base de la méthode de Clustering Statistique est le modèle *mixture finie*.
- Une **mixture** est un ensemble de K distributions de probabilités qui représenteront les K clusters.
  - ➙ Chaque distribution est celle des instances d'un cluster .

  Plus exactement, chaque distribution $\Theta_{i=1..k}$ donne la probabilité pour une instance particulière d'avoir un certain ensemble de valeurs d'attributs *"S'il devait appartenir réellement au cluster C_i"*.

- Les clusters ne sont pas semblables :
  - Chaque cluster a une distribution qui représente sa population.
  - Chaque instance appartient à un seul cluster mais on ne sait pas lequel.

Un cas simple de Mélange Gaussienne

• **Exemple** : une BD, les instances ont un seul attribut explicatif numérique

☞ Deux clusters A et B.
☞ (Pour l'exemple) on connaît le cluster de chaque instance.

• Hypothèse : les clusters ont chacun une distribution **Normale** (gaussienne) avec des paramètres (μ, σ^2) potentiellement différents.

• La tâche est de calculer (μ, σ^2) pour chaque cluster afin de déterminer les population pour ces deux clusters.

• **Ici, la tâche est facile !**
◦ on a la BD, le cluster de chaque instance et
◦ on veut définir les paramètres des modèles qui ont généré ces instances

→ Calculer, pour le cluster A : μ_A, σ_A et P_A (idem pour B).

• Le modèle de Mélange (GMM = *Gaussian Mixture Models*) combine plusieurs distributions **Normales** (ici, 2).

☞ Cf. Ci-dessus sur K-means, Mixture et EM.

Un cas simple de Mélange Gaussienne (suite)

Exemple : $P_A + P_B = 1$

A	51	B	62	B	64	A	48	A	39	A	51
A	43	A	47	A	51	B	64	B	62	A	48
B	62	A	52	A	52	A	51	B	64	B	64
B	64	B	64	B	62	B	63	A	52	A	42
A	45	A	51	A	49	A	43	B	63	A	48
A	42	B	65	A	48	B	65	B	64	A	41
A	46	A	48	B	62	B	66	A	48		
A	45	A	49	A	43	B	65	B	64		
A	45	A	46	A	40	A	46	A	48		

FIGURE 5: Données de l'exemple de Mixture finie des clusters A et B (ici connus)

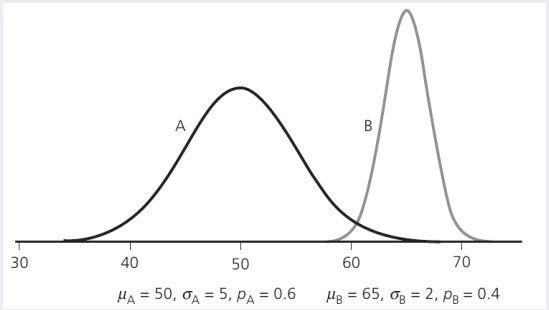


FIGURE 6: Distributions Normales de l'exemple de Mixture finie

Un cas simple de Mélange Gaussienne (suite)

Paramètres du modèle : 5 paramètres $\mu_A, \sigma_A, P_A, \mu_B, \sigma_B$, $P_B = 1 - P_A$.

• Si on connaît la *distribution* de chaque instance, **trouver les 5 params** serait facile (les x_i sont des instances de la distribution A ou B) :

- Soit $A = \{x_1, x_2, \cdots, x_{n_A}\}$ $\quad$ $B = \{y_1, y_2, \cdots, y_{n_B}\}$
- Estimer $\mu_A, \sigma_A, \mu_B, \sigma_B$ séparément par (pour A puis pour B) :

  - Moyenne de l'échantillon $\quad \mu_A = \frac{1}{n_A} \sum_1^{n_A} x_i$

  - Variance de l'échantillon $\quad \sigma_A^2 = \frac{1}{n_A - 1} \sum_1^{n_A} (x_i - \mu_A)^2$

- Si parmi n instances, n_A sont dans A et n_B dans B, alors $P_A = \frac{n_A}{n}$
- Pour B : $P_B = 1 - P_A$

• NB : $n_A - 1$ est une tech. d'échantillonnage ; même effet que n si n grand

Un cas simple de Mélange Gaussienne (suite)

• On trouve la probabilité du cluster de chaque instance, et surtout celle d'une nouvelle instance à l'aide des 5 paramètres :

➛ Pour une instance x, sa probabilité $Pr[A|x]$ d'appartenir au cluster A :

$$Pr[A|x] = \frac{Pr[x|A] \ . \ Pr(A)}{Pr[x]} = \frac{f(x;\mu_A,\sigma_A) \ . \ P_A}{Pr[x]}$$

Où $f(x;\mu_A,\sigma_A)$ est la densité de probabilité (la *vraisemblance*) pour la distribution *Gaussienne* de A avec :

$$f(x;\mu_A,\sigma_A) = \frac{1}{\sqrt{2\pi}\ \sigma_A} \ . \ e^{-\frac{(x-\mu_A)^2}{2\sigma_A^2}}$$

• Le traitement est le même que pour **Bayésien naïf** :
- traitement des attributs numériques et mixtes.
- le dénominateur $Pr[x]$ disparaitra (normalisation).

• NB : $f(x)$ n'est pas exactement la proba $Pr[x|A]$ (= 0 pour une valeur particulière) mais une vraisemblance et la normalisation permet d'avoir un résultat final correct.

Un cas simple de Mélange Gaussienne (suite)

Le point de vu génératif permet d'envisager, par le *Mélange (Mixture) Fini*, comment la BD a été générée.

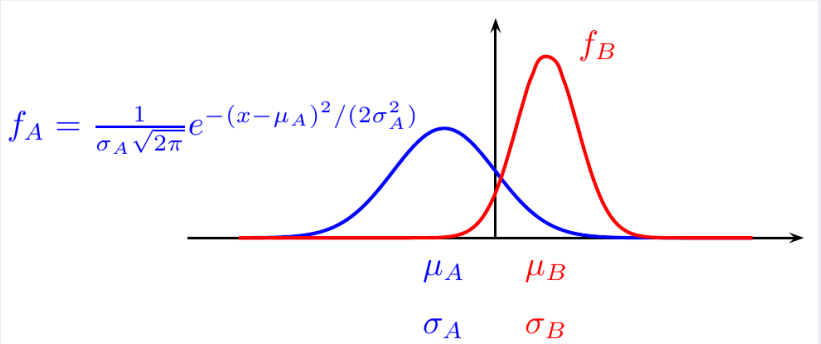
• Lorsque les 5 params sont connus, on peut générer n valeurs aléatoires de la manière suivante (sachant $P_A + P_B = 1$) :

1- Générer $P_A \times n$ valeurs selon la distribution f_A

2- Générer $P_B \times n$ valeurs selon la distribution f_B

Cette génération dépend de cinq paramètres :

$p_A, \mu_A, \sigma_A, \mu_B, \sigma_B$.



➞ On peut potentiellement **générer** chaque instance selon ces paramètres.

• Pour un nouveau point x', décider du cluster (comme un cas bi-classes de Bayes) à l'aide du rapport $Pr[A|x']/Pr[B|x']$ (➞ *si* > 1, *alors cluster=A, ...*).

La Méthode EM

- Dans l'exemple précédent, on a supposé connaître (et utilisé) les clusters.
  ➙ Considérons le problème inverse.

Hypothèse et intérêt du Clustering probabiliste :

On ne connaît pas les classes (clusters) des instances

- On ne connaît ni la distribution d'origine de chaque instance ;
- Ni donc les 5 params du modèle Mixture finie.

Une méthode : Espérance-Maximisation *(EM = Expectation Maximization) :*

- On commence par une valeur estimée pour chacun des 5 params
  Ici pour $k = 2$: on affect les instances aléatoirement dans 2 clusters.
- On calcule alors la probabilité du cluster de chaque instance (cf. ci-dessus),
- On utilise ces probabilités pour ré estimer les 5 params ...
- Répéter jusqu'à la maximisation de la vraisemblance (voir plus loin)

☞ Dans K-means, on a utilisé la technique similaire !

La Méthode EM (suite)

- **La méthode** *Expectation Maximization* :
  - ➙ étape *Expectation* : calcul des probabilités des clusters (expected class values) de chaque instance ;
  - ➙ étape *Maximization* : calcul des distributions et la maximisation de la vraisemblance des distributions étant donnée la BD.

Autrement dit :

① Estimer des paramètres du modèle à l'aide des instances
② Maximiser la vraisemblance que les données viennent de ce modèle.
→ Ce qui modifie les paramètres de l'étape ①
③ Répéter ces 2 étapes jusqu'à stabilisation

Remarques :

- Pour $k = 2$ clusters, estimation de "5 paramètres",
- Pour $k > 2$ clusters, on peut avoir recours à une technique de base comme K-means, pour trouver initialement un nombre raisonnable de clusters.
  - ➙ Voire, tester sur différents nombres de clusters obtenus par K-Means.

La Méthode EM (suite)

Déroulement de la méthode (cas bi-clusters) :

- But : décider du cluster des instances $x_i \in S$ (clusters $\in \{A, B\}$).
- On ne connaît aucun des cinq paramètres.

1. Répartir les instances dans A et $B = S \setminus A$ deux clusters de départ.
2. Calculer $P_A = \frac{n_A}{n}, \mu_A = \frac{\sum_1^{n_A} x_i}{n_A}, \sigma_A^2 = \frac{\sum_1^{n_A} (x_i - \mu_A)^2}{n_A - 1}, \mu_B, \sigma_B^2$
3. Expectation : Calculer $Pr(A|x) = \frac{f(x;\mu_A,\sigma_A).P_A}{Pr[x]}$ et $Pr(B|x), \forall x \in S$
4. Maximization de la vraisemblance : calculer (voir justification plus loin)

$$\mu_A = \frac{\sum_{x \in S} Pr(A|x) \,.\, x}{\sum_{x \in S} Pr(A|x)} \qquad \sigma_A^2 = \frac{\sum_{x \in S} Pr(A|x) \,.\, (x - \mu_A)^2}{\sum_{x \in S} Pr(A|x)} \qquad \text{et} \qquad P_A = \frac{\sum_{x \in S} Pr(A|x)}{n}$$

5. Idem pour B.
6. Aller à 3 (Jusqu'à stabilisation des 5 paramètres)

➛ Notons que le P_A de l'étape 2 n'est fait que pour l'initialisation.

La Méthode EM (suite)

Note importante : dans l'algorithme précédent, on a utilisé un ajustement mineure de la méthode EM

- Les valeurs sont des probabilités de clusters et non les clusters eux mêmes.
- Ces probabilités se comportent comme des pondérations :

Si w_i est la probabilité que l'instance x_i appartienne au cluster A, on aura :

$$\mu_A = \frac{w_1 x_1 + w_2 x_2 + \ldots w_n x_n}{w_1 + w_2 + \ldots + w_n} = \frac{\sum_i w_i \ . \ x_i}{\sum_i w_i} \qquad et \qquad P_A = \frac{\sum_i w_i}{n}$$

$$\sigma_A^2 = \frac{w_1 (x_1 - \mu_A)^2 + w_2 (x_2 - \mu_A)^2 + \ldots + w_n (x_n - \mu_A)^2}{w_1 + w_2 + \ldots + w_n} = \frac{\sum_i w_i \ . \ (x_i - \mu_A)^2}{\sum_i w_i}$$

On prend donc en compte **TOUS les** x_i et pas seulement ceux du cluster A.

La terminaison de l'algorithme EM

Comparons la méthode EM à K-means :

- Rappel : K-means s'arrête sur une sorte de *point fixe*
  ➛ quand les classes (clusters) des instances ne changent plus.

- Dans EM, c'est moins simple :
  - L'algorithme converge vers son point fixe mais ne l'atteint jamais.
  - Mais on peut savoir combien on y est proche en calculant la vraisemblance globale que les données **viennent effectivement de la BD** dont on a calculé les 5 params. On s'arrête quand un point acceptable est atteint.

- **La vraisemblance globale** est le produit des probabilités des instances indépendantes x_i (pour un cas bi-clusters).

- On peut démontrer que
  la vraisemblance générale augmente à chaque itération de l'EM. $\cdots \rightsquigarrow$

Addendum : Vraisemblance Générale

• Rappel : on a vu que $Pr[A|x_i] = \frac{Pr[x_i|A] \ . \ Pr(A)}{Pr[x_i]}$ (idem pour B).

Avec $Pr[x_i|A] = f(x_i; \mu_A, \sigma_A) = \frac{1}{\sqrt{2\pi}\ \sigma_A} \ . \ exp(-\frac{(x_i - \mu_A)^2}{2\sigma_A^2})$

➙ le dénominateur $Pr[x]$ utilisé pour la normalisation (cf. Bayésien).

• Cette somme ($Pr[x_i]$) correspond à la **probabilité de générer** x_i, étant donné les 5 paramètres (=normalisation pour obtenir des probas $\in [0, 1]$).

$$Pr[x_i] = Pr[x_i|A] \ . \ Pr[A] + Pr[x_i|B] \ . \ Pr[B]$$

• Or, en supposant l'indépendance des instances x_i, la probabilité de générer la BD est $Pr[x] = Pr[x_1].Pr[x_2]...Pr[x_n]$
où pour chaque x_i : $Pr[x_i] = Pr[x_i|A] \ . \ Pr[A] + Pr[x_i|B] \ . \ Pr[B]$

• Par ailleurs, la vraisemblance de la BD étant donnés les 5 paramètres θ est définie par : $\mathcal{L}(x; \theta) = Pr[x]$ ../..

Addendum : Vraisemblance Générale (suite)

- Ce qui donne :

$$\mathcal{L}(x;\theta) = Pr[x] = \prod_{i}^{n} \{P_A \times Pr[x_i|A] + P_B \times Pr[x_i|B]\}$$

$$\boxed{\mathcal{L}(x;\theta) = \prod_{i}^{n} \{P_A \times f_A(x_i) + P_B \times f_B(x_i)\}}$$

  ➛ $P_A \times f_A(x) + P_B \times f_B(x)$ ci-dessus $\sim$ probabilité de générer x
  (Comme pur le calcul Bayésien Naïf (cf. chap4-II))

- Le passage par *log* (simplifier les calculs, moins de perte de précision) :

$$\boxed{\ell_x = \sum_{i}^{n} log[P_A \times f_A(x) + P_B \times f_B(x)]}$$

- ℓ est une mesure de qualité du clustering :
  elle augmente à chaque itération de l'EM
  - Critère de terminaison de l'algorithme EM :
  - Jusqu'à stabilisation (voir addendum).

Addendum : La vraisemblance des paramètres θ

Rappelons l'objectif :

- On veut exprimer la vraisemblance $\mathcal{L}(x;\theta)$ en estimant $Pr[x] \approx \mathcal{N}(x;\mu,\sigma)$
- Pour $\theta = (\mu,\sigma)$, on souhait trouver une estimation $\hat{\theta} = (\hat{\mu},\hat{\sigma})$

- En terme de probabilité et loi *Normale*, on sait que la vraisemblance de $\hat{\theta}$ sachant $\{x_1, x_2, \ldots, x_n\}$ est $L(\theta|x_1,\ldots,x_n) = Pr[x_1,\ldots,x_n|\hat{\theta}]$.

- Or, si les x_i sont indépendants :

$$Pr[x_1,\ldots,x_n|\hat{\theta}] = Pr[x_1|\hat{\theta}] \times Pr[x_2|\hat{\theta}] \times \ldots Pr[x_n|\hat{\theta}]$$

$$\rightarrow \quad Pr[x_1,\ldots,x_n|\hat{\theta}] = \prod_1^n Pr[x_i|\hat{\theta}]$$

EM se stabilisera lorsque $L(\hat{\theta}|x_1,\ldots,x_n) \simeq Pr[x_1,\ldots,x_n|\hat{\theta}]$

$\rightarrow$ Lorsque $\hat{\theta} = \underset{\theta}{argmax}\ L(\theta|x_1,\ldots,x_n) = \underset{\theta}{argmax} \prod_1^n Pr[x_i|\theta]$

Par définition même de la vraisemblance en terme de probabilité Gaussienne. $\cdots \rightsquigarrow$

Addendum : La vraisemblance des paramètres θ (suite)

☞ L'égalité précédente est (presque) atteinte si les itérations ne modifient guère le vecteur θ.

• Pour simplifier les calculs de la vraisemblance générale, on utilise *log* :

➛ $\ell(\theta) = log\{L(\theta|BD)\} = log\{L(\theta|x_1, \ldots, x_n)\}$ et à l'aide de ci-dessus

➛ $\ell(\theta) = log\{Pr[BD|\theta]\} = \sum_i^n log\{Pr[x_i|\theta]\}$

Et si $Pr[x_i|\theta]$ suit une simple loi Normale, on devra calculer

$$\hat{\theta} = \underset{\theta}{argmax} \sum_i^n log\{Pr[x_i|\theta]\}$$

☞ Par ailleurs, la vraisemblance des données pour un cas bi-clusters :

$$\ell_{BD} = \sum_i^n log\left(P_A \times f_A(x_i) + P_B \times f_B(x_i)\right)$$

➛ On a noté $P_A = Pr[A]$ pour retrouver la notation précédente.

