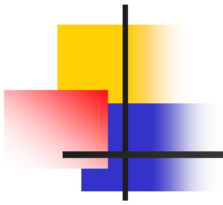
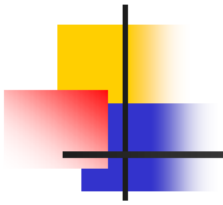


INF5081

Apprentissage non supervisé
Clustering

bouguessa.mohamed@uqam.ca



Plan

1 – Mise en contexte

2 – Clustering avec *K*-means

3 – Clustering hiérarchique

4 – Clustering basé sur la densité

5 – Clustering des graphes

Mise en contexte

- Jusqu'ici, il n'a été question dans ce cours que de l'apprentissage supervisé : les données ont toujours été pourvues d'une étiquette ou d'une valeur numérique fournie par un oracle (expert).
- Dans ce chapitre, nous nous plaçons en dehors de cette hypothèse, afin d'aborder la problématique de l'apprentissage non supervisé (le forage de données descriptif).
- Spécifiquement, nous nous focalisons sur les techniques de classification non supervisée ou le clustering.

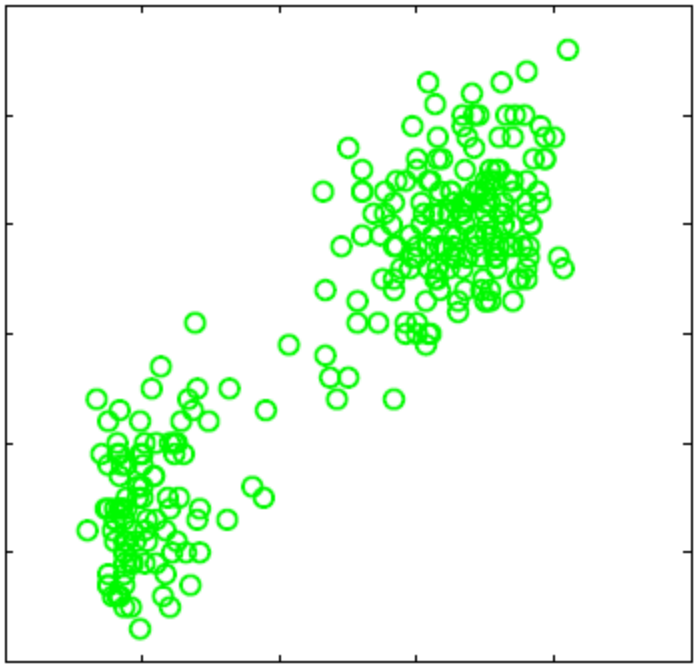
Clustering?

- Le processus du clustering vise à construire des groupes (clusters) d'objets similaires à partir d'un ensemble hétérogène d'objets. Chaque cluster issu de ce processus doit vérifier les deux propriétés suivantes :

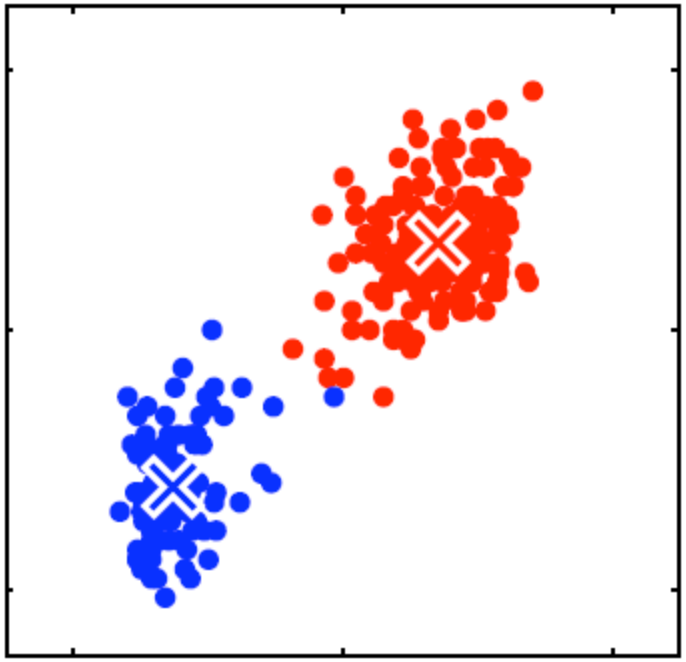
1. La cohésion interne (les objets appartenant à ce cluster soient les plus similaires possibles).
2. L'isolation externe (les objets appartenant aux autres clusters soient les plus distincts possibles).

- Le processus de clustering repose sur une mesure précise de la similarité des objets que l'on veut regrouper. Cette mesure est appelée distance ou métrique.

Exemple 1

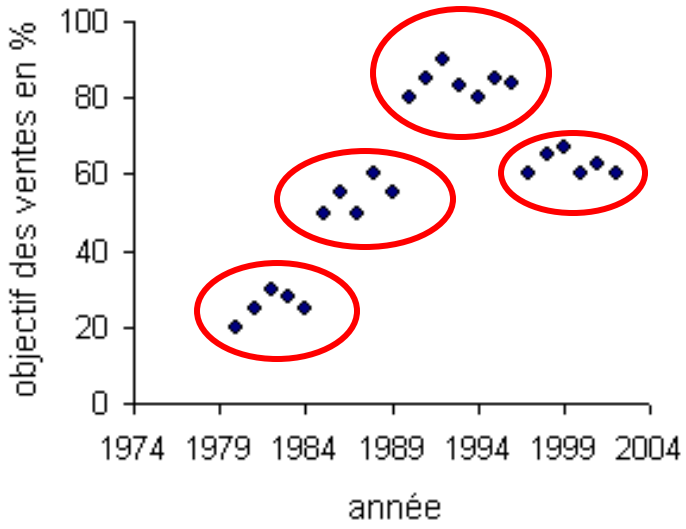


Entré : ensemble de données (non étiquetés)



Sortie : clusters identifiés

Exemple 2



- Les techniques de clustering suivent le même principe général qui consiste à maximiser la similarité des objets à l'intérieur d'un cluster, et minimiser la similarité des objets entre les clusters.

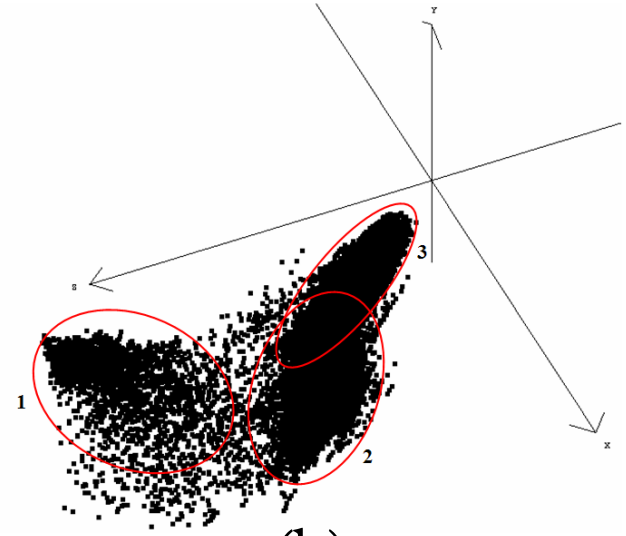
Applications

- Traitement d'image.
- Finances.
- Études démographiques.
- Recherche génétique.
- Analyse des données.
- Prospection du Web (Web Mining).

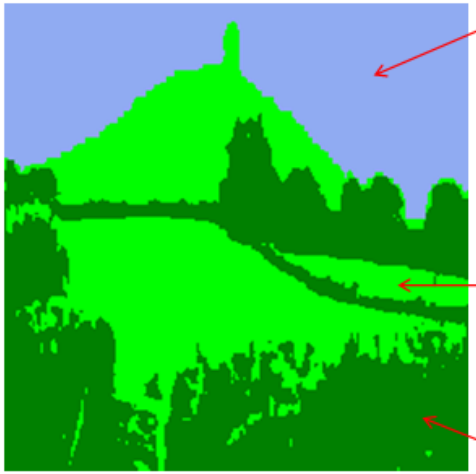
Exemple

- Segmentation d'image

(a)



(b)



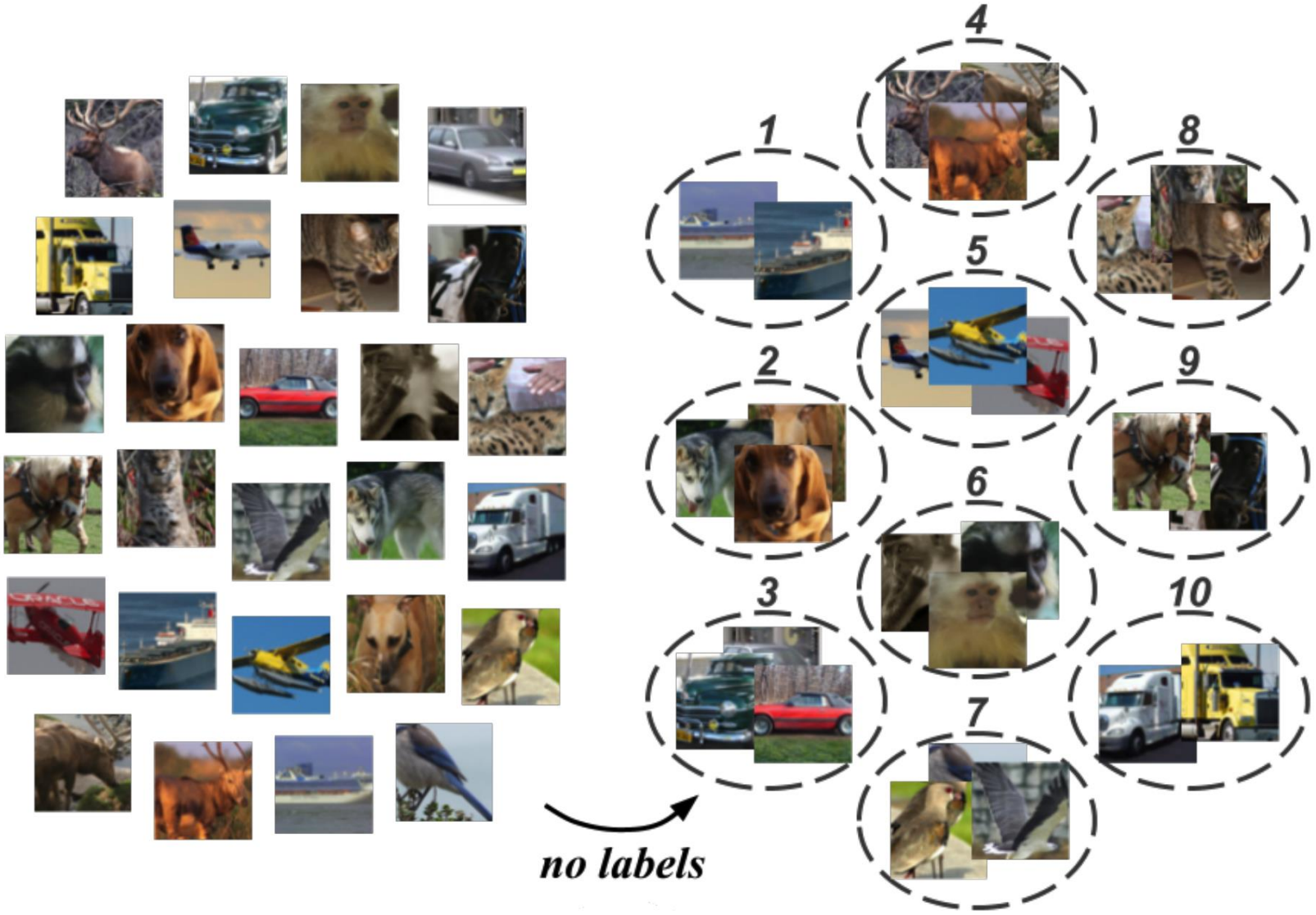
(c)

(a) L'image originale

(b) Représentation dans l'espace de couleurs 3D : Rouge-Vert-Bleu

(c) L'image segmentée

Exemple



Classification vs Clustering

- Le but principal du clustering est la découverte automatique des structures similaires dans l'espace d'objets
- La classification supervisée consiste à l'assignation d'un objet à une classe spécifique parmi un certain nombre de classes prédéfinies.
- En d'autres termes, contrairement au clustering où l'algorithme doit découvrir lui-même des groupes (clusters) d'objets, la classification supervisée suppose qu'il existe déjà une classification des données.

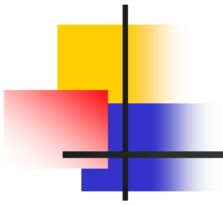
➢ Le clustering : tâche descriptive.