➛ EM itère jusqu'à ce que l'augmentation de L devienne négligeable (ε)

➛ Par ex. , 10 itérations successives avec un écart $\varepsilon = 10^{-10}$

Addendum : justification Mathématique

Justification pour le cas bi-clusters et uni-variante :

- Maximer le Log-vraisemblance revient à calculer la dérivée de l'expression :

$$\ell(\theta) = \sum_{i}^{n} log\{Pr[x_i|\theta]\}$$

sachant que $$Pr[x_i|\theta] = f(x_i; \mu_A, \sigma_A) = \frac{1}{\sqrt{2\pi}\ \sigma_A}\ .\ e^{-\frac{(x_i - \mu_A)^2}{2\sigma_A^2}}$$

- Pour chacun des x_i on a le terme (de la somme) $log\{Pr[x_i|\theta]\}$

$$\ell(\theta) = log\{Pr[x_i|\theta]\} = -\frac{1}{2}\ log(2\pi\sigma^2) - \frac{1}{2\sigma^2}(x_i - \mu)^2$$

$$\frac{\partial \ell(\theta)}{\partial \mu} = \frac{1}{\sigma^2}(x_i - \mu)$$

$$\frac{\partial \ell(\theta)}{\partial \sigma^2} = -\frac{1}{2\sigma^2} + \frac{(x_i - \mu)^2}{2\sigma^4}$$

../..

Addendum : justification Mathématique (suite)

- Le maximum se trouve où la dérivée est nulle :

$$\frac{\partial \ell(\theta)}{\partial \mu} = \sum_{i}^{n} \frac{1}{\sigma^2}(x_i - \hat{\mu}) = 0$$

$$\rightarrow \quad \sum_{i}^{n}(x_i - \hat{\mu}) = 0$$

$$\rightarrow \quad \hat{\mu} = \frac{1}{n}\sum_{i}^{n} x_i$$

- De la même manière, on trouvera :

$$\hat{\sigma}^2 = \frac{1}{n}\sum_{i}^{n}(x_i - \hat{\mu})^2$$

☞ On note bien que ces calculs fournissent les valeurs optimales des paramètres (maximisation de la log vraisemblance des paramètres de la loi Normale) qui à son tour permet, par EM, de maximiser la vraisemblance de la BD par $\ell_{BD} = \sum_{i}^{n} log\,(P_A \times f_A(x_i) + P_B \times f_B(x_i))$

Addendum : résumé de la méthode EM bi-classes

But : diviser l'ensemble S de n valeurs en deux clusters A et B (on ne connaît aucun des cinq paramètres) :

1. Choisir A et $B = S \setminus A$ deux clusters de départ.
2. Calculer $p_A, \mu_A, \sigma_A, \mu_B, \sigma_B$ (utiliser $f(x; \mu, \sigma)$)
3. Expectation (espérance) : Calculer $Pr(A|x)$ et $Pr(B|x)$ pour tout $x \in S$ (utiliser $f(x; \mu_A, \sigma_A)$ et P_A)
4. Maximisation : calculer (à l'ide de l'étape 3)

$$\mu_{A_{t+1}} := \frac{\sum_{x \in S} Pr(A|x) \,.\, x}{\sum_{x \in S} Pr(A|x)} \qquad \sigma^2_{A_{t+1}} := \frac{\sum_{x \in S} Pr(A|x) \,.\, (x - \mu_A)^2}{\sum_{x \in S} Pr(A|x)}$$

$$\text{et } P_{A_{t+1}} := \frac{\sum_{x \in S} Pr(A|x)}{n} \qquad \text{Idem pour } B.$$

5. Aller à 3 (Jusqu'à stabilisation)

Addendum : résumé de la méthode EM bi-classes (suite)

Bilan de la méthode EM bi-classes (suite)

Pour l'étape (3) : Espérance

- Une fois les 5 paramètres connus, on peut calculer le cluster de chaque x_i

$$Pr[A|x_i] = \frac{Pr[x_i|A] \times Pr[A]}{Pr[x_i]} \sim \frac{f_A(x_i) \times P_A}{Pr[x_i]} \qquad \text{De même pour } B.$$

Par ailleurs, on sait aussi calculer $Pr[A|x_i]$ via :

$$\frac{Pr[A|x_i]}{Pr[B|x_i]} = \frac{f_A(x_i) \times P_A}{f_B(x_i) \times P_B} \qquad \text{Avec } Pr[A|x_i] + Pr[B|x_i] = 1$$

➙ On a alors : $Pr[A|x_i] = \frac{f_A(x_i) \times P_A}{f_A(x_i) \times P_A + f_B(x_i) \times P_B}$

➙ $Pr[B|x_i] = \frac{f_B(x_i) \times P_B}{f_A(x_i) \times P_A + f_B(x_i) \times P_B}$

Rappel : comme dans Bayes, le dénominateur peut être ignoré.
Dans ce cas (bi-classes), ce sont les valeurs relatives des probabilités qui déterminent les clusters.

Addendum : un exemple d'EM (données manquantes)

- Retrouver des valeurs manquantes d'une BD (avec 4 valeurs x_1, x_2, x_3, x_4)
- La BD : uni-variable, attribut continue, loi Normale.
- On connaît $\frac{1}{2}$ des valeurs d'une BD
  ➛ L'autre moitié (x_3, x_4) manquante ou corrompue.
- Calculer les valeurs manquantes par EM (meilleures estimations)
- Hypothèse : la variance est unitaire (=1).
- **Étape E** : estimer les variables manquantes (μ puis $=x_3$ et x_4).
- **Étape M** : ré-estimer les paramètres de la distribution pour maximiser la vraisemblance des données, sachant la BD (complétée).

Addendum : un exemple d'EM (données manquantes) (suite)

Déroulement d'un exemple de maximisation d'espérance :

- Initialisation : BD $= [4, 10, ?, ?]$, $\sigma = 1$,
- On choisit $\mu_0 = 0$, ➙ (Nouvelle) BD : [4, 10, 0, 0]
  ($\mu = 0$ appliqué pour trouver les valeurs init des 2 manquantes)
- Nouvelle $\mu = 14/4 = 3.5$, ➙ Nouvelle BD : [4, 10, 3.5, 3.5]
- Nouvelle $\mu = 5.25$, ➙ Nouvelle BD : [4, 10, 5.25, 5.25]
- Nouvelle $\mu = 6.125$, ➙ Nouvelle BD : [4, 10, 6.125, 6.125]
- Nouvelle $\mu = 6.5625$, ➙ Nouvelle BD : [4, 10, 6.5625, 6.5625]
- Nouvelle $\mu = 6.7825$, ➙ Nouvelle BD : [4, 10, 6.7825, 6.7825]
- Nouvelle $\mu = 6.890625$, ...
- .....
- Nouvelle $\mu = 7$, Nouvelle BD : [4, 10, 7,7] ➙ Ne change plus.

Autres remarques sur l'algorithme EM

• Optimum local / global : l'algorithme EM converge vers un maximum local qui n'est pas nécessairement global.

➛ On peut répéter plusieurs fois les itérations avec des estimations initiales variées et choisir le plus grand maximum local.

• Veto : comme dans Bayes : le problème de véto de zéro
➛ même traitement.

• Loi Normale : l'hypothèse de la distribution Normale ne marche pas pour tous attributs :
- Par exemple, le poids (des personne) a un minimum.
- Dans des cas similaires, on utilise la distribution *Log-Normale*.
- Pour les valeurs discrètes numériques, on peut utiliser une distribution *Poisson*

➛ N.B. : ce remplacement (pour k clusters 1, 2, ...) de $\mathcal{N}(\mu, \sigma)$ par une loi de poisson pour le cas discret (dite *loi de bâtonnets*) $P(k) = \frac{\lambda^k}{k!} e^{-\lambda}$, $\lambda > 0$ peut se faire si le paramètre λ devient grand (> 5) auquel cas $\mu = \sigma = \lambda$.

Autres remarques sur l'algorithme EM (suite)

Généralisation d'EM :

• **Passage à plusieurs attributs** : le principe reste inchangé.

• Dans EM, on fait l'hypothèse de l'indépendance des attributs (comme pour Bayes).

• Si l'hypothèse d'indépendance n'est pas vérifiée (ou si des attributs corrélés existent), on utilise la distribution *bi-variée* :

➛ La moyenne inchangée ; mais on utilise la *co-variance* (pour notre BD basique).

➛ Extension à multi-variées si plusieurs attributs corrélés.

• **Passage à plusieurs classes** : voir "Compléments" (principe inchangé).

Complément : Généralisation à k clusters

- On a une BD : $\{x_1, x_2, ..., x_n\}$ dont on ne connaît pas les clusters (parmi k)
- On connaît (ou on estime) les poids des clusters :
$$Pr[w_1], Pr[w_2], Pr[w_3], \cdots, Pr[w_k]$$
- On ne connaît pas les moyennes $\mu_1, \mu_2, ..., \mu_k$

- On a : $L = Pr[BD|\mu_1, \mu_2, ..., \mu_k] = Pr[x_1, x_2, ..., x_n|\mu_1, ..., \mu_k, \sigma_1, ..., \sigma_k]$
$$= \prod_{i=1}^{n} Pr[x_i|\mu_1, ..., \mu_k, \sigma_1, ..., \sigma_k]$$
$$= \prod_{i=1}^{n} \sum_{j=1}^{k} Pr[x_i|w_j, \mu_1, ..., \mu_k, \sigma_1, ..., \sigma_k] \ . \ Pr[w_j]$$
$$= \prod_{i=1}^{n} \sum_{j=1}^{k} f(x_i; \mu_j, \sigma_j) \ . \ Pr[w_j]$$

➡ $$\ell = \sum_{i}^{n} log \sum_{j=1}^{k} f(x_i; \mu_j, \sigma_j) \ . \ Pr[w_j]$$

➡ Maximiser ℓ revient à résoudre $\frac{\partial}{\partial \mu_j} log(Pr[x_1, x_2, ..., x_n|\mu_1, \mu_2, ..., \mu_k]) = 0$

➡ idem $\partial\sigma_j$ sachant $\sum_{j=1}^{k} Pr[w_j] = 1$ ➡ (cf. résolution par Mult. Lagrange).

Complément : EM général : démarche

Exemple : une BD de n éléments mono-variés (x_i), on veut K clusters (C_j).

- On itère : à la t^{eme} itération, on constitue le vecteur
$$\Theta_t = \{\mu_1(t), \cdots, \mu_k(t), \sigma_1(t), \cdots, \sigma_k(t), P_1(t), \cdots, P_k(t)\}$$

Etape E : calculer les cluster (Expected) de tout x_i :

$$P[C_j|x_i, \Theta_t] = \frac{P[x_i|C_j, \Theta_t] \ . \ P[C_j|\Theta_t]}{P[x_i|\Theta_t]} = \frac{P[x_i|C_j, \mu_j(t), \sigma] \ . \ P_j(t)}{\sum_{j=1}^{c} P[x_i|C_j, \mu_j(t), \sigma] \ P_j(t)}$$

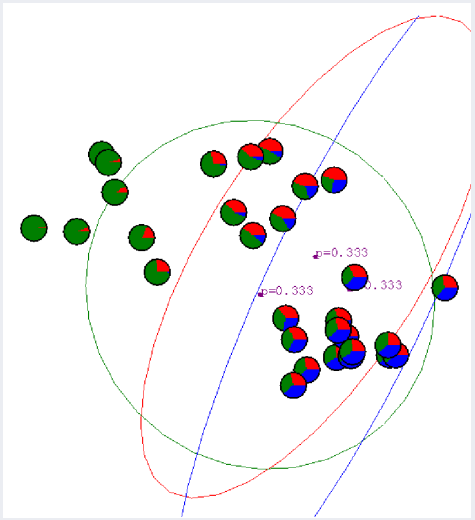
N.B. : $P[x_i|C_j, \mu_j(t), \sigma] = f(x_i; \mu_j, \sigma_j)$

Etape M : calculer le maximum de vraisemblance de μ sachant les distributions des clusters : $\mu_j(t+1) = \frac{\sum_k P[C_j|x_i, \Theta_t] \ . \ x_i}{\sum_k P[C_j|x_i, \Theta_t]}$

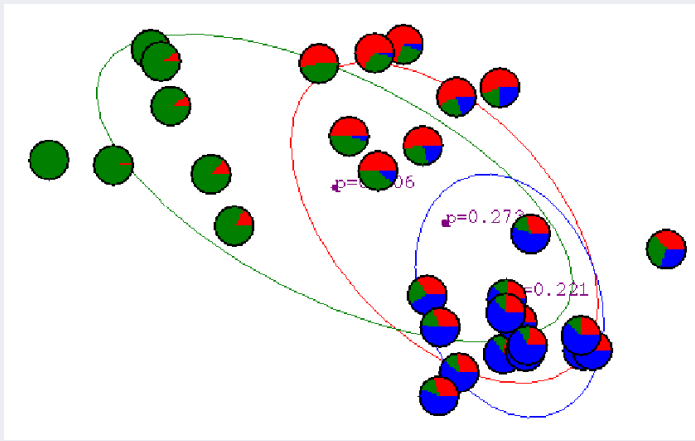
Idem $\sigma_j(t+1)$ et $P_j(t+1) = \frac{\sum_j P[C_j|x_i, \Theta_t]}{n}$

Terminaison : Quand il n'y a plus de changement des clusters.

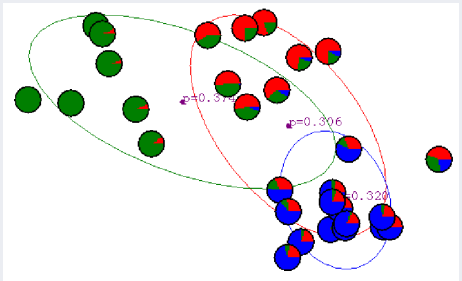
Compléments : Exemple avec 3 clusters



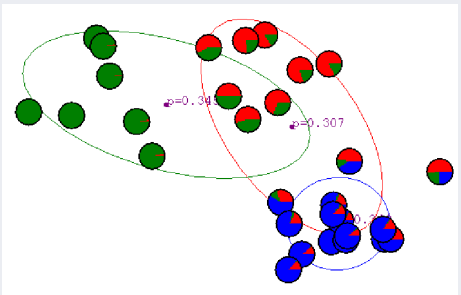
(a) Initialisation



(b) 1e itération

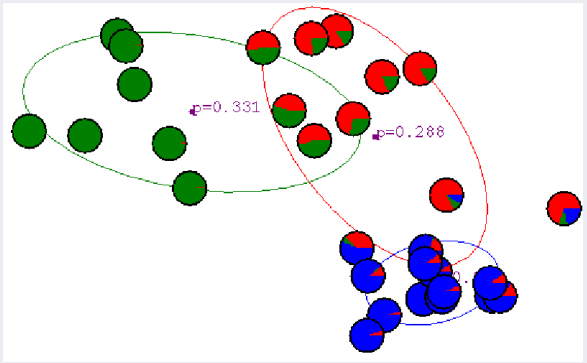


(c) 2e itération

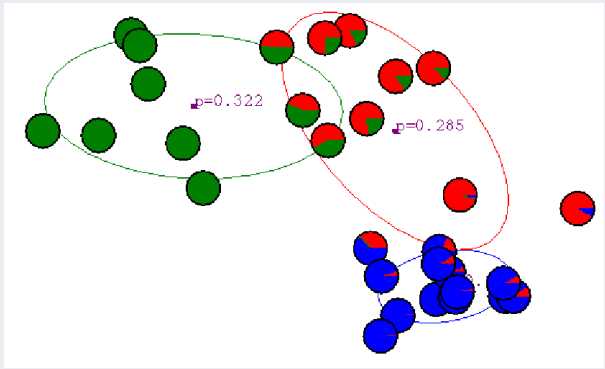


(d) 3e itération ../..

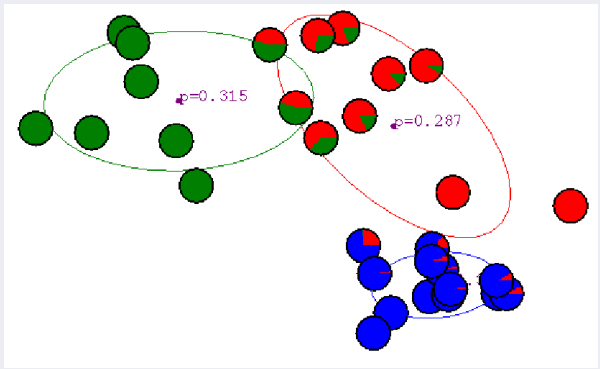
Compléments : Exemple avec 3 clusters (suite)



(e) 4e itération

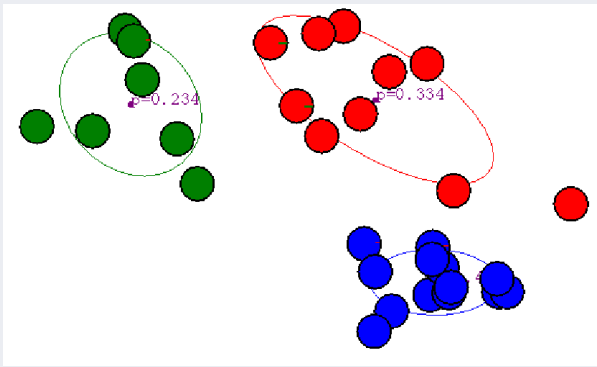


(f) 5e itération



(g) 6e itération ...

...



Après la 20e itération

Addendum : MLE et k clusters

- La généralisation à un cas de k cluster fait appel à la technique MLE.
- **MLE** : *Estimation du Maximum de Vraisemblance* (Maximum Likelihood Estimation).
- Technique importante en EC et en analyse de données
- MLE aussi intéressante (**pour "apprendre" des Gaussiens**)

Cas variance connue :

- Supposons avoir $x_1, x_2, ..., x_n \sim N(\mu, \sigma)$ où on ne connait pas μ mais on connaît σ

➙ Question : pour quelle valeur de μ les données $x_1, x_2, ..., x_n$ sont vraisemblables (MLE) ?