➢ La classification supervisée : tâche prédictive.

Algorithmes de clustering : prérequis

- Mise à l'échelle (Scalability)
- Capacité de gérer différents types d'attributs
- Identifier des clusters avec des formes différentes
- Capacité de gérer le bruits
- Capacité d'identifier des clusters à partir des données de grandes dimensions
- Besoin minimum de connaissance du domaine d'application pour déterminer les paramètres d'un algorithme

➢ Aucun algorithme de clustering ne peut considérer tous ces éléments. La majorité des techniques de clustering cible certains éléments et ignore les autres.



Plan

1 – Mise en contexte

2 – Clustering avec *K*-means

3 – Clustering hiérarchique

4 – Clustering basé sur la densité

5 – Clustering des graphes

Notations

- Soit $X = \{x_1, \ldots, x_n\}$ un ensemble d'objets à grouper (l'ensemble de données).
- Le but est de grouper les objets de l'ensemble X en K clusters, de tel que chaque objet appartient à un et un seul cluster.
- $C = \{C_1, C_2, \ldots, C_K\}$ dénote l'ensemble des K cluster identifier par l'algorithme de clustering de telle sorte que

$$1.\ \forall i, j\ C_i \bigcap C_j = \phi$$

$$2.\ \bigcup_{i=1}^{K} C_i = X$$

Notations

- Une manière pratique de décrire l'ensemble C (l'ensemble de clusters identifiés) consiste à utiliser une notation matricielle.
- Soit U la matrice caractéristique de la partition X

$$U = \begin{pmatrix} u_{11} & \cdots & u_{1K} \\ \vdots & \ddots & \vdots \\ u_{n1} & \cdots & u_{nK} \end{pmatrix}$$

→ où $u_{ij} = 1$ si et seulement si $x_i \in C_j$, et $u_{ij} = 0$ sinon.

→ Remarquons que la somme de la i^éme ligne est égale à 1 (un élément appartient à un seul cluster) et la somme des valeurs de la j^iéme colonne vaut n_j le nombre d'éléments du cluster C_j.

Procédure de clustering

- But :
  grouper les objets selon une mesure de similarité

- Processus d'identification des clusters :

  étant donné le nombre de clusters *K*, trouver une partition en *K* clusters qui **optimise un critère (ou une fonction) de partitionnement.**

Procédure de clustering

- La première chose à faire consiste à clarifier formellement le sens du mot optimal.
- La solution généralement adoptée est de choisir une mesure numérique de la qualité d'une partition.
- Cette mesure est parfois appelée critère, fonctionnelle ou bien encore fonction objective.
- Le but d'une procédure de clustering est donc de trouver la partition qui optimisent (la plus petite ou la plus grande valeur) d'une fonctionnelle bien définie.

Procédure de clustering

- Pour trouver la meilleure partition (celle qui donne la valeur optimale d'une fonction objective) nous utilisons des techniques de clustering qui convergent vers des optima locaux de cette fonctionnelle. Les partitions ainsi trouvées sont souvent satisfaisantes.

- L'algorithme ***K*-means** est un exemple concret de ces techniques

On le nome aussi : C-moyennes dures

" Hard C-Means : HCM "

➢ Le nom « hard » vient du fait que l'algorithme génère une partition « hard » là où un objet appartient à un et un seul cluster seulement.

Principe de l'algorithme *K*-means

- L'algorithme ou *K*-means partitionne l'ensemble de données *X* à un certain nombre de clusters *K* (le nombre de clusters *K* est fourni par l'utilisateur).
- Chaque cluster est représenté par son centre.
- On commence avec *K* clusters et on raffine les clusters itérativement.
- *K*-means génère une partition Hard (c.-à-d.. un point x_i appartient à un seul cluster seulement)

Fonction objective à optimiser

- HCM minimise la fonction suivante:

$$f(U,V) = \sum_{i=1}^{n} \sum_{j=1}^{K} u_{ij} \left\| x_i - \mu_j \right\|^2$$

- U : la matrice qui caractérise la partition.
  u_{ij} sont les éléments de U ($u_{ij} = 0$ ou 1 seulement)
- $V = \{\mu_1, \mu_2, \ldots, \mu_K\}$ dénote les centres de clusters

- Le but est de maximiser la similarité des objets à l'intérieur d'un cluster, et ce en minimisant la distance entre chaque point et son centre de cluster.
- On cherche à trouver des clusters denses est bien séparés.

Schéma général de *K*-means

- Pour réaliser un clustering avec HCM on doit implémenter les étapes suivantes

1. sélectionner aléatoirement un ensemble de *K* objets comme centres initiaux
2. **répéter**
   a. former *K* clusters et ce en assignant chaque point au centre le plus proche
   b. recalculer les centres de clusters
3. **jusqu'à** stabilité de la partition (les centres ne changent pas)

L'algorithme *K*-means

Entrée : $X=\{x_1, x_2, \ldots, x_n\}$ ensemble de données, K : le nombre de clusters, ε : le seuil pour la convergence de l'algorithme

Sortie : la matrice U qui indique l'appartenance d'un objet à cluster, et les centres de clusters $V=\{\mu_1, \ldots, \mu_K\}$

1 . Initialiser les centres de cluster aléatoirement μ_c^0 $(c = 1, \ldots, K)$

2 . Calculer la matrice $U_{(n*K)}$

pour $i = 1, \ldots, n$

pour $j = 1, \ldots, K$

si $dist(x_i, \mu_c) < dist(x_i, \mu_j)$ **alors** $u_{ij} = 0$; //$c = 1, \ldots, K$

sinon $u_{ij} = 1$;

3 . Calculer les nouveaux centres de clusters

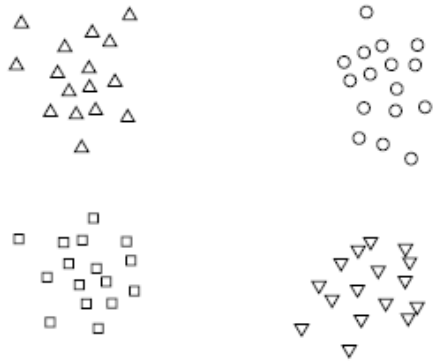
$$\mu_j^1 = \frac{\sum_{i=1}^{n}(u_{ij} \times x_i)}{\sum_{i=1}^{n} u_{ij}}; \; j = 1, \ldots, K$$

4 . **si** $\left\|\mu_c^0 - \mu_c^1\right\| < \varepsilon$ **alors** aller à 5

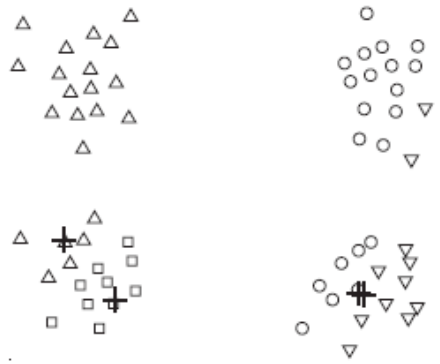
sinon $\mu_c^0 = \mu_c^1$ aller à 2

5 . Fin de l'algorithme

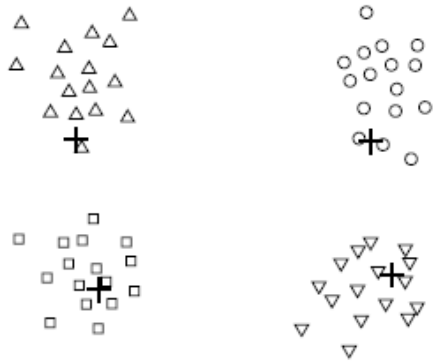
Illustration



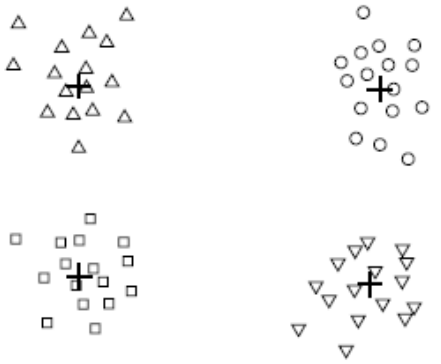
(a) Initial points.



(b) Iteration 1.

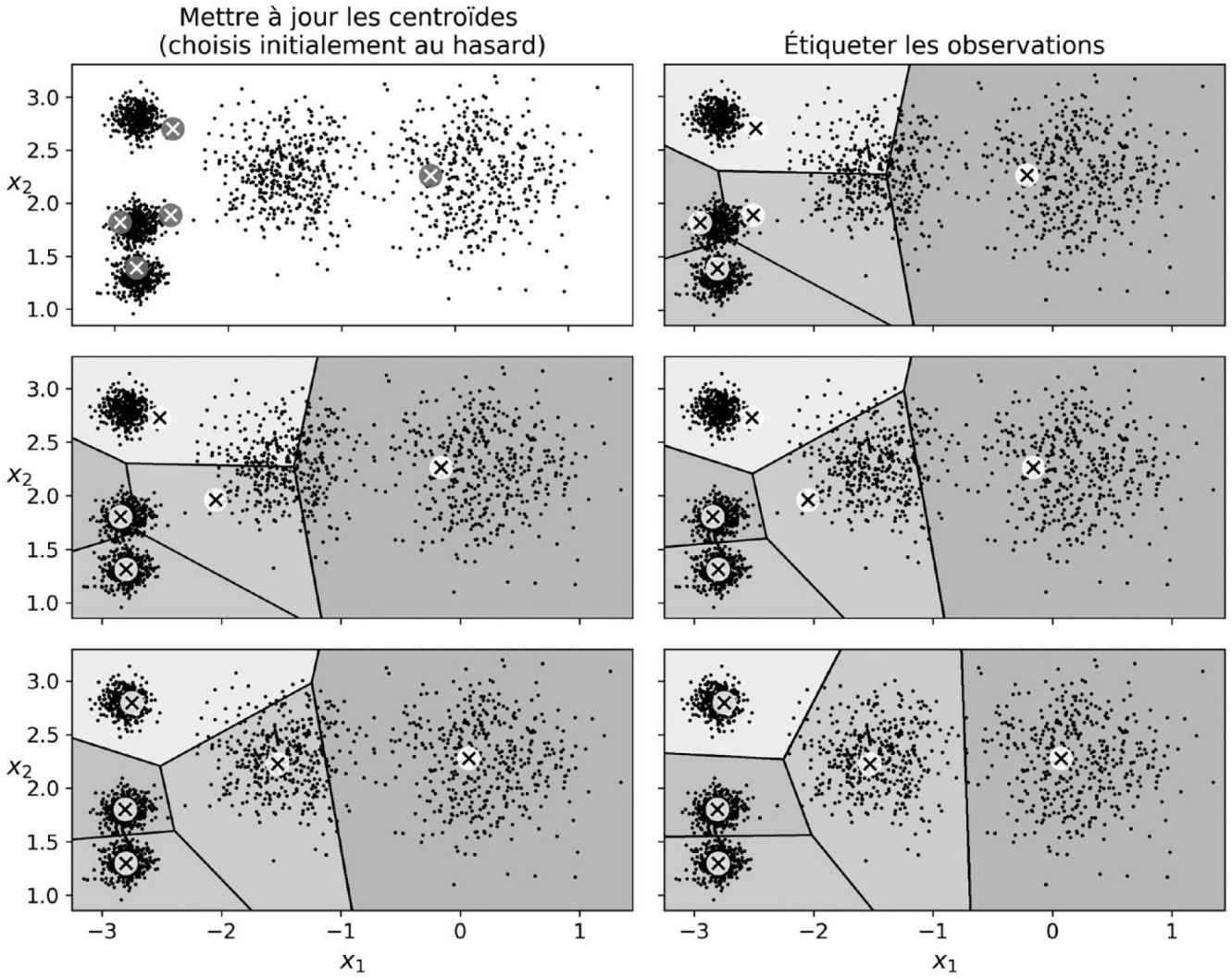


(c) Iteration 2.



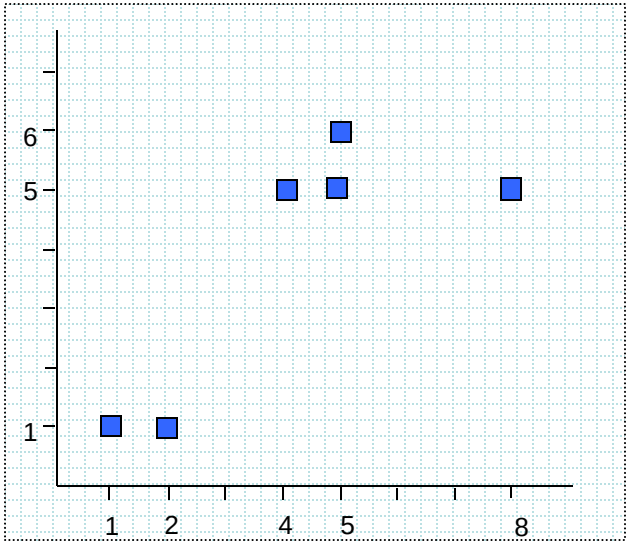
(d) Iteration 3.