➙ En plus clair : pour quelle valeur de μ maximise-t-on $Pr[\mu|x_1, x_2, ..., x_n, \sigma^2]$ (MAP)

☞ La distribution est *Normale*. ../..

Addendum : MLE et k clusters (suite)

- On veut donc maximiser μ
- $\mu^{mle} = \underset{\mu}{argmax}\ Pr[\mu|x_1, x_2, ..., x_n, \sigma^2]$

on a : $Pr[\mu|X, \sigma^2] \propto Pr[X|\mu, \sigma^2].Pr[\mu|\sigma^2] \propto Pr[X|\mu, \sigma^2]$ (σ est connue).

$$= \underset{\mu}{argmax} \prod_{i=1}^{n} Pr[x_i|\mu, \sigma^2]$$

$$= \underset{\mu}{argmax} \sum_{i=1}^{n} log(Pr[x_i|\mu, \sigma^2]) \qquad log \text{ est monotone}$$

$$= \underset{\mu}{argmax} \sum_{i=1}^{n} log(f(x_i; \mu, \sigma)) \qquad \text{avec } f(x_i) = \frac{1}{\sqrt{2\pi}\ \sigma}\ exp(-\frac{(x_i - \mu)^2}{2\sigma^2})$$

$$= \underset{\mu}{argmax} \sum_{i=1}^{n} log\left(\frac{1}{\sqrt{2\pi}\ \sigma}\ exp(-\frac{(x_i - \mu)^2}{2\sigma^2})\right)$$

$$= \underset{\mu}{argmax} \sum_{i=1}^{n} \left(log(\frac{1}{\sqrt{2\pi}\ \sigma}) + log(\ exp(-\frac{(x_i - \mu)^2}{2\sigma^2}))\right) \qquad \frac{1}{\sqrt{2\pi}\ \sigma} \text{ est cste}$$

Addendum : MLE et k clusters (suite)

$$= \underset{\mu}{argmax} \quad n \times log(\frac{1}{\sqrt{2\pi}\ \sigma}) + \sum_{i=1}^{n} log(\ exp(-\frac{(x_i - \mu)^2}{2\sigma^2}))$$

N.B. : on a $log(e^{\alpha}) = \alpha$

$$= \underset{\mu}{argmax} \quad n \times log(\frac{1}{\sqrt{2\pi}\ \sigma}) + \sum_{i=1}^{n} \quad -\frac{(x_i - \mu)^2}{2\sigma^2}$$

- ➡ $n \times log(\frac{1}{\sqrt{2\pi}\ \sigma})$ est constante et n'affecte pas μ
- ➡ maximiser la somme négative revient à

$$\text{minimiser } (x_i - \mu)^2 \text{ ➡ } \underset{\mu}{argmin} \sum_{i=1}^{n} (x_i - \mu)^2$$

N.B. : Pour trouver le maximum (extrémum), on utilise la dérivée. ⋯ ⇝

Addendum : MLE et k clusters (suite)

MLE (suite) :

- $\mu^{mle} = \underset{\mu}{argmax}\ Pr[\mu|x_1, x_2, ..., x_n, \sigma^2]$

$$\rightarrow \quad = \underset{\mu}{argmin} \sum_{i=1}^{n} (x_i - \mu)^2$$

- On obtient μ^{mle} en posant $\frac{\partial}{\partial \mu} \sum_{i=1}^{n} (x_i - \mu)^2 = 0$

- $$\frac{\partial}{\partial \mu} \sum_{i=1}^{n} (x_i - \mu)^2 = 0$$

$$-\sum_{i=1}^{n} 2(x_i - \mu) = -2\sum_{i=1}^{n} (x_i - \mu) = 0$$

$$\rightarrow \quad \sum_{i=1}^{n} (x_i - \mu) = \sum_{i=1}^{n} (x_i) - n \times \mu = 0$$

$$\rightarrow \quad \mu^{mle} = \frac{\sum_{i=1}^{n} x_i}{n} \qquad \textbf{pour } \sigma \textbf{ connue.}$$

Addendum : MLE et k clusters (suite)

MLE (suite) :

$$\mu^{mle} = \frac{\sum_{i=1}^{n} x_i}{n} \qquad \textbf{pour } \sigma \textbf{ connue.}$$

➛ **La meilleure estimation de la moyenne d'une distribution est la moyenne de l'échantillon !**

• NB : si σ n'est pas connue, on trouve après calculs $\sigma^2_{mle} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu^{mle})^2$

➛ Et pour la cas multi-variante, on calculera :

$$\mu^{mle} = \frac{1}{n}\sum_{i=1}^{n}\overrightarrow{x_i} \qquad \sigma^2_{mle} = \frac{1}{n}\sum_{i=1}^{n}(\overrightarrow{x_i} - \mu^{mle})(\overrightarrow{x_i} - \mu^{mle})^T$$

Addendum : MLE et k clusters (suite)

Résumé de la technique générale pour trouver MLE :

• Trouver MLE des paramètre $\overrightarrow{v}$ sachant la forme de $Pr[BD|\overrightarrow{v}, autres\ infos]$:

1- Ré écrire $\ell = log(Pr[BD|\overrightarrow{v}, autres])$:
2- Calculer et résoudre $\frac{\partial \ell}{\partial \overrightarrow{v}} = 0$
3- Vérifier que c'est un maximum

• On peut également utiliser le *multiplicateur de Lagrange* si équation sous contraintes (cf. $\sum probas = 1$)

Addendum : Remarques sur EM

- Si la variable x a une distribution *Normale* $N(\mu, \sigma)$, la fonction de densité

$$\mathbf{f(x)} = \frac{\mathbf{1}}{\sqrt{\mathbf{2\pi}}\ \sigma}\ \exp(-\frac{\mathbf{(x-\mu)^2}}{\mathbf{2\sigma^2}})$$

- La probabilité qu'une valeur se trouve dans l'intervalle $[a, b]$ est donnée par

$$\int_a^b f(x).dx \qquad \text{et pour } \varepsilon \text{ petit } \int_{\alpha-\frac{\varepsilon}{2}}^{\alpha+\frac{\varepsilon}{2}} f(x).dx \approx \varepsilon.f(\alpha)$$

- On a

$$\int_{-\infty}^{\infty} f(x).dx = 1, \qquad \int_{\mu-\sigma}^{\mu+\sigma} f(x).dx = \approx 0.68$$

$$\int_{\mu-2\sigma}^{\mu+2\sigma} f(x).dx = \approx 0.95$$

- Différence entre la Vraisemblance et la Probabilité :

➛ Une probabilité est, par définition statistique, une fréquence à laquelle un évènement a lieu quand le nombre d'expériences tend vers l'infini.

➛ La fonction de densité de proba est en fait une vraisemblance car sa valeur dépend des paramètres (ici, μ et σ).

Addendum : Remarques sur EM (suite)

Fonction de vraisemblance d'un modèle paramétré :

• Étant donné une famille de fonction s de densités de probas (ou masses de proba dans le cas de distributions discrètes)

$x \mapsto f(x \mid \theta)$, Où θ est le paramètre.

• La fonction de vraisemblance est $\theta \mapsto f(x \mid \theta)$,

Réécrite par $\mathcal{L}(\theta \mid x) = f(x \mid \theta)$, Où x est l'observation.

• En d'autres mots, lorsque $f(x|\theta)$ est vue comme une fonction de x avec θ fixé, elle sera une fonction de densité de proba et si elle est observée comme une fonction de θ avec x fixé, elle sera une fonction de vraisemblance.

☞ $\mathcal{L}(\theta \mid x)$ n'est pas la proba que ces paramètres soient justes, étant donné l'observation x, mais sa vraisemblance.

Tenter d'interpréter la vraisemblance d'une hypothèse étant donné l'observation comme une proba de l'hypothèse est une erreur courante avec des conséquences parfois graves (e.g. en médecine).

Classification Hiérarchique

- Produit un ensemble de clusters imbriqués organisé comme un arbre (Hiérarchique).
- Peut être visualisé comme un **dendrogram** (hiérarchie de classes)
- **Exemple** : un diagramme arborescent qui représente une séquence de fusion-partitions

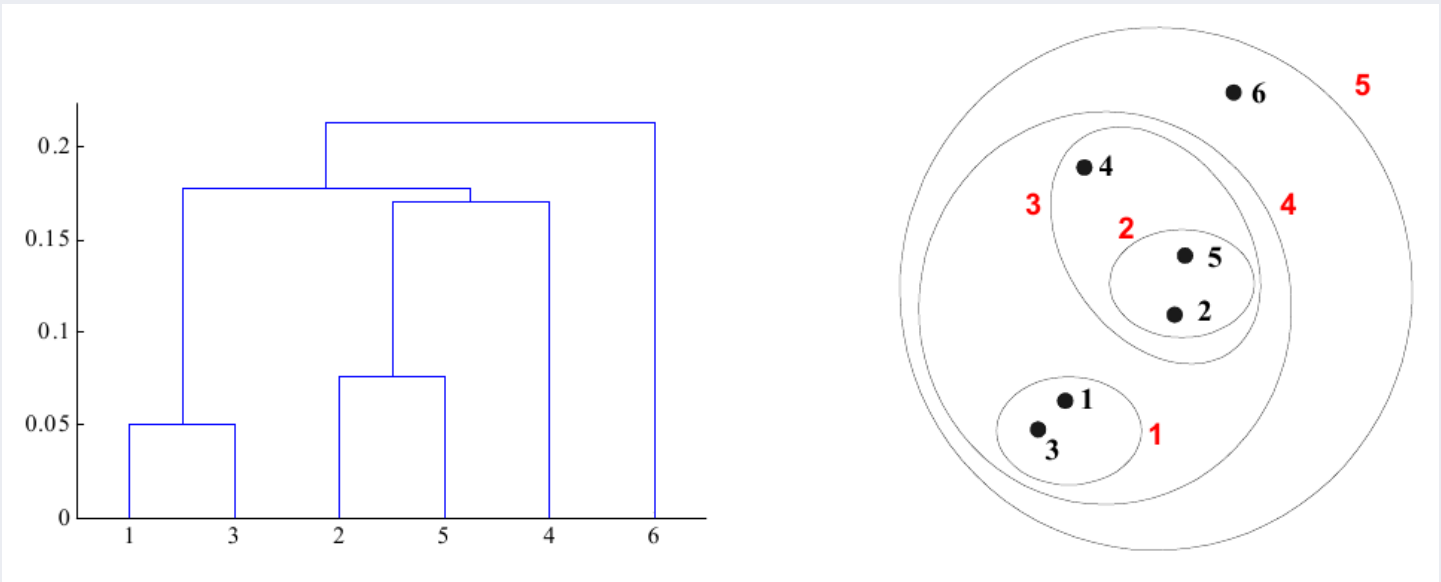
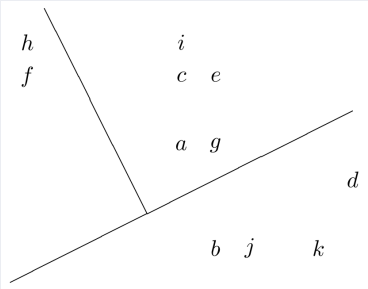


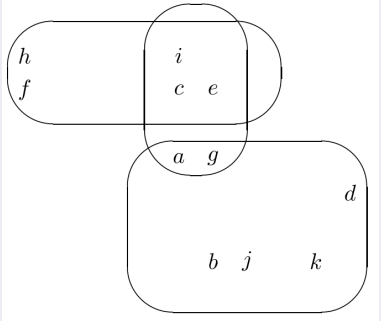
FIGURE 7: exemple de dendrogram

Classification Hiérarchique (suite)

Quelques types de clustering



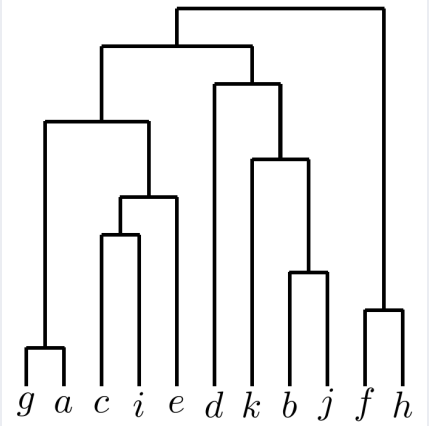
Clustering "ordinaire" (cf. *Kmeans*)



Hiérarchie de clusters

	1	2	3
a	0.3	0.4	0.3
b	0.1	0.2	0.7
c	0.4	0.5	0.1
$\vdots$		$\vdots$	

Probabiliste : $'a'$ *liés à 3 clusters*



Hiérarchique : les 3 clusters de $'a'$

Classification Hiérarchique (suite)

Caractéristiques :

- On ne connaît pas (par avance) le nombre de clusters
  - ➙ Le nombre de clusters désirés s'obtient en "coupant" le dendrogram.

Dendrogramme et performances du Clustering

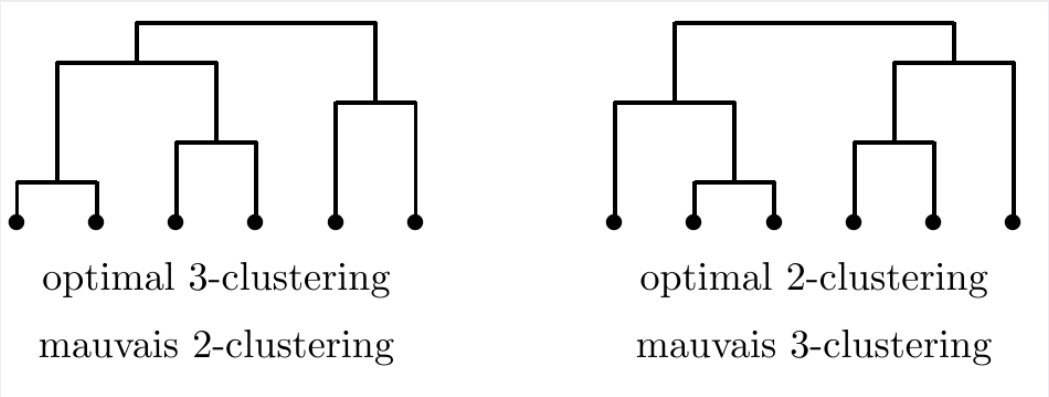


FIGURE 8: Exemples de coupe de dendrogram

N.B. : La structure hiérarchique (donnant un dendrogramme) peut correspondre à une taxonomie / ontologie réelle

- ➙ E.g. : dans le cas des animaux / objets structurés, etc.

Classification Hiérarchique (suite)

Il y a **deux types** principaux de clustering Hiérarchiques :

➙ Classification Agglomérative (ascendante) :
- ➛ Débute avec un point (instance) par cluster
- ➛ A chaque étape, on fusionne une paire de clusters "proches" jusqu'à l'obtention de k (désiré) clusters ; voire un seul !

➙ Classification Divisive (descendante) :
- ➛ Commence par un seul cluster contenant tous les éléments
- ➛ A chaque étape, on divise un cluster en deux jusqu'à l'obtention de k (désiré) clusters ; voire un par point par cluster !
- ➛ cf. *Dichotomique*

• La plupart des algorithmes hiérarchiques utilise une **matrice de similarité** (ou distance) pour fusionner deux clusters ou partitionner un cluster à la fois.

Classification Hiérarchique Agglomérative

Algorithme basique de la méthode **Agglomérative** :

Données : D l'ensemble d'apprentissage ; la valeur k
Résultats : Un ensemble de k clusters

Calculer la matrice de proximité (similarité) entre les points
Placer un point ***(une instance) par cluster***

Répéter
    Fusionner deux clusters proches
    Mettre à jour la matrice de proximité
        (qui doit représenter les distances entres clusters)
Jusqu'à k clusters ($k \geq 1$)

- Élément clef : distance entre deux clusters
  ➛ Différentes définitions donnent différents algorithmes.

Exemple Clustering Agglomérative

Déroulement : on commence par des clusters singletons
+ la matrice de similarité (ou de distances).

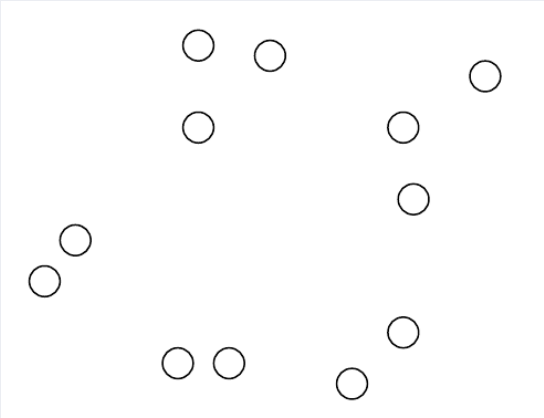


Fig : Les points à classer

	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						

Matrice de proximité des points

Le dendrogramme initial →

Clusters initiaux (singletons)

Exemple Clustering Agglomérative (suite)

Clustering Agglomérative (suite déroulement).