Illustration

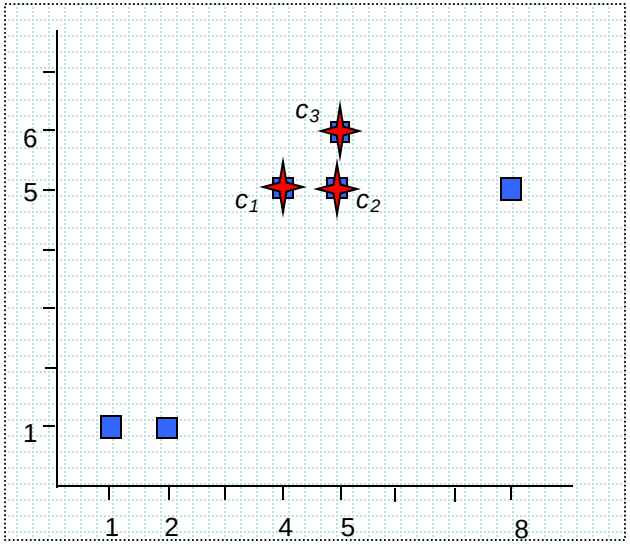


*Source : Aurélien Geron. Machine Learning avec Scikit-Learn : Mise en œuvre et cas concrets. Dunod, 2019.

Exemple



← Ensemble de 6 objets à regrouper.



← Initialisation aléatoire des centres de clusters

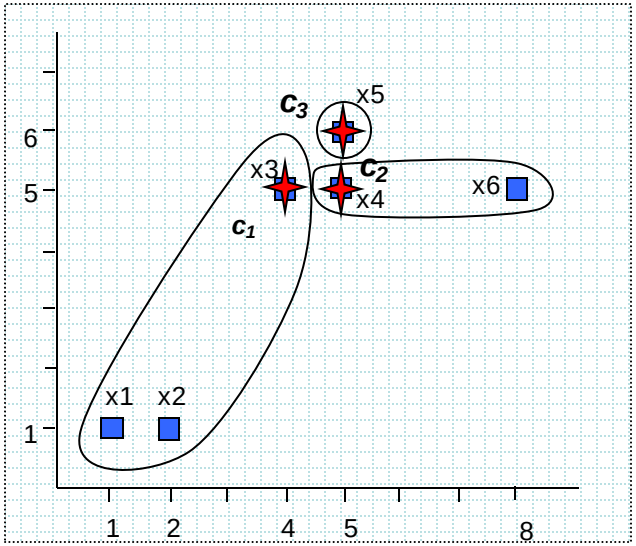
Exemple (suite)

Itération 1

Estimation de la matrice de distance →

	c1	c2	c3
x1	5.0	5.7	6.4
x2	4.5	5.0	5.8
x3	0.0	1.0	1.4
x4	1.0	0.0	1.0
x5	1.4	1.0	0.0
x6	4.0	3.0	3.2

Estimation de la matrice U ↘

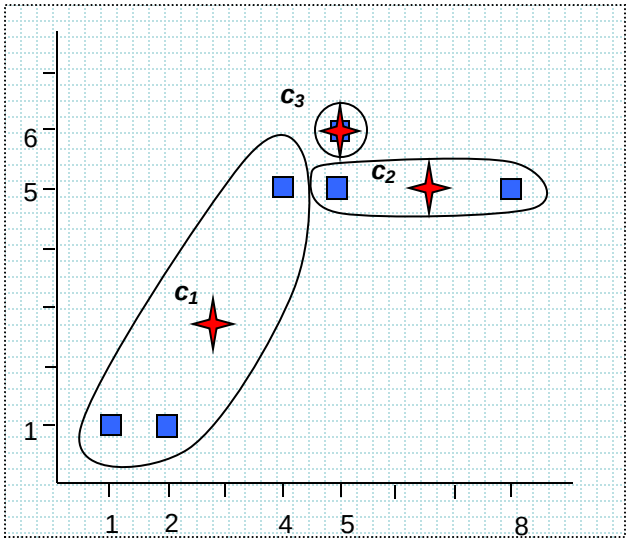


	c1	c2	c3
x1	1	0	0
x2	1	0	0
x3	1	0	0
x4	0	1	0
x5	0	0	1
x6	0	1	0

Exemple (suite)

Itération 1 (suite)

Mise à jour des centres des clusters



$$\mu_1 = \left(\frac{1+2+4}{3}, \frac{1+1+5}{3}\right) = (2.3, 2.3)$$

$$\mu_2 = \left(\frac{5+8}{2}, \frac{5+5}{2}\right) = (6.5, 5)$$

$$\mu_3 = (5, 6)$$

Exemple (suite)

Itération 1 (suite et fin)

Vérifier la convergence de l'algorithme :

- Est-ce que les centres de clusters se sont déplacés de leurs anciennes positions ? → Oui
- Est-ce que la partition est stable ? (pas de changement dans la matrice U) → Non

➢ L'algorithme n'a pas encore convergé, on passe donc à l'itération suivante

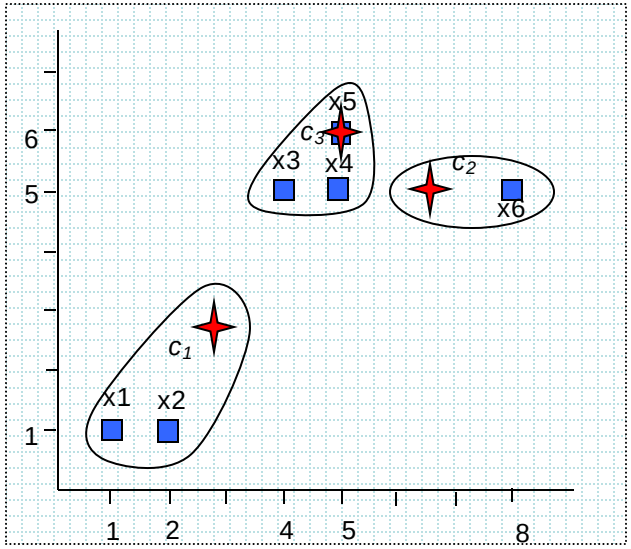
Exemple (suite)

Itération 2

Estimation de la matrice de distance →

	c1	c2	x3
x1	1.9	6.8	6.4
x2	1.4	6	5.8
x3	3.1	2.5	1.4
x4	3.8	1.5	1.0
x5	4.5	1.8	0.0
x6	6.3	1.5	3.2

Estimation de la matrice U ↘

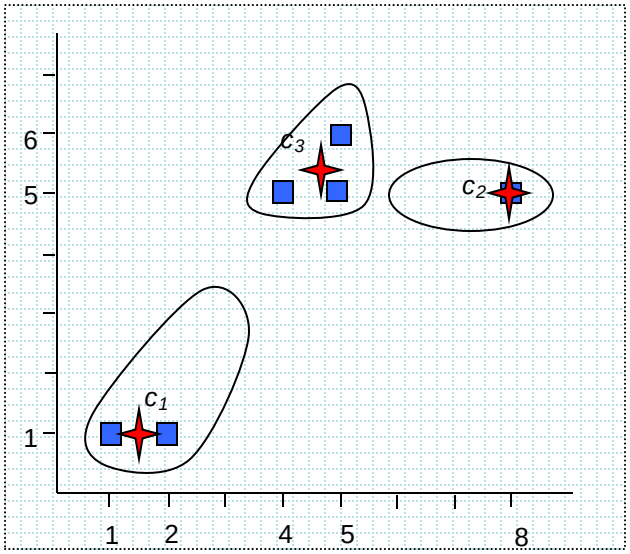


	c1	c2	c3
x1	1	0	0
x2	1	0	0
x3	0	0	1
x4	0	0	1
x5	0	0	1
x6	0	1	0

Exemple (suite)

Itération 2 (suite)

Mise à jour des centres des clusters



$$\mu_1 = \left(\frac{1+2}{2}, \frac{1+1}{2}\right) = (1.5,1)$$

$$\mu_2 = (8,5)$$

$$\mu_3 = \left(\frac{4+5+5}{3}, \frac{5+5+6}{3}\right) = (4.7,5.3)$$

Exemple (suite)

Itération 2 (suite et fin)

Vérifier la convergence de l'algorithme :

- Est-ce que les centres de clusters se sont déplacés de leurs anciennes positions ? → Oui
- Est-ce que la partition est stable? (pas de changement dans la matrice U)→ Non

➢ L'algorithme n'a pas encore convergé, on passe donc à l'itération suivante

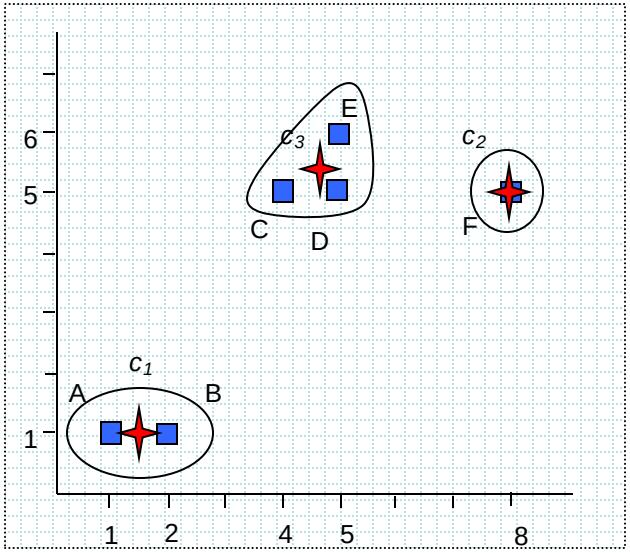
Exemple (suite)

Itération 3

Estimation de la matrice de distance →

	c1	c2	x3
x1	0.5	8.1	5.7
x2	0.5	7.2	5.1
x3	4.7	4	0.7
x4	5.3	3	0.5
x5	6.1	3.2	0.7
x6	7.6	0	3.3

Estimation de la matrice U ↘

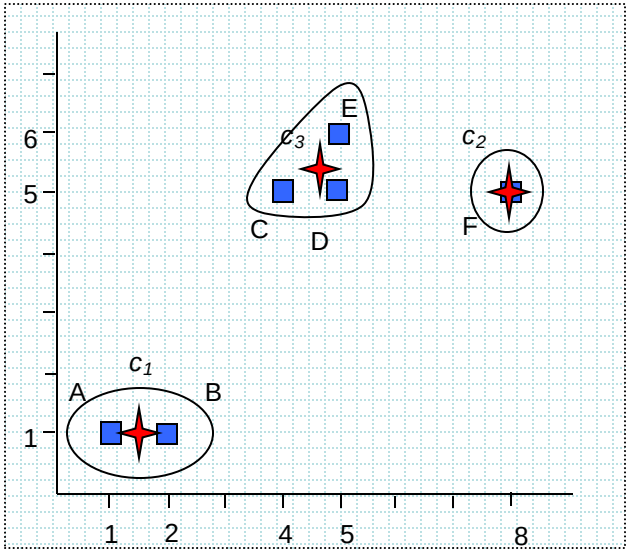


	c1	c2	c3
x1	1	0	0
x2	1	0	0
x3	0	0	1
x4	0	0	1
x5	0	0	1
x6	0	1	0

Exemple (suite)

Itération 3 (suite)

Mise à jour des centres des clusters



$$\mu_1 = \left(\frac{1+2}{2}, \frac{1+1}{2}\right) = (1.5,1)$$

$$\mu_2 = (8,5)$$

$$\mu_3 = \left(\frac{4+5+5}{3}, \frac{5+5+6}{3}\right) = (4.7,5.3)$$

Exemple (suite et fin)

Itération 3 (suite et fin)