➙ Soit les clusters de l'étape i :

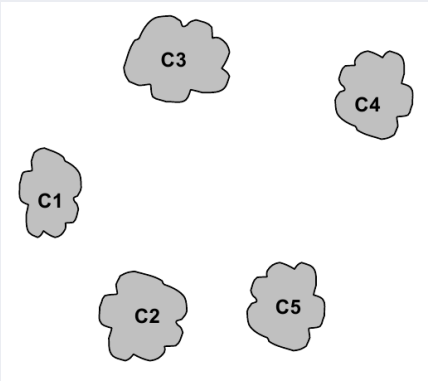
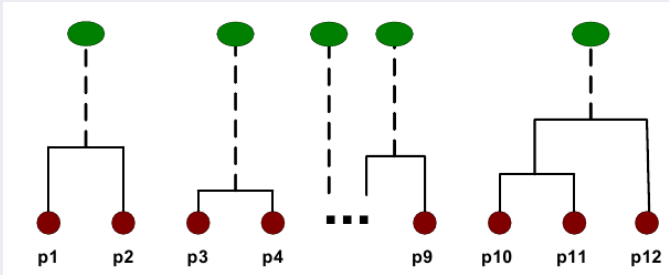


Fig : Les clusters de l'étape courante

	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Matrice de proximité (des clusters)



Représentation Hiérarchique des clusters

Exemple Clustering Agglomérative (suite)

Clustering Agglomérative (suite déroulement)...

Fusion de deux clusters et la MAJ de la matrice :

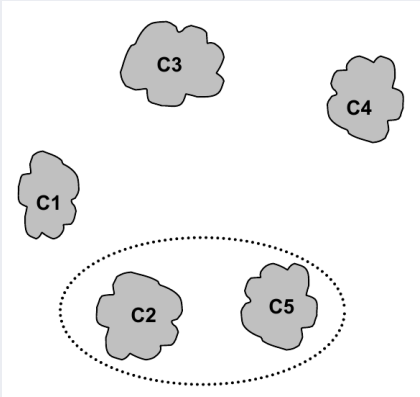
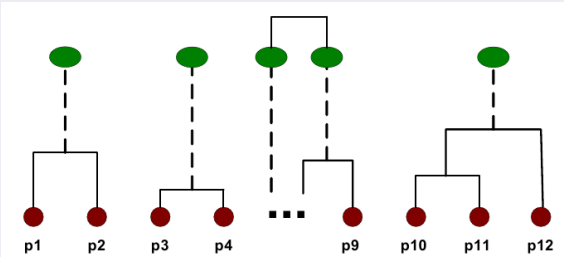


Fig : Fusion de deux clusters

	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

⇓ *Fusion de la matrice de proximité* ⇓



Représentation Hiérarchique de la fusion

	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

Représentation Hiérarchique de la fusion

La similarité inter-classes

Pour fusionner, on utilise un critère de distance (inv. *similarité*)

- Critères de similarité entre deux clusters (pour la fusion) basé sur :
  - **Minimum** des distances entre tous couples de points (un par cluster)
  - **Maximum** des distances entre tous couples de points (un par cluster)
  - **Moyenne** des distances de tous couples (un par cluster)
    ➛ Entre les **centroïdes / médoïdes** de chaque groupe

- Ces mesures doivent satisfaire une fonction objective :
  ➛ Ex : carré des erreurs (Méthode de *Ward*, v. plus loin)

- N.B. : différents types de similarités utilisées :
  - **Lien Simple** (*Single Link*)
    ➛ entre 2 singletons : entre *Min*, *Max*, *Médoïdes*,
  - **Lien Complet** (*Complete Link*) :
    ➛ entre deux groupes : entre *Centroïdes* et la *moyenne* de la distance de tous les points 2-à-2.

La similarité inter-classes (suite)

- Min des distances entre tous couples de points (un par cluster) ;
- Max des distances entre tous couples de points (un par cluster) .

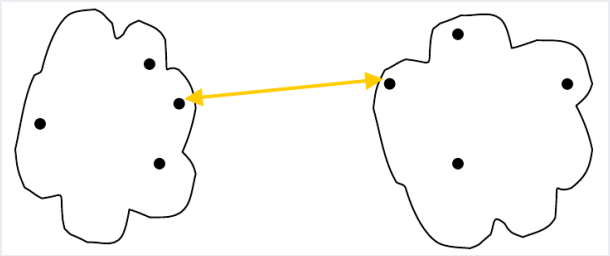


Fig : Distance Minimum

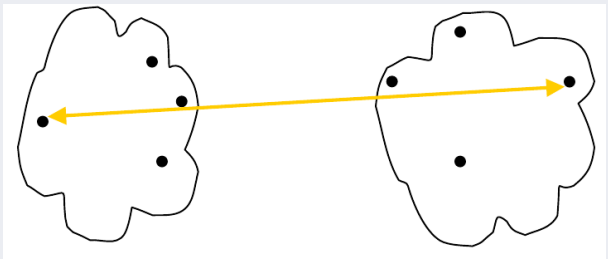


Fig : Distance Maximum

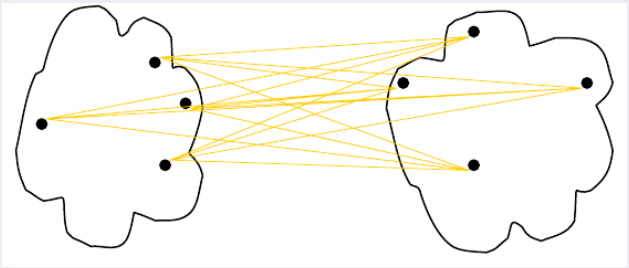
	p1	p2	p3	p4	p5	...
p1						
p2						
p3						
p4						
p5						
.						
.						

La matrice de proximité des points

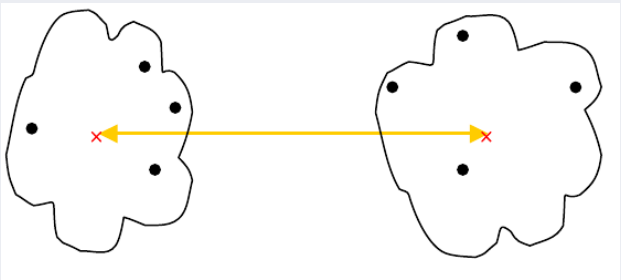
La similarité inter-classes (suite)

La similarité inter-classes (suite) :

- Moyenne des distances entre tous couples (un par cluster)



- Entre les centroïdes de chaque groupe



- Ces deux mesures utilisent la matrice de proximité des points.

Comportement des similarités inter-classes

☞ Il n'y a pas une distance qui fonctionne dans tous les cas !

- La distance Min (*single Link*) donne de **bons résultats**.
  - ➛ La similarité de 2 clusters est basée sur celle des deux plus similaires (proches) points, un dans chaque cluster.
  - ➛ Est déterminée par la distance minimum entre tout couple (un par cluster) par le graphe (ou la matrice) de similarité (vu ci-dessus).

- La distance Min est susceptible aux "bruits" et aux "outliers".

	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00

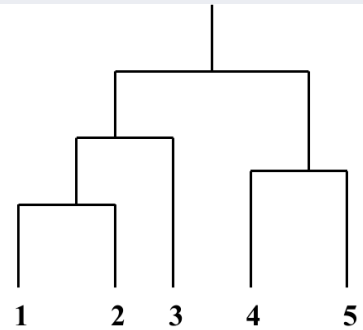


FIGURE 9: *Matrice de proximité et le Dendrogramme*

- N.B. : une matrice de similarité diagonale suffirait !

Comportement des similarités inter-classes (suite)

- La distance Max (entre deux clusters) tend à éviter des clusters larges.
- Elle est biaisée par des nuages circulaires (ne sait pas bien les séparer !) :

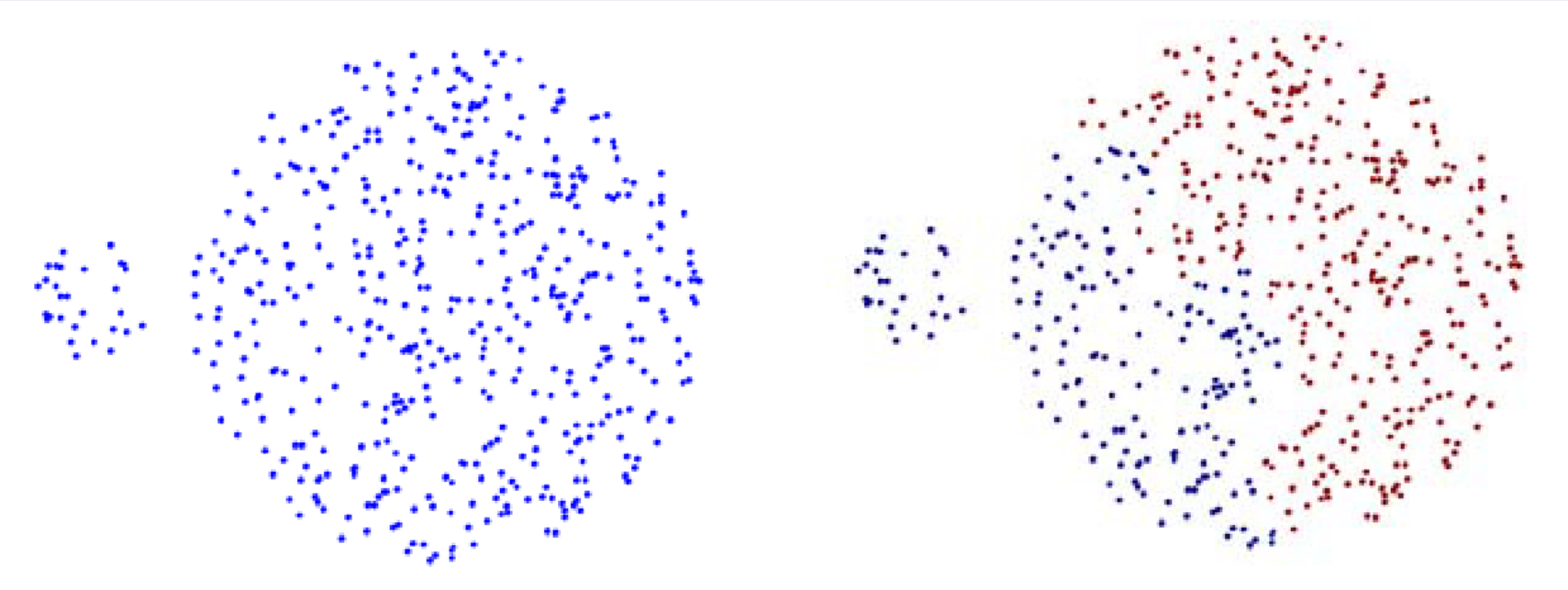
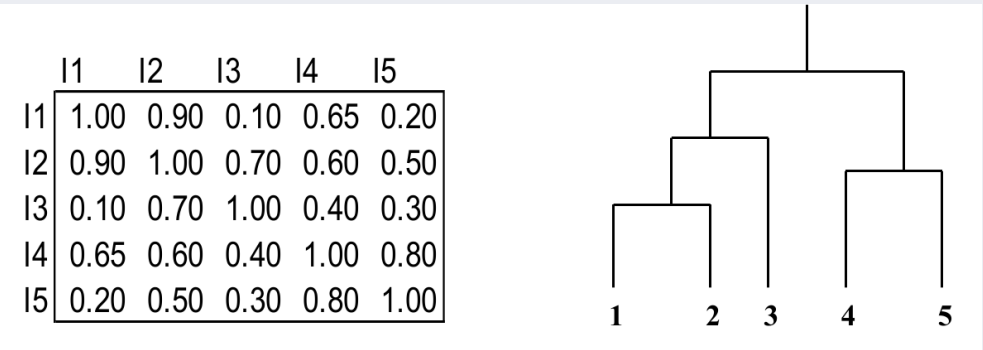


FIGURE 10: Nuage d'origine et les clusters produits

Comportement des similarités inter-classes (suite)

- La **Moyenne** des distances entre tous couples : $Dist(C_i, C_j) = \dfrac{\sum_{\substack{p \in C_i \\ q \in C_j}} Dist(p, q)}{[C_i| \times |C_j|}$



	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00

FIGURE 11: Matrice de proximité et le Dendrogramme

- Atténue l'effet des distances directes (qui favorisent la constitution de clusters larges).
- Est un compromis entre les distances *Min* et *Max*
- Peu sensible aux "bruits" et "outliers"
- Est biaisée par les nuages circulaires

Comportement des similarités inter-classes (suite)

Méthode de Ward (pour minimiser l'erreur) :

• Dans cette méthode, la décision de fusion entre 2 clusters est basée sur la croissance de l'erreur au carré quand les 2 clusters sont fusionnés.

➛ Est comme la distance moyenne des groupes lorsque la distance entre les points est levée au carré.

• Est moins susceptible aux "bruits" et aux "outliers"

• Est biaisée par les nuages circulaires

• Analogue à un K-means hiérarchique
➛ Peut être utilisée pour initialiser K-means

Comportement des similarités inter-classes (suite)

Une comparaison des distances (clusters départagés selon le type de distance) :

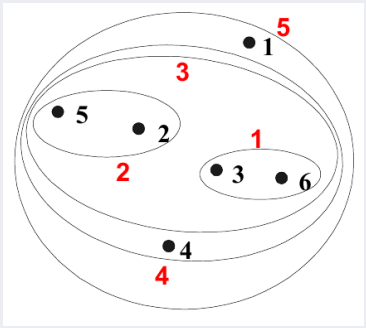
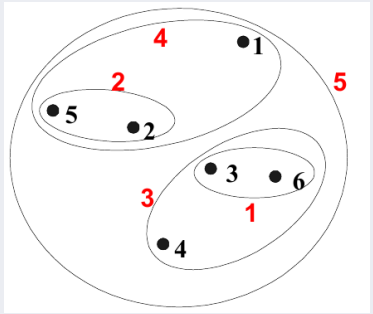
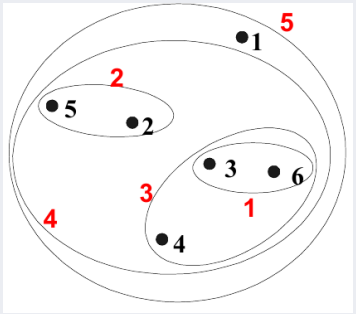


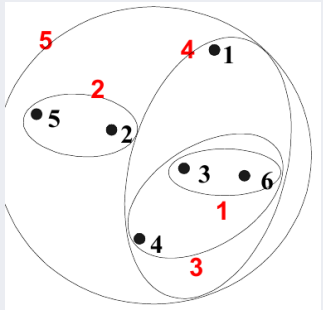
Fig : Distance Min



Distance Max



Moyenne du groupe



Méthode Ward

Remarques générales (Clustering Hiérarchique)

- **Complexité** $O(N^2)$ en **espace** :
  ➡ matrice de proximité avec N = nbr de points

- $O(N^3)$ **en temps** : N étapes et chaque étape $O(N^2)$
  ➡ Peut être ramené à $O(N^2\ log(N))$ dans certaines approches (optimisées).

Inconvénients du Clustering Hiérarchique :

- Une fois une décision (fusion, division) prise, elle ne peut pas être révoquée.
- Pas de véritable fonction objective (globale) à minimiser

- Les différentes approches ont des problèmes avec :
  - Sensibilité aux "bruits" et "outliers"
  - Difficulté avec des nuages de forme/taille différentes
  - Tendance à réduire les grands clusters

Classification Hiérarchique Divisive

En 2 étapes : (1) construction MST (un seul cluster) puis (2) division.

Étape 1 : Construction MST (Arbre de recouvrent de poids min) :

- Commencer le MST A en partant de n'importe quel point
- Chercher les paires (p, q) les + proches tels que $p \in A$ et $q \notin A$ (sinon : circuit) ; Ajouter q à A et établir l'arête qui l'y relie.

Étape 2 : Division (couper certains liens sur le MST construit)

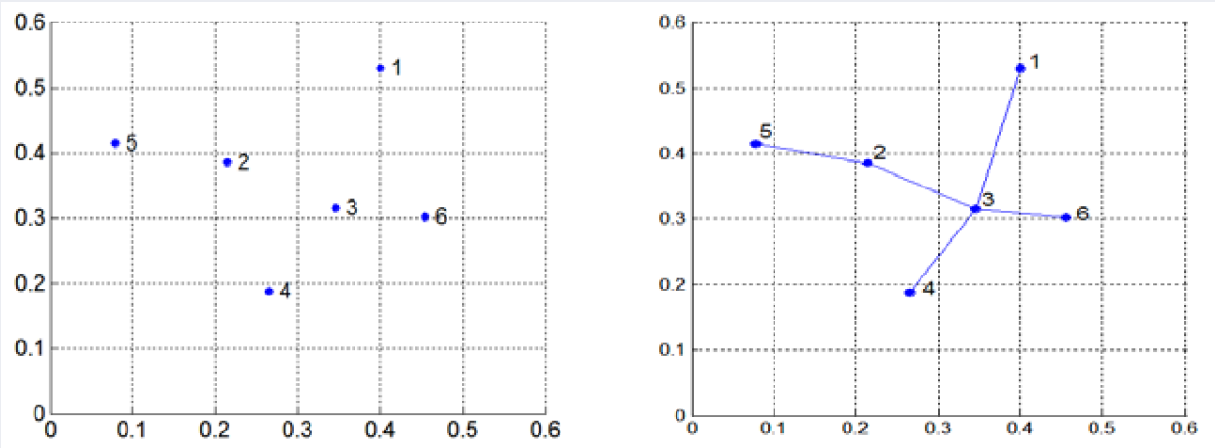


FIGURE 12: Construction du MST

Classification Hiérarchique Divisive (suite)

Algorithme de principe Classification Hiérarchique Divisive :

1. Construire MST (étape 1)
2. Répéter

   Créer un nouveau cluster en coupant le lien correspondant à la distance la plus grande (ou la plus petite similarité)
3. jusqu'à l'obtention de k cluster ($k \geq 1$)

N.B. : construction du MST par différents algorithmes (Kruskal, PRIM, ...)