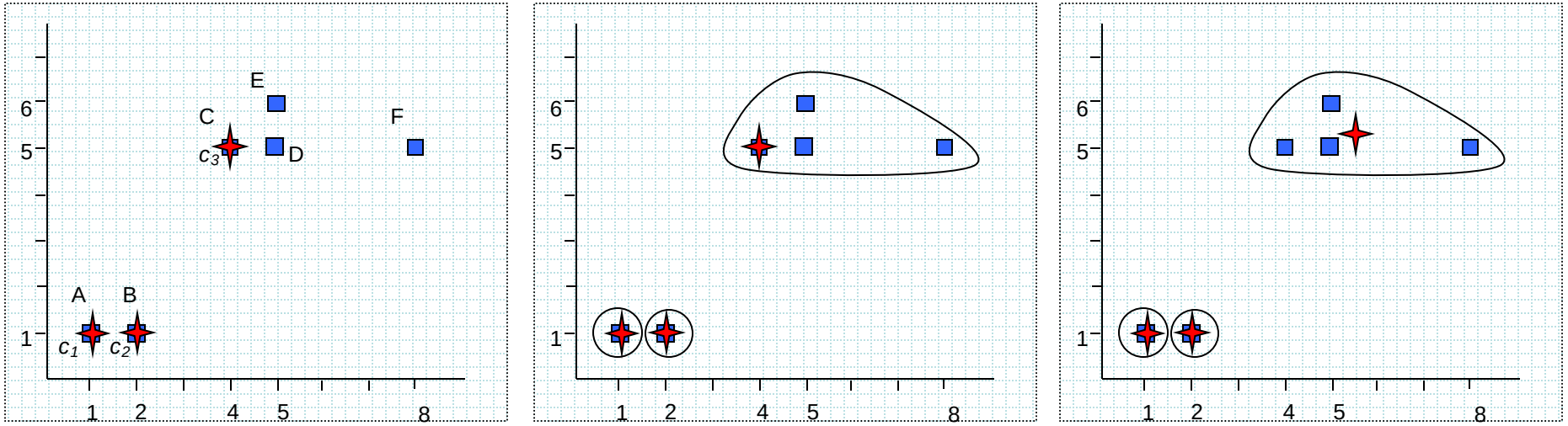
Vérifier la convergence de l'algorithme :

- Est-ce que les centres de clusters se sont déplacés de leurs anciennes positions ? → Non
- Est-ce que la partition est stable ? (pas de changement dans la matrice U)→ Oui

➢ L'algorithme a convergé

➢ Fin

Contre exemple



(a) (b) (c)