➙ Quelques rappels / lectures ! ....

Clustering Semi Hiérarchique : DBSCAN

• DBSCAN est un algorithme basé sur la densité des clusters.

Quelques définitions (pour DBSCAN) :

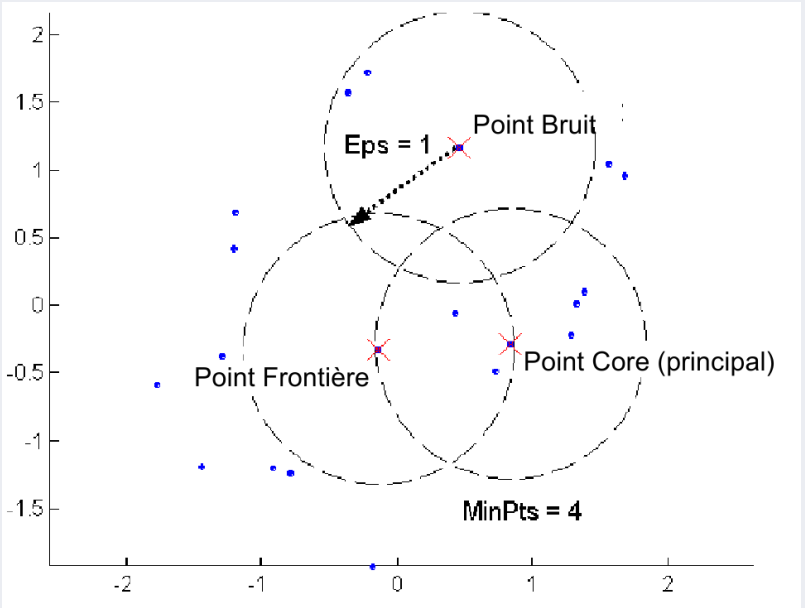
• **Densité** : nombre d'instances dans un cercle appelé (**Eps**) d'un rayon donné,

• **Point Core** (ou **dense**) : un point d'une région qui a plus d'un nombre prédéfini (**MinPts**) de points dans *Eps*.

➙ Ce sont les instances à l'intérieur d'un cluster,

• **Point Frontière** : un point d'une région qui a moins de **MinPts** dans un *Eps* mais qui est au voisinage d'un point *Core*,

• **Point bruit** : tout point qui n'est ni *Core* ni *Frontière*



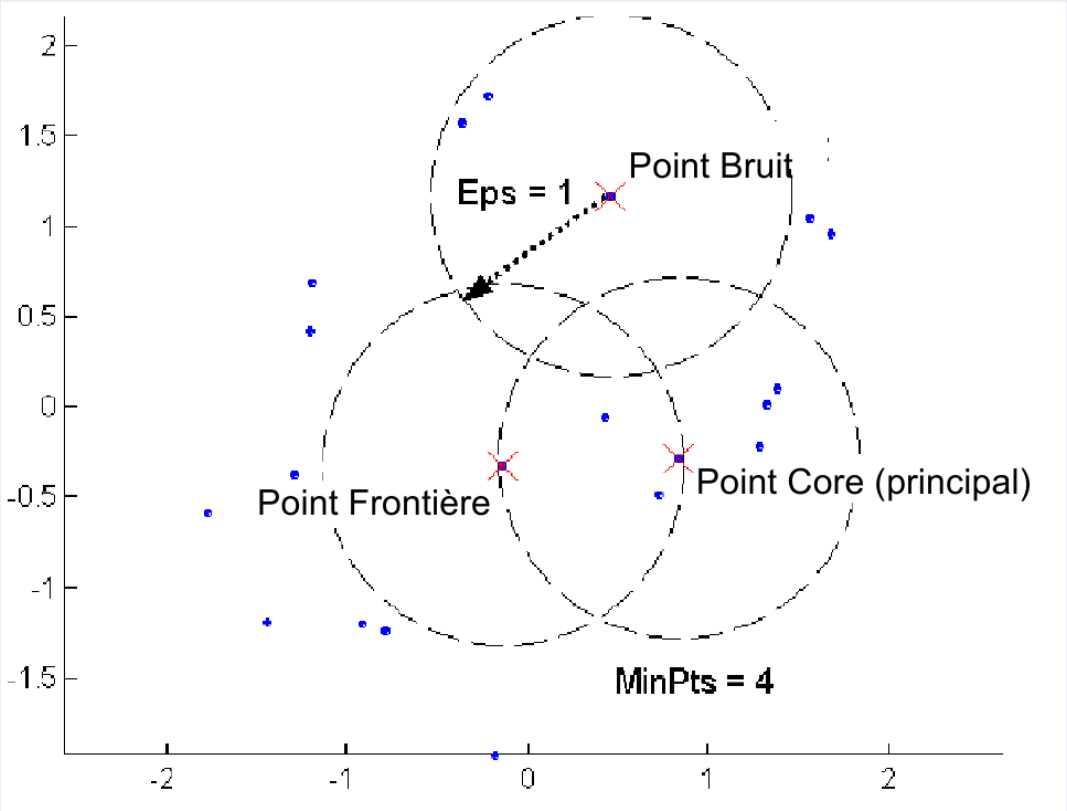
Clustering Semi Hiérarchique : DBSCAN (suite)

Notions DBSCAN :

- **Core (dense)** : dans un cluster d'au moins **Minpts** éléments

- **Frontière** : moins que *Minpts* points mais au voisinage d'un Core

- **Bruit** : ni Core ni Frontière



Principe de DBSCAN

Principe de DBSCAN :

- Fixer un seuil s et une densité δ.
- Un point $\overrightarrow{x}$ est **dense** s'il y a au moins δ autres points $\overrightarrow{y}$ qui satisfont $dist(\overrightarrow{x}, \overrightarrow{y}) \leq s$.
- Critères de Clustering :
  1. Si $dist(\overrightarrow{x}, \overrightarrow{y}) \leq s$ et $\overrightarrow{x}$ est *dense*
     Alors $\overrightarrow{x}$ et $\overrightarrow{y}$ appartiennent au même cluster.
  2. Si $\overrightarrow{x}$ et $\overrightarrow{y}$ appartiennent au même cluster et $dist(\overrightarrow{y}, \overrightarrow{z}) \leq s$ et $\overrightarrow{y}$ est *dense*
     Alors $\overrightarrow{x}$ et $\overrightarrow{z}$ appartiennent au même cluster (transitive)
  3. Deux points appartiennent au même cluster seulement si les 2 règles ci-dessus l'imposent.
- Cf. Atteignabilité (vu au début de ce chapitre).

DBSCAN

Algorithme DBSCAN :

- On commence par appliquer les critères ci-dessus, puis on élimine les bruits résiduels de ces critères,

- Ensuite, on procède à un Clustering sur les points restants par l'algorithme suivant :

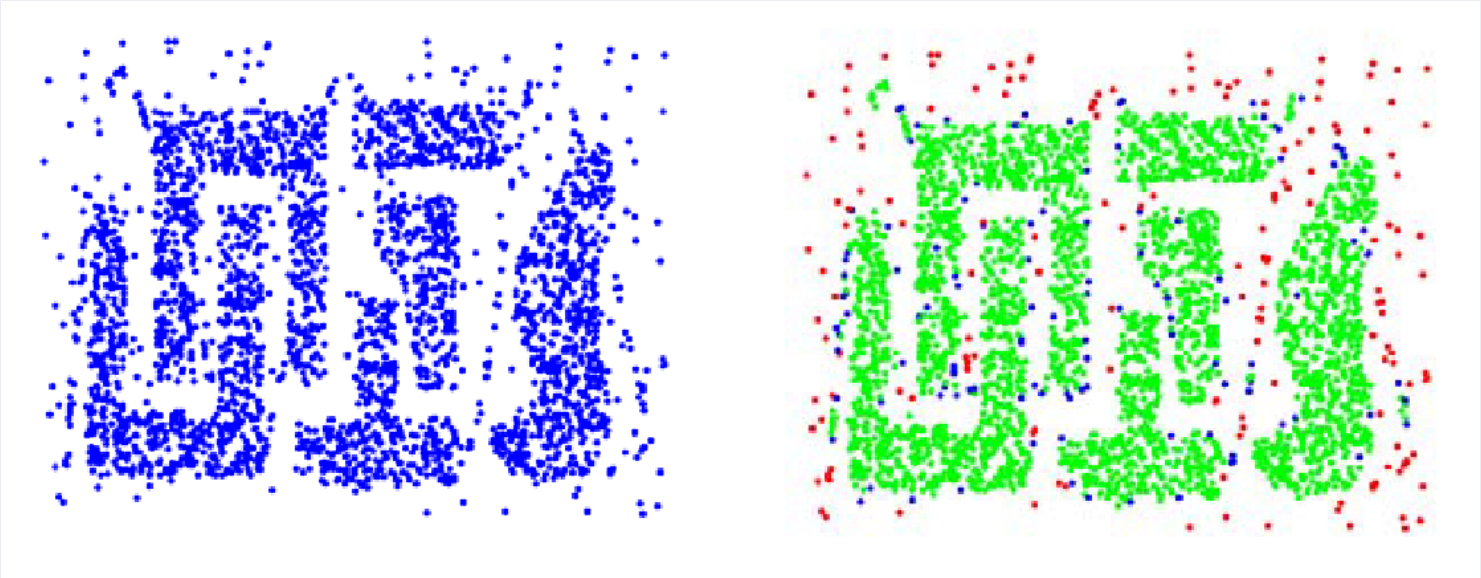
```
Données : D l'ensemble d'apprentissage
Résultats :  Un ensemble de k clusters

Lable_cluster_actuel := 1
Pour Tous points Dense
    Si ce point Dense n'a pas de label de cluster
        Lable_cluster_actuel := Lable_cluster_actuel +1
        Donner le label Lable_cluster_actuel au point Dense courant

    Pour Tous points dans le voisinage d'Eps sauf le ième (le point lui même)
        Si le point n'a pas de label
            Lui donner le label point "bruit"
```

DBSCAN (suite)

Exemple :



DBSCAN : points d'origine et points dense , frontière , bruit

(Eps = 10, MinPts=4)

DBSCAN (suite)

Un exemple (quand DBSCAN marche !) :

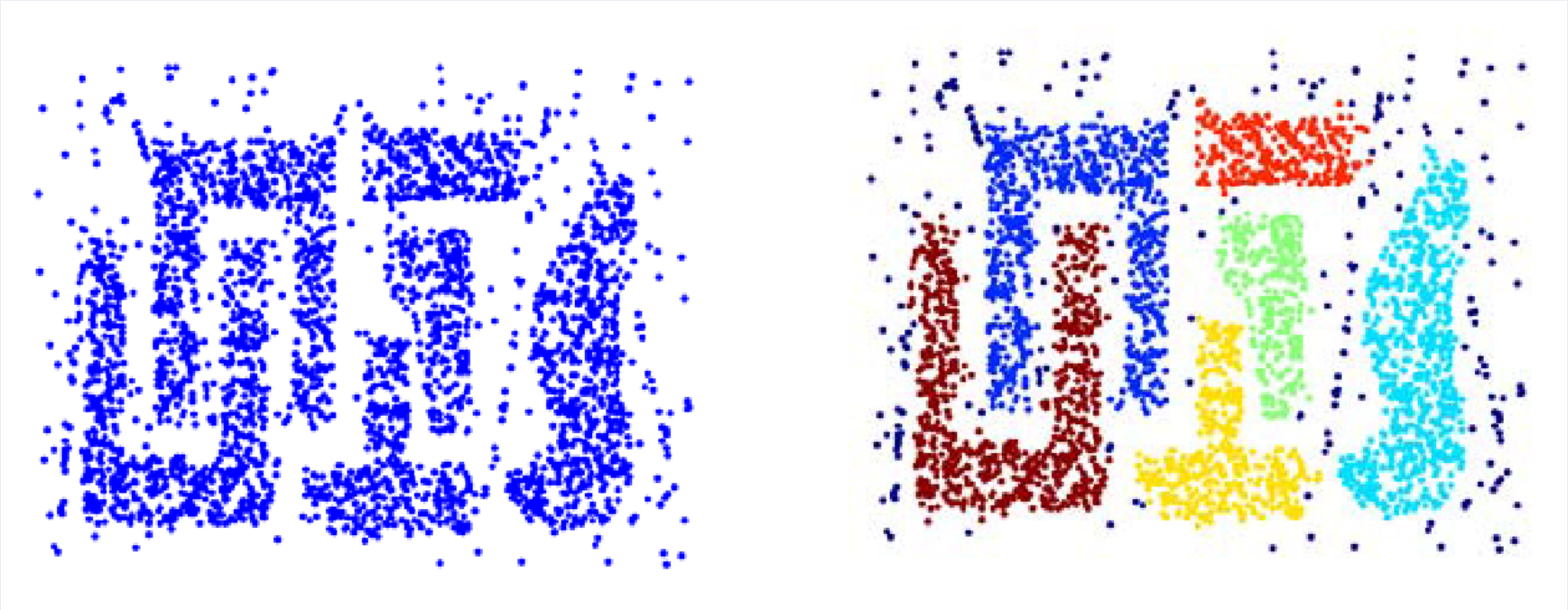


FIGURE 13: DBSCAN : points d'origine et les 6 clusters

• DBSCAN est résistant aux bruits et au problème de *taille* et de *forme* d'origine.

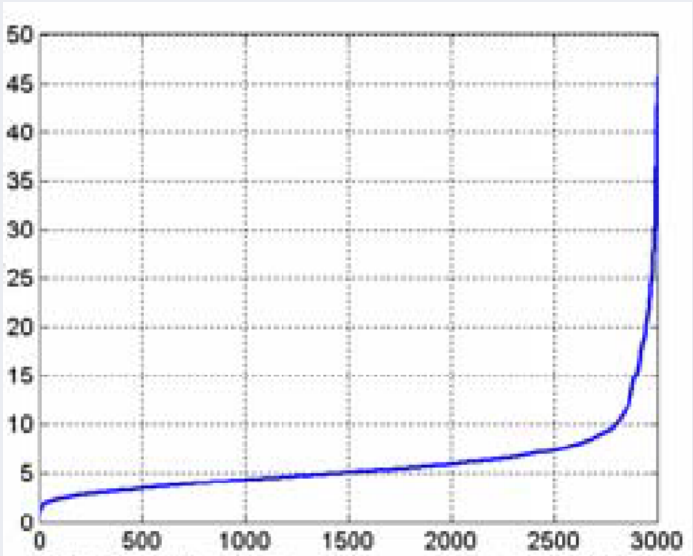
DBSCAN (suite)

Comment déterminer Eps (rayon core) et MinPts ?

• Idée : pour toute instance, les k plus proches voisins sont en gros à la même distance.
• Les points bruits ont leurs $k$ème plus proches voisins à une distance plus grande

Donc : déterminer les distances de chaque instance à son $k$ème plus proches voisins
→ k donnera une idée de **MinPts** ainsi que du **Eps**

➡ La figure pour $k = 4$



Axe X : instances **triées** selon la distance au 4^e PPV

Axe Y : Distance au 4^e PPV

DBSCAN (suite)

Complément : DBSCAN et le problème des "ponts"

- Supposons que la distance entre deux points voisins est plus petit que (ϵ).
- Les points noirs (au milieu) sont non-denses et n'appartiennent à aucun cluster (pour un *MinPts* élevé)
- Le point rouge (au milieu) le plus à droite est non-dense.
- Le point bleu (au mileu) le plus à gauche est non-dense.

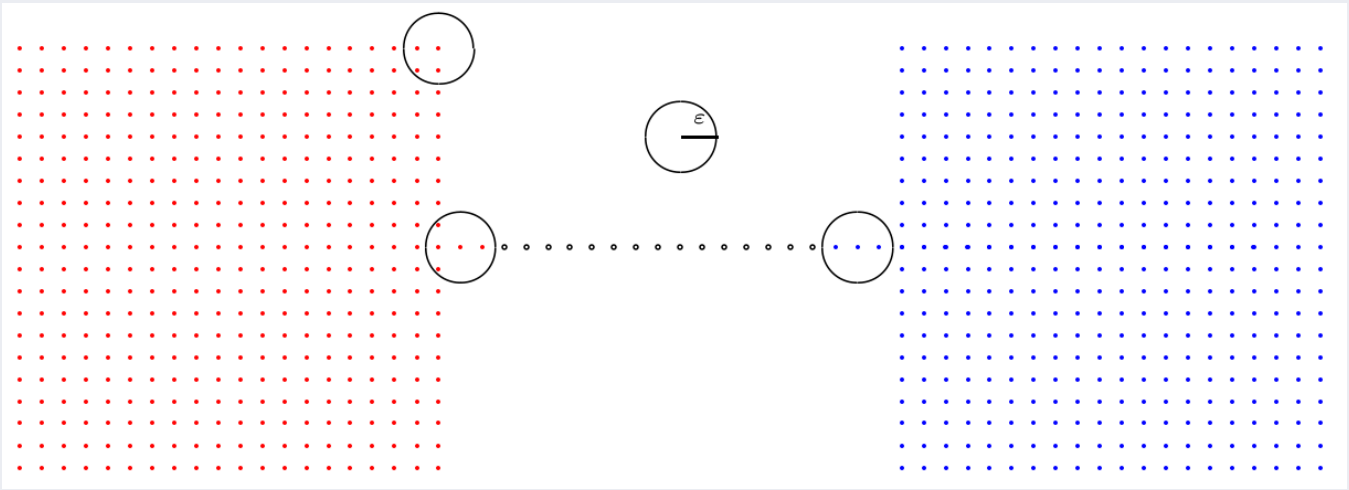


FIGURE 14: Problème de "pont" en DBSCAN (ϵ)

Clustering et les Algorithmes Génétiques

But : partitionner $S = \{o_1, ..., o_n\}$ en k clusters $C_1, C_2, ..., C_k$.

Dans cette technique Génétique, une population = un clustering.

Une introduction :

- Soit $K = \{1, 2, ..., k\}$ l'ensemble des clusters.
- Un élément $< i_1, i_2, ..., i_n > \in K^n$ représente un clustering où les instances $o_1 \in C_{i_1}, o_2 \in C_{i_2}, ..., o_n \in C_{i_n}$.

  ➙ Cette séquence décrit les clusters des instances de S.

- Notez que chaque permutation de $1, 2, ..., k$ donne le même clustering.
- Chaque élément (clustering) de K^n est un individu (ou *chromosome*).
- *L'aptitude* d'un individu pourrait être l'inverse de la dispersion intra-cluster.
- Une population est un ensemble d'individus.

Clustering et les Algorithmes Génétiques (suite)

Sur la construction d'un clustering :

- Le clustering initial peut être obtenu par exemple via *K-Means*,
- Puis on retient le meilleur k (p.ex. meilleure *vraisemblance* / SSE / ...).

Ajustement des Clusters (initial ou pour la sélection) :

- Dans un clustering obtenu, on cherche pour chaque cluster C les instances N_C qui divergent de leur cluster respectif :

  → ceux qui empirent l'erreur SSE, diminuent la vraisemblance, ...

  - Une valeur courante de la divergence : 25%
  - Si N_C est unique, on lui associe un cluster
  - Sinon, on refait un sous clustering (au pire, N_C sous-clusters)

- Les techniques génétiques peuvent s'appliquer également dans les méthodes *Agglomératives* et *Divisives*.

- On peut également procéder à un clustering Génétique dans un cadre de méta-méthode *Ensemble*.

Clustering et les Algorithmes Génétiques (suite)

Les opérateurs : *Croisement*, *Mutation* et *Sélection*

• Le **cross-over** de $< i_1, i_2, ..., i_n >$ et de $< j_1, j_2, ..., j_n >$ entre les positions l et $l+1$ donne $< i_1, i_2, ...i_l, j_{l+1}, ...j_n >$ et $< j_1, j_2, ..., j_l, i_{l+1}, ..., i_n >$.

• La **mutation** change une valeur dans un individu (une instance change de cluster) de manière arbitraire.

• Le principe de **survie des meilleures correspondances** (*survival of the fittest*) sélectionne une nouvelle population avec le même nombre d'individus.

• La probabilité pour un individu de se retrouver (une ou plusieurs fois) dans la prochaine génération est proportionnelle à son aptitude (*roue de la fortune*).

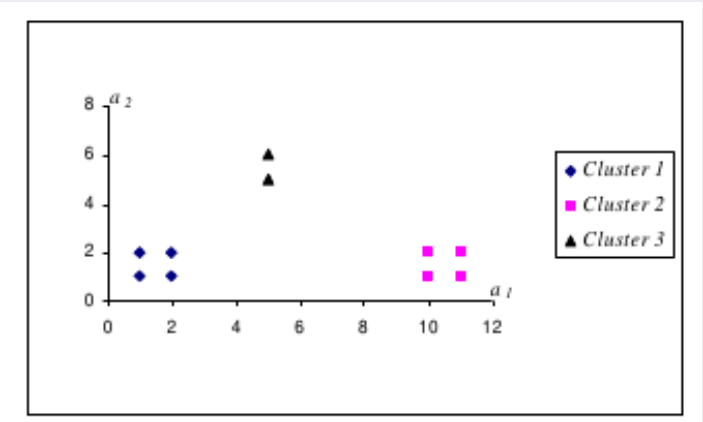
L'algorithme de principe :

+ Fixer la population de départ
+ Répéter
  - Appliquer cross-over et mutation
  - Sélectionner les individus de la prochaine génération (survival of the fittest)
+ Jusqu'à stabilisation (ou seuil erreur acceptable)

Clustering et les Algorithmes Génétiques (suite)

Un exemple : soit les 10 instances réparties en 3 clusters :

Object ($\mathbf{x}_i$)	a_1	a_2	Cluster - $\mathbf{C}_j$
$\mathbf{x}_1$	1	1	Cluster 1 ($\mathbf{C}_1$)
$\mathbf{x}_2$	1	2	Cluster 1 ($\mathbf{C}_1$)
$\mathbf{x}_3$	2	1	Cluster 1 ($\mathbf{C}_1$)
$\mathbf{x}_4$	2	2	Cluster 1 ($\mathbf{C}_1$)
$\mathbf{x}_5$	10	1	Cluster 2 ($\mathbf{C}_2$)
$\mathbf{x}_6$	10	2	Cluster 2 ($\mathbf{C}_2$)
$\mathbf{x}_7$	11	1	Cluster 2 ($\mathbf{C}_2$)
$\mathbf{x}_8$	11	2	Cluster 2 ($\mathbf{C}_2$)
$\mathbf{x}_9$	5	5	Cluster 3 ($\mathbf{C}_3$)
$\mathbf{x}_{10}$	5	6	Cluster 3 ($\mathbf{C}_3$)



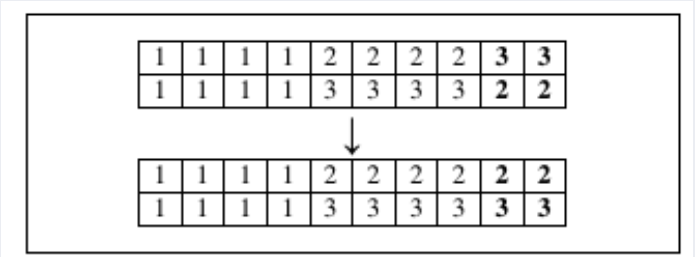
- Deux représentations équivalentes et un exemple de cross-over appliqué :

1	1	1	1	0	0	0	0	0	0
0	0	0	0	1	1	1	1	0	0
0	0	0	0	0	0	0	0	1	1

Les 10 instances et leur Cluster

Un bit en position 1..3 pour $C_1..C_3$

(Une représentation sur 2 bits est possible)



Application de l'opérateur cross-over

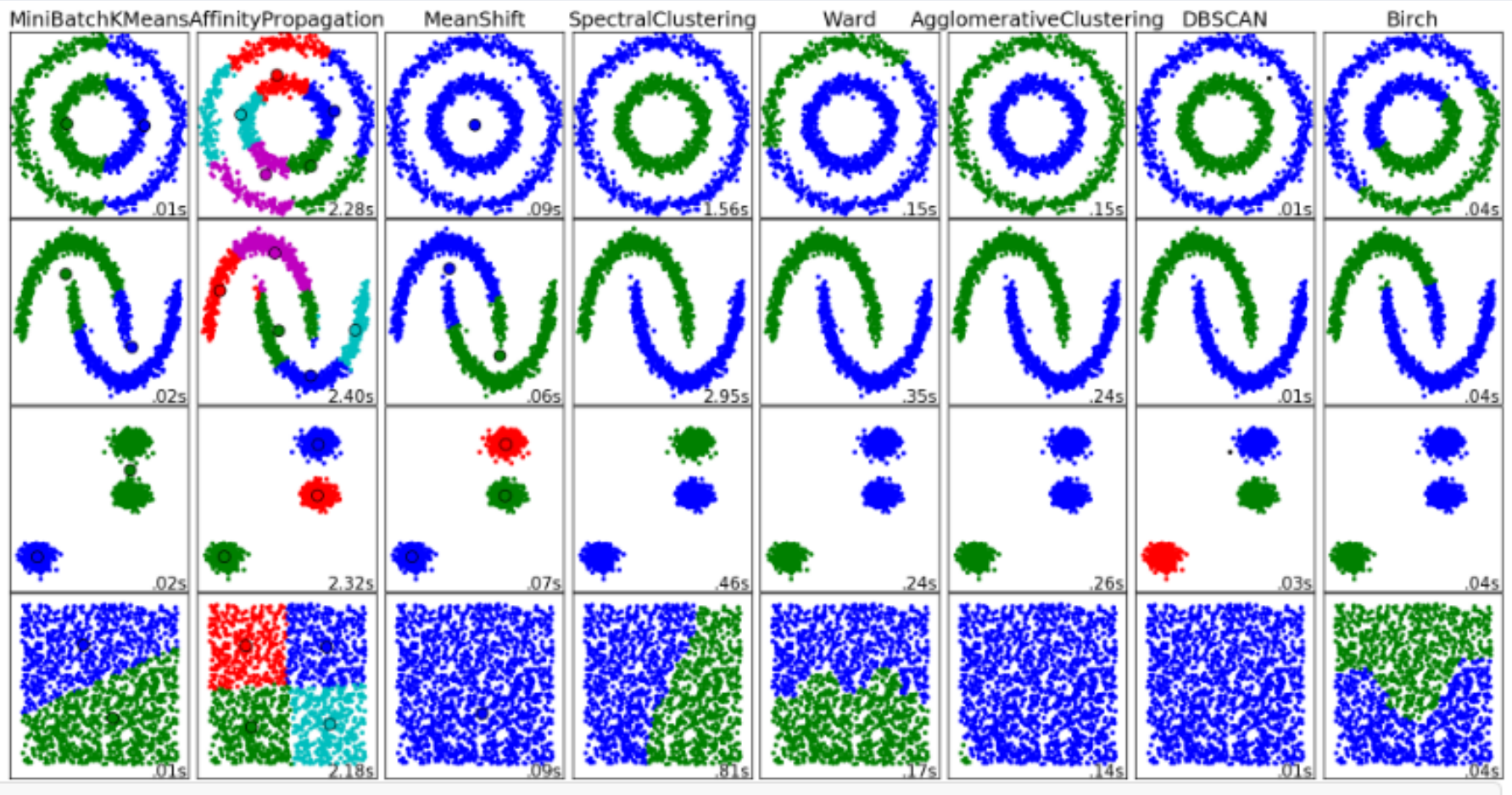
(Clusters 1..3 pour plus de clarté)

Comparaison de quelques algorithmes

Quelques algorithmes (Thanks to *scikit-learn*)

☞ Le dernier B.D. est homogène (un seul cluster) !

★ ?? ★★ ★★★



Comparaison de quelques algorithmes (suite)

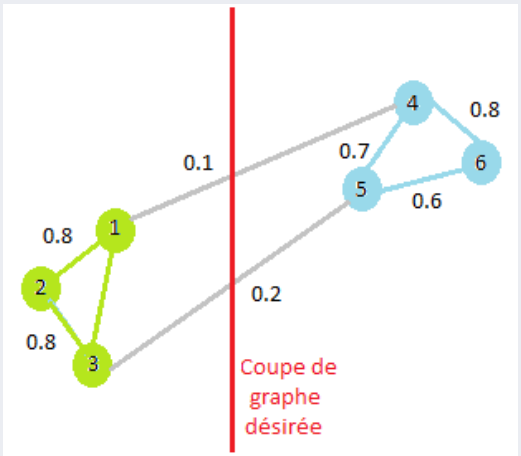
A propos : Clustering Spectral (Fig. Thanks to Wiki) :

• Le partitionnement spectral utilise le **spectre** de la matrice de similarités pour un partitionnement dans un espace de dimension inférieure.

• C'est un problème de partitionnement de graphe.

➢ Spectre de la matrice des similarité = les vecteurs propres (pour faire simple !)

➢ On regroupe ensuite ces vecteurs selon leur similarité (e.f. dist. cosin)



• On procède à une "coupe" (+ évaluation par mesures internes / externes)

◦ Similaire à un clustering Hiérarchique Divisif où les pondérations étaient / peuvent être par rapport à un centroïde / médoïde.

Évaluation du Clustering

- Rappelons qu'un clustering est presque une question de goût !
  ➙ (c'est dans l'oeil de celui qui regarde !)

- Pour la validation d'un clustering, on doit évaluer la qualité des clusters (et donc le modèle).

- Besoin d'évaluer :
  - Pour éviter de chercher des motifs dans les données avec bruits
  - Pour comparer les algorithmes de Clustering
  - Pour comparer les ensembles de clusters produits (2 modèles différents)
  - Pour comparer 2 clusters (même modèle)

☞ La validation du Clustering est une tâche TRES difficile.

Évaluation du Clustering (suite)

Exemple de 3 modèles à comparer :

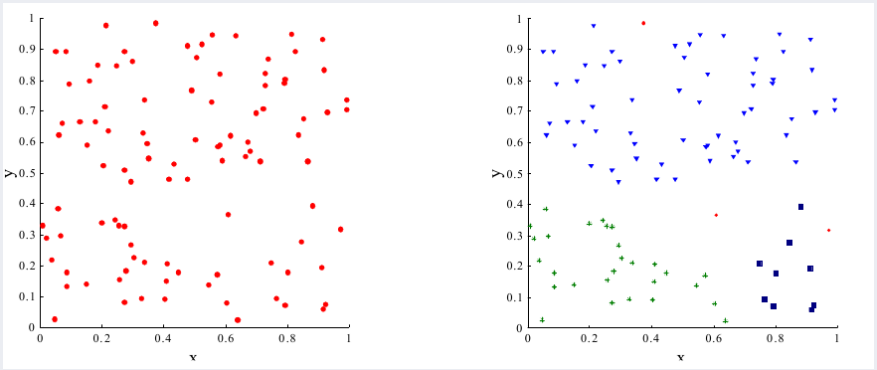


FIGURE 15: Points aléatroires d'origine et le Modèle DBSCAN

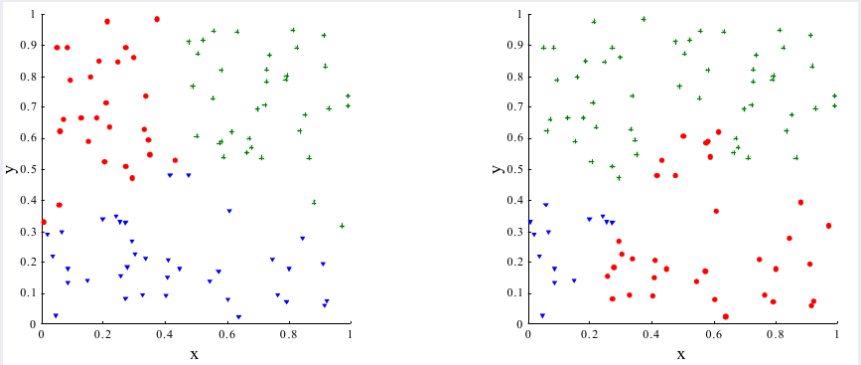


FIGURE 16: Modèle K-means et le Modèle Agglomérative (Moyenne distances)

Évaluation du Clustering (suite)

Objectifs et Démarche de validation

1. Déterminer la tendance (des regroupements) dans une BD :
   ➛ distinguer si des structures non aléatoires existent.
2. Comparer des résultats du clustering aux résultats déjà connus
   ➛ E.g. à une autre classification connue
3. Évaluer comment un Clustering représente les données (sans autres information)
4. Comparer deux Clusterings pour choisir le meilleur
5. Déterminer le nombre correct de Clusters :
   ➛ A partir de k=2,3 ou 4, décider si l'on traite toute la BD, une partie de la BD ou juste quelques Clusters

Mesures de validation de Clusters

- **Mesures numériques** pour juger des résultats d'un Clustering :
  - **Indice Externe** : comment les clusters (*Labels* donnés aux instances) trouvés correspondent à une (autre) classification des Données
    ➛ *Entropie*
  - **Indice Interne** : mesure de la qualité du Clustering sans information externe
    ➛ SSE (somme des erreurs au carré)
  - **Indice Relatif** : pour comparer deux Clustering différents (ou 2 Clusters)
    ➛ On utilise souvent un indice externe/interne (SSE ou Entropie)
- Parfois, ces *Indices* sont appelés des **critères**.

Indice Externe : Corrélation

- Deux métriques utilisée pour mesurer la corrélation :
  - Matrice de **proximité**
  - Matrice d'**incidence** : matrice $n \times n$ pour n instances :
    - ➛ l'entrée =1 si le couple d'instances associée est dans le même cluster
    - ➛ l'entrée =0 si le couple d'instances associée est dans 2 clusters différents

Une mesure de validation :

- Calculer la corrélation entre les deux matrices
  - ➛ Elles sont symétriques : il suffit de calculer la corrélation enter $n(n-1)/2$ entrées.
- Une corrélation élevée veut dire :
  - ➛ les points appartenant au même Cluster sont proches.

☞ mesure **mal adaptée** pour des clusterings basés sur la densité ou la contigüité.

Indice Externe : Corrélation (suite)

Exemple de mesure de Corrélation :

Corrélation entre la matrice d'Incidence et de Proximité.
➛ Pour K-means appliqué à 2 exemples de nuages :

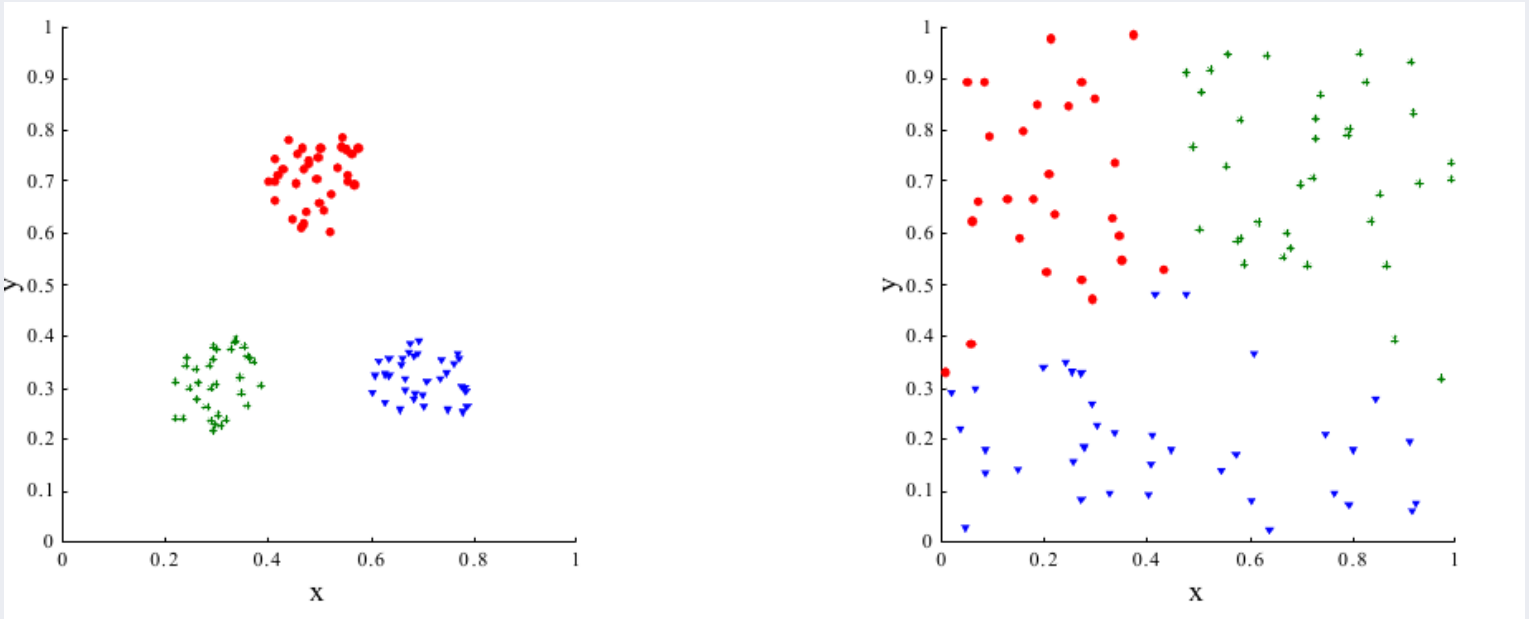


FIGURE 17: Corrélation élevée

Corrélation basse (plus dispersés)

Mesure Externe : Matrice de proximité

Mesure de Validation avec seule la matrice de Proximité

- On ordonne la matrice selon les labels des clusters (avant la visualisation).

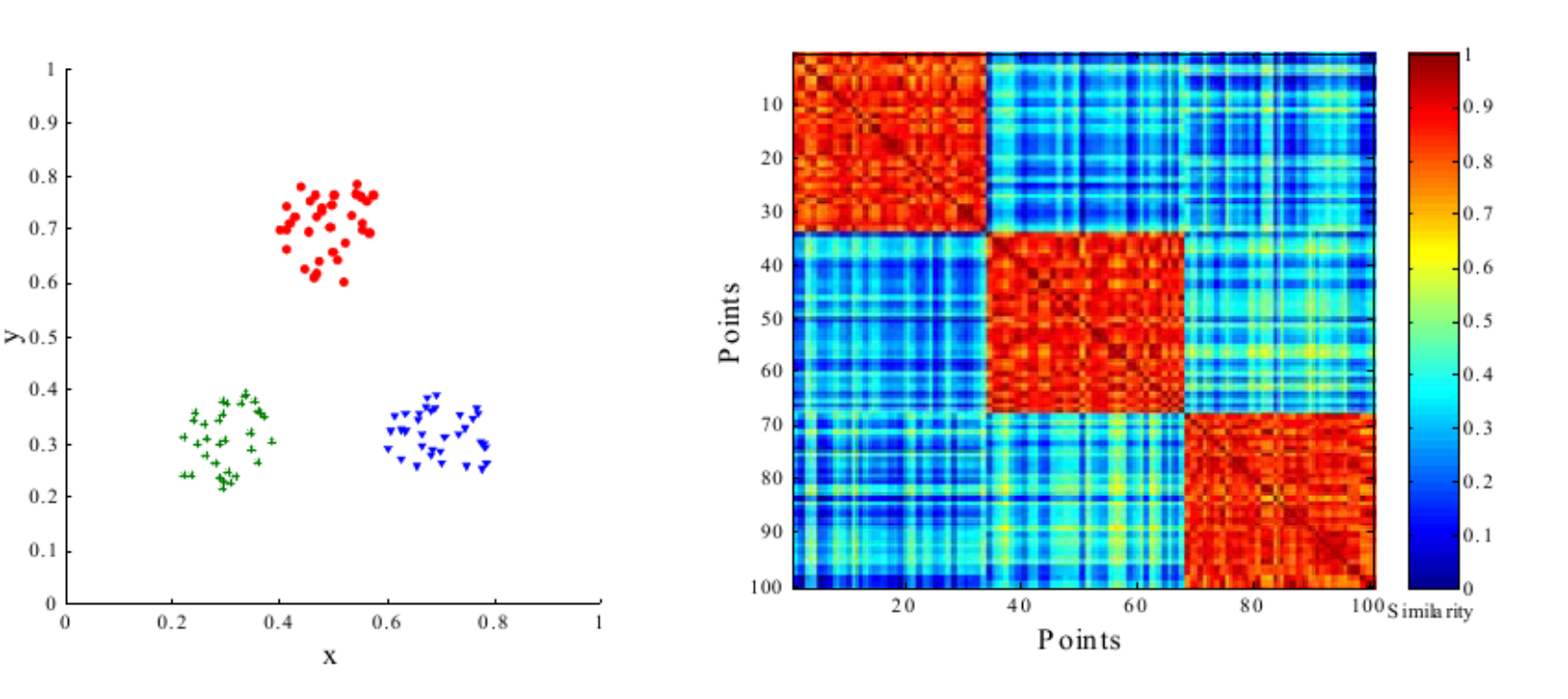


FIGURE 18: Visualisation du Clustering (clusters et carrés de la matrice de similarité **denses**)

Mesure Externe : Matrice de proximité (suite)

Exemple 2 de matrice de Proximité (seule) :

- Un exemple sur des données aléatoires (pas trop dense)

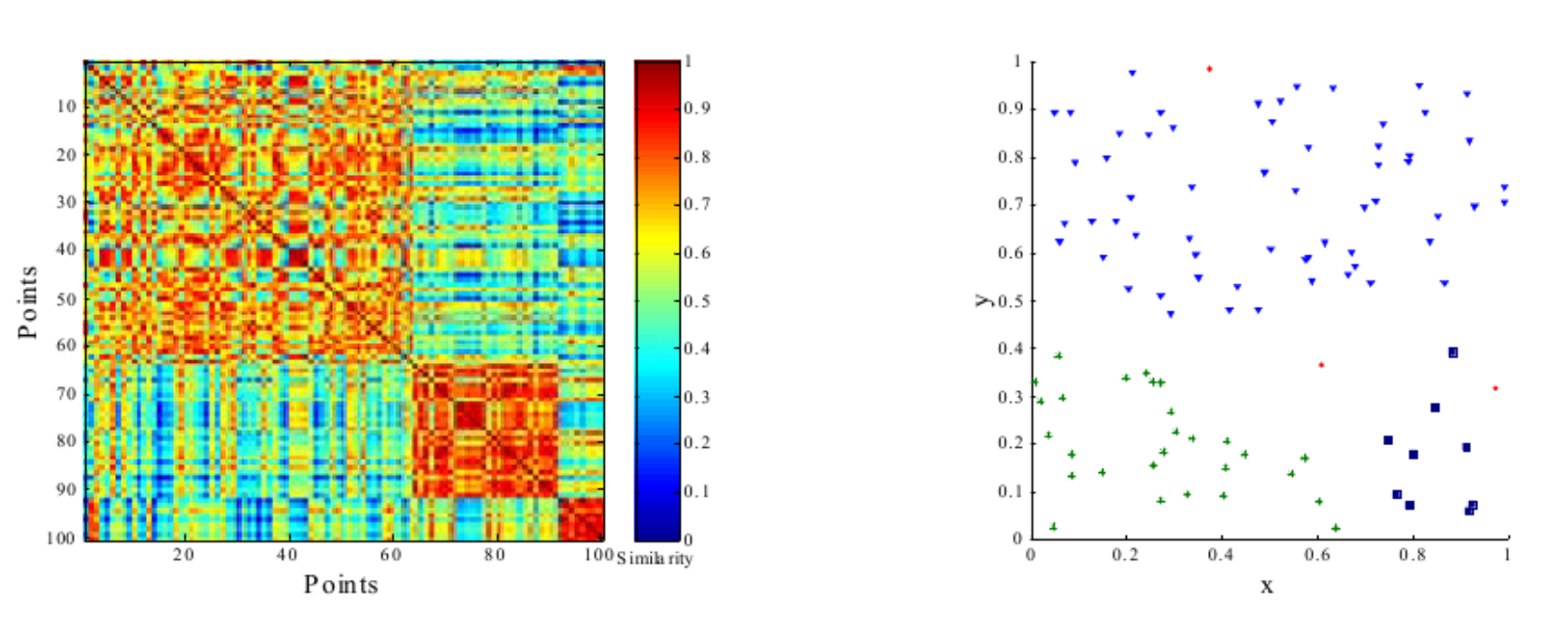


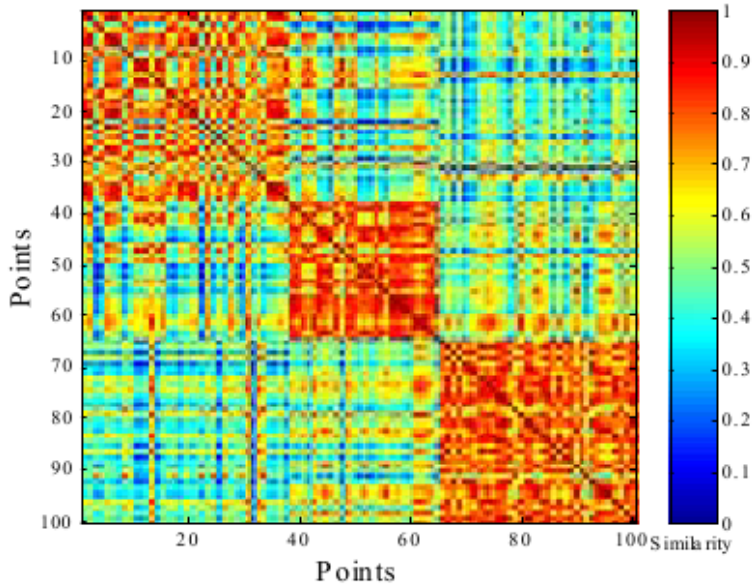
FIGURE 19: Visualisation du Clustering par **DBSCAN** (matrice de similarité)

- On note le manque de densité dans les carrés.

Mesure Externe : Matrice de proximité (suite)

Exemple 3 de matrice de Proximité :

- Même exemple sur des données aléatoires (pas trop dense) avec **K-means**



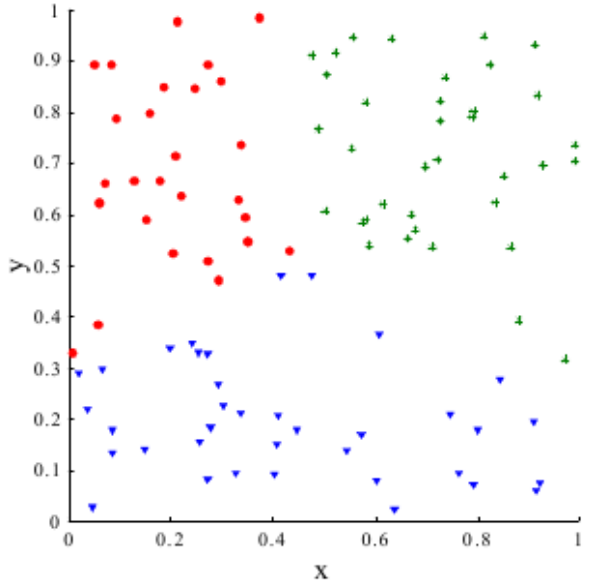


FIGURE 20: Visualisation du Clustering par **K-Means** (matrice de similarité)

Mesure Externe : Matrice de proximité (suite)

Exemple 4 de matrice de Proximité :

• Même exemple sur des données aléatoires avec **Agglomérative Link complet** (moyennes)

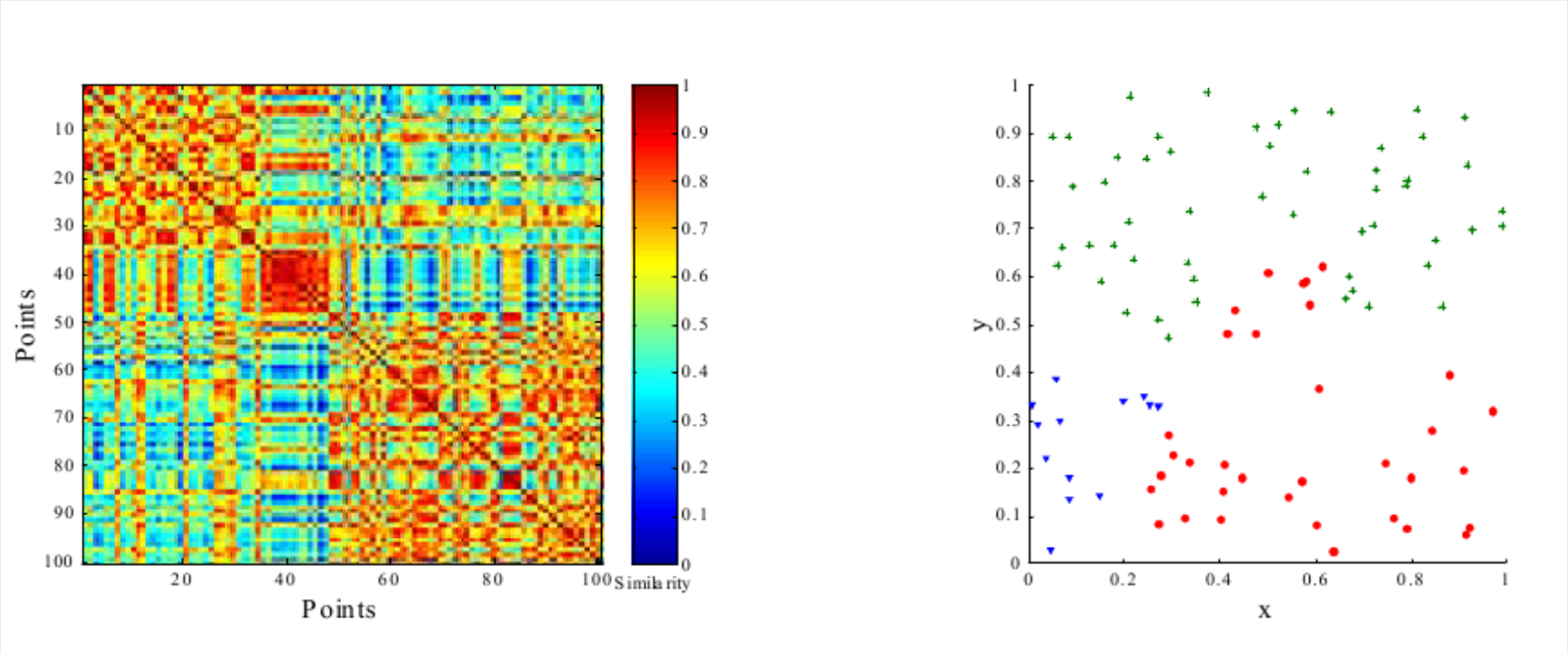


FIGURE 21: Visualisation du Clustering par **Agglomérative Link complet** (matrice de similarité)

Mesure Externe : Matrice de proximité (suite)

Exemple 5 de matrice de Proximité :

- Un autre exemple (avec **DBSCAN**) dense.

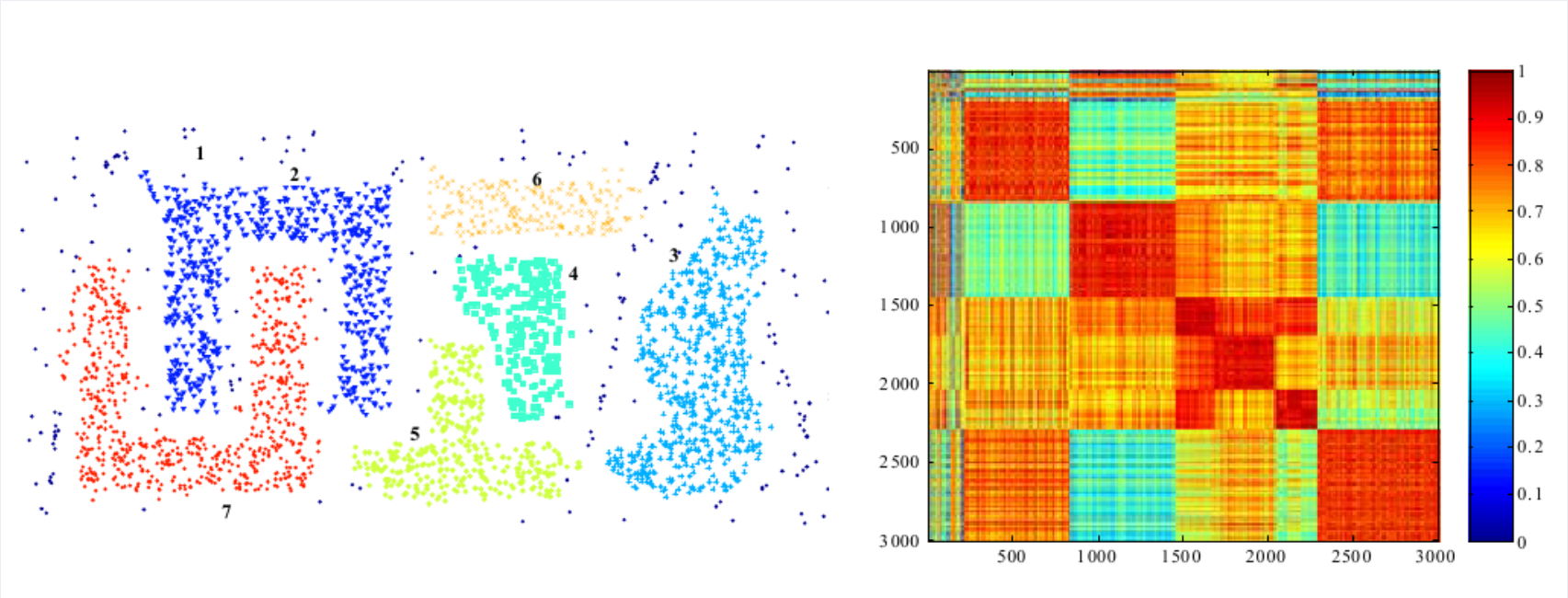
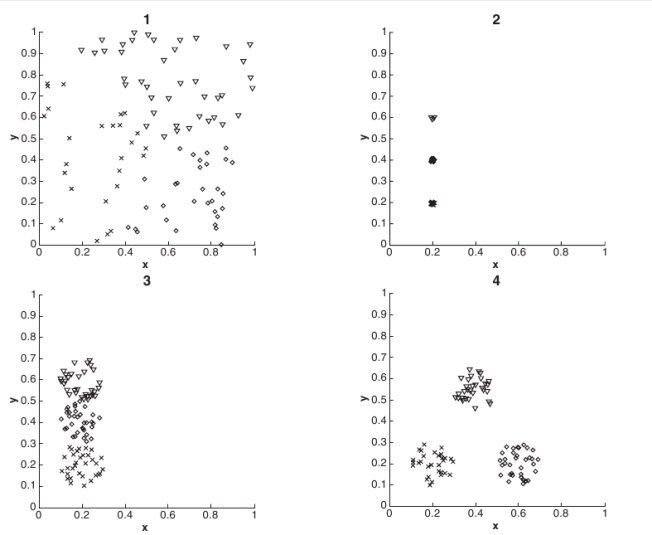
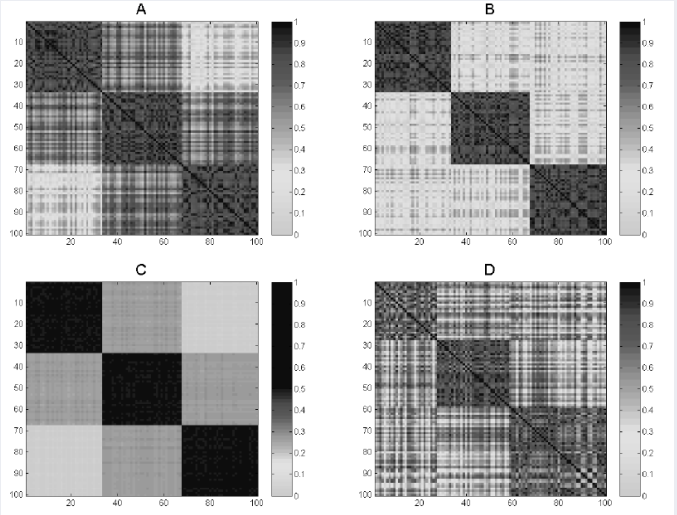


FIGURE 22: Visualisation du Clustering par **DBSCAN** (matrice de similarité)

Mesure Externe : Matrice de proximité (suite)

Exemple : correspondance des nuages aux densités (des matrices de proximités triées par cluster)





Réponse : 1 à D, 2 à C, 3 à A, 4 à B.

Indice Interne : SSE

- Dans certains cas plus compliqués, les clusters ne sont pas bien séparés
- On utilise un *indice Interne* pour mesurer la qualité du Clustering sans information externe :
  - SSE (moyenne) : bonne mesure pour comparer 2 Clusterings ou 2 Clusters
  - Peut aussi être utile pour estimer le nombre de Clusters

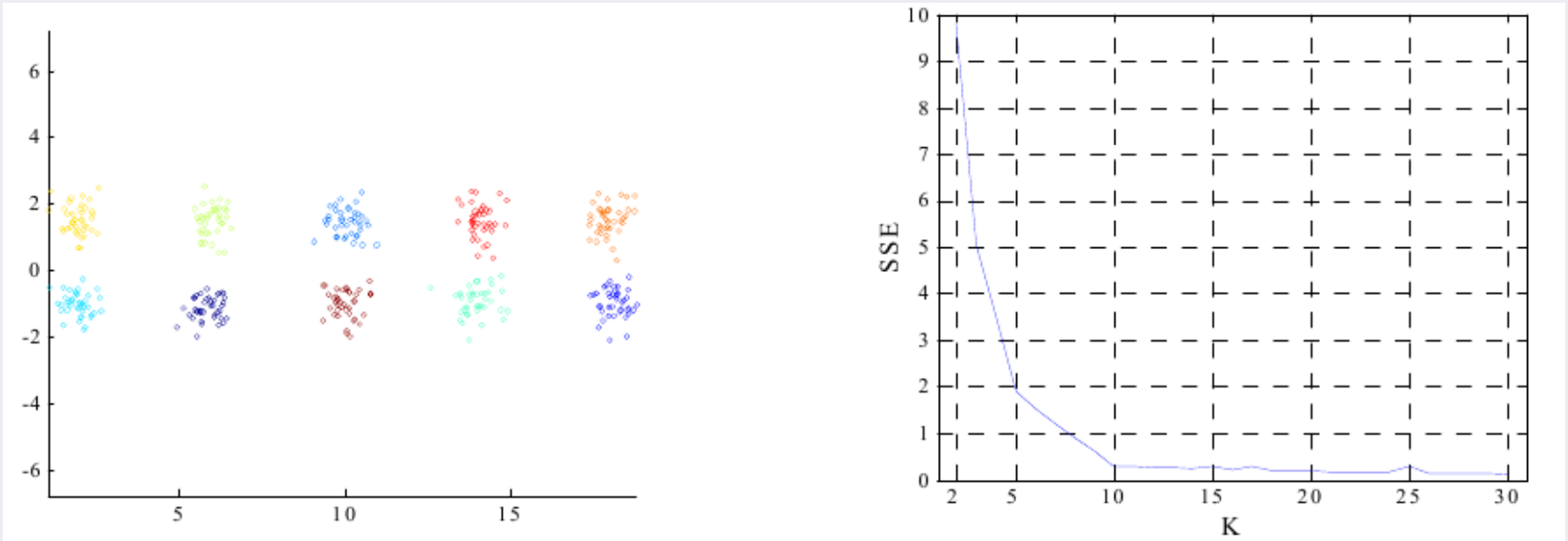


FIGURE 23: Validation du Clustering : SSE vs. K

Indice Interne : SSE (suite)

Exemple de Validation Interne par SSE :

Un cas plus compliqué avec K-means

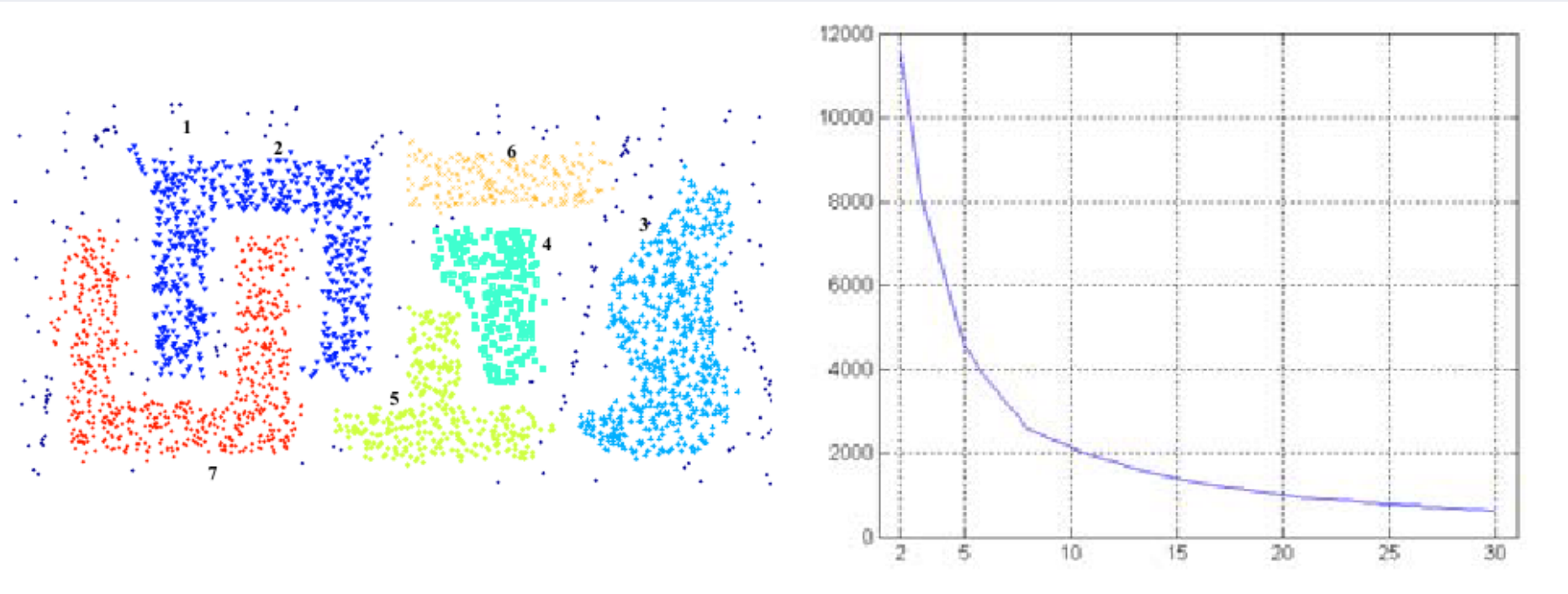


FIGURE 24: SSE pour K-means (les mêmes axes que le cas précédent)

Mesures Internes Statistiques

- En complément des autres mesures.
- Mesures de **Cohésion** et de **Séparation** de Clusters.
- **Cohésion** : mesure la proximité des instances dans un Cluster (cf. SSE)
- **Séparation** : mesure la qualité de séparation (distinction) d'un cluster des autres.

Définitions (erreur au carré) :

➛ La Cohésion mesurée à l'aide du SSE intra-clusters (W : Within) :

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

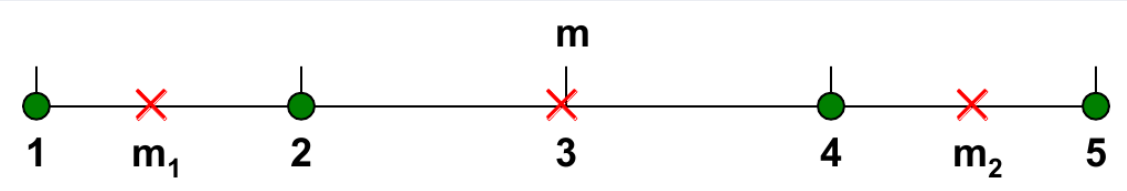
➛ La Séparation mesurée à l'aide du SSE inter-clusters (B : Between) :

$$BSS = \sum_i |C_i|(m - m_i)^2 \qquad |C_i| = taille\ Cluster\ i$$

Mesures Internes Statistiques (suite)

Cohésion et Séparation (suite) :

Exemple : $BSS + WSS =$ constante pour 2 clustering différentes



K=1 cluster : $WSS = (1-3)^2 + (2-3)^2 + (4-3)^2 + (5-3)^2 = 10$

$BSS = 4 \times (3-3)^2 = 0$

$Total = 10 + 0 = 10$

K=2 clusters : $WSS = (1-1.5)^2 + (2-1.5)^2 + (4-4.5)^2 + (5-4.5)^2 = 1$

$BSS = 2 \times (3-1.5)^2 + 2 \times (4.5-3)^2 = 9$

$Total = 1 + 9 = 10$

Mesures Internes Statistiques (suite)

Cohésion et Séparation (suite) :

• On peut aussi utiliser le graphe de proximité pour mesurer la Cohésion et la Séparation.

- La *Cohésion* (WSS, interne) est la somme des poids donnés par les distances dans un cluster
- La *Séparation* (BSS, externe) est la somme des poids donnés par les distances entre 2 noeuds (un dans chaque cluster)

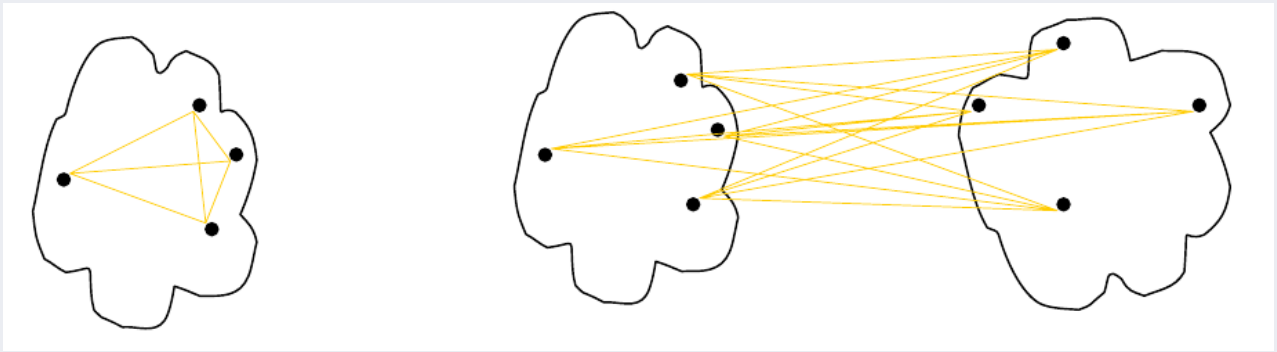


FIGURE 25: Cohésion et Séparation

Mesures Internes Statistiques (suite)

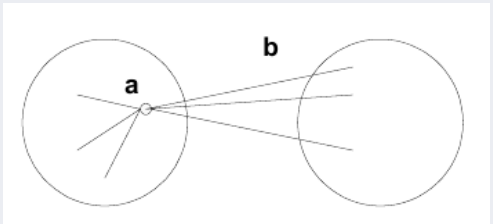
Cohésion et Séparation : Coefficient de Silhouette

- Le *Coefficient de Silhouette* combine des idées de Cohésion et de Séparation
  Mais pour des instances individuelles comme pour les Clusters.

- Pour une instance I, calculer :
  - **a** = moyenne des distances entre I et les autres instances de son cluster
  - **b** = min(moyenne des distances entre I et les instances d'un autre cluster)

  - Le **Coefficient de Silhouette** pour I est :
    $s = 1 - \frac{a}{b}$ si $a < b$ (ou plus rarement $s = -\frac{b}{a} + 1$ si $a \geq b$)

Typiquement entre 0 ou 1
Mieux vaut être proche de 1



- On peut calculer la **Silhouette Moyenne** pour un cluster ou pour un Clustering.

Qulaité par l'Entropie et la Pureté

- L'**Entropie** et la **Pureté** permettent de mesurer la cohésion intra-clusters.
- On peut utiliser une classification (info externe) en phase Apprentissage.

- **Entropie** : pour chaque cluster, on calcule la distribution des données par :
  - Pour le cluster C_j, calculer $P_{ij} = \frac{n_{ij}}{n_j}$ la probabilité pour une instance du cluster C_j d'appartenir à la classe i (n_{ij} = nbr d'instance de classe i dans C_j, $n_j = |C_j|$)

  - Calculer l'Entropie de C_j par $h(C_j) = \sum_{1}^{L} P_{ij} \times log(P_{ij})$ (L : nbr de classes).

  - Calculer l'entropie totale d'un ensemble de k clusters $H = \sum_{1}^{k} \frac{n_j}{n} h(C_j)$

- **Pureté (purity)** : pour un cluster C_j, $pureté(C_j) = max(P_{ij})$ et

$$\text{la pureté totale du Clustering} = \sum_{1}^{k} \frac{n_j}{n}\ pureté(C_j)$$

Qulaité par l'Entropie et la Pureté (suite)

Un exemple : une base documentaire

Cluster	Entertainment	Financial	Foreign	Metro	National	Sports	Entropy	Purity
1	3	5	40	506	96	27	1.2270	0.7474
2	4	7	280	29	39	2	1.1472	0.7756
3	1	1	1	7	4	671	0.1813	0.9796
4	10	162	3	119	73	2	1.7487	0.4390
5	331	22	5	70	13	23	1.3976	0.7134
6	5	358	12	212	48	13	1.5523	0.5525
Total	354	555	341	943	273	738	1.1450	0.7203

FIGURE 26: K-mean : mesure de qualité par l'entropie et la pureté (purity) sur une BD. Documentaire

- Rapport inverse Entropie (incertitude) et la pureté.
- La classe de chaque document est connue,
- 6 clusters qui mélangent différentes classes.
- **Le cluster 3** semble le plus intéressant (beaucoup de sport, peu des autres).