(a) Une autre initialisation aléatoire des centres

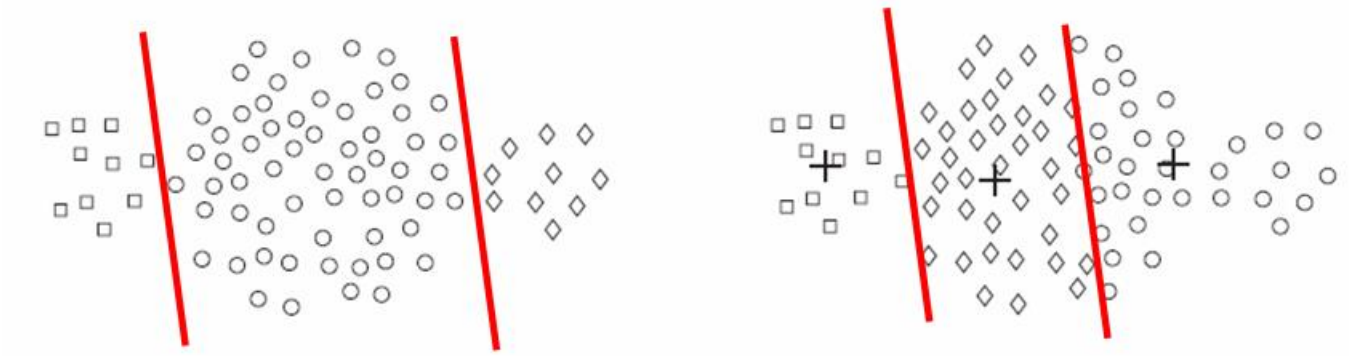
(b) Mise à jour

(c) Clustering final → ce résultat ne relate pas le clustering réel

- **K-means est sensible à la sélection initiale des centres**

Comportement de *K*-means

- Problème avec les clusters de tailles différentes

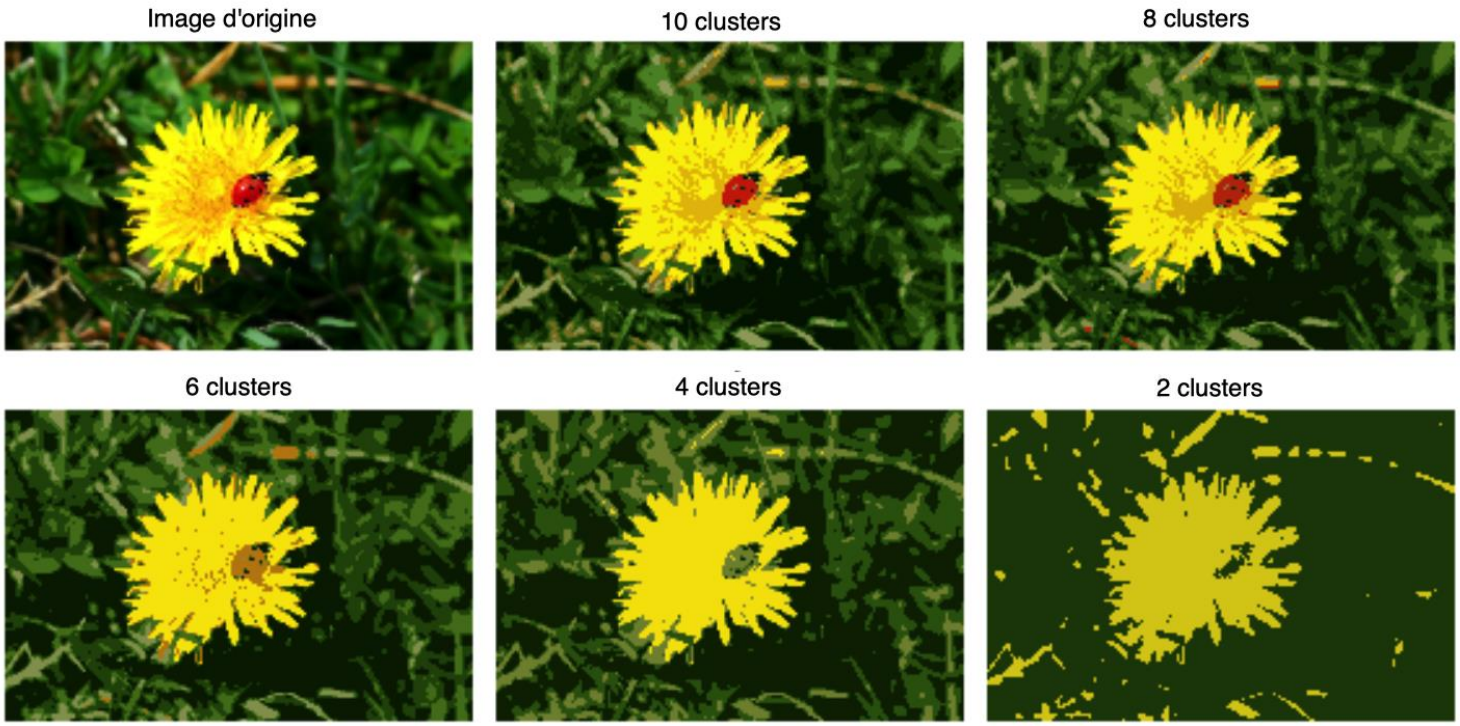


(a) (b)

(a) Clustering réel

(b) Clustering avec *K*-means

Comportement de *K*-means

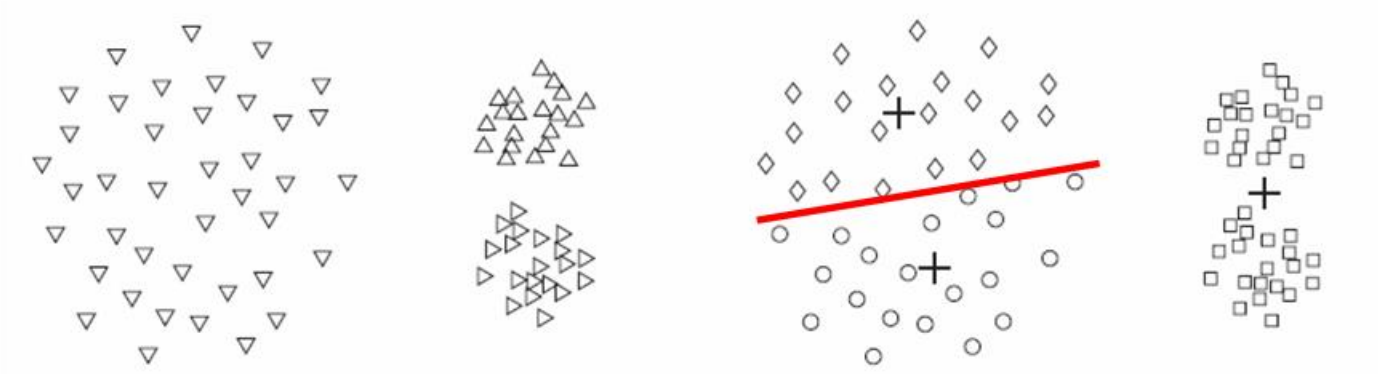


Si on utilise moins de 8 clusters, on remarque que le rouge vif de la coccinelle ne réussit pas à obtenir son propre cluster : il se retrouve mélangé à d'autres couleurs de l'environnement. Ceci est dû au fait que l'algorithme des k-means préfère les clusters de taille similaire. La coccinelle est petite, bien plus petite que le reste de l'image et, bien que sa couleur soit vive, l'algorithme des k-means ne réussit pas à lui associer un cluster.

*Source : Aurélien Geron. Machine Learning avec Scikit-Learn : Mise en œuvre et cas concrets. Dunod, 2019.

Comportement de *K*-means

- Problème avec les clusters de densités différentes



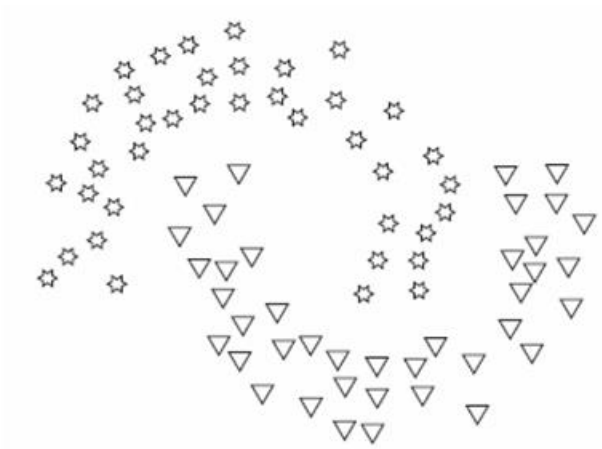
(a) (b)

(a) Clustering réel

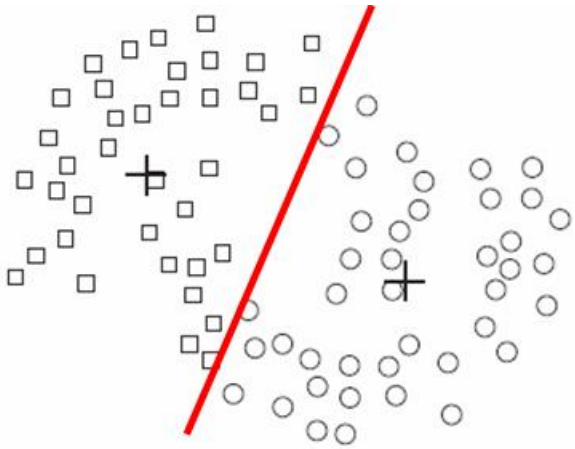
(b) Clustering avec *K*-means

Comportement de *K*-means

- Problème avec les clusters de formes non sphériques.



(a)



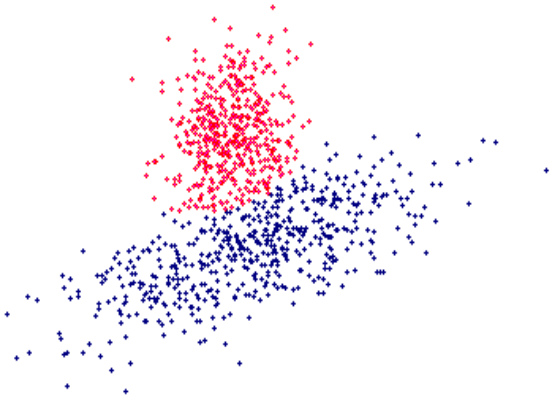
(b)

(a) Clustering réel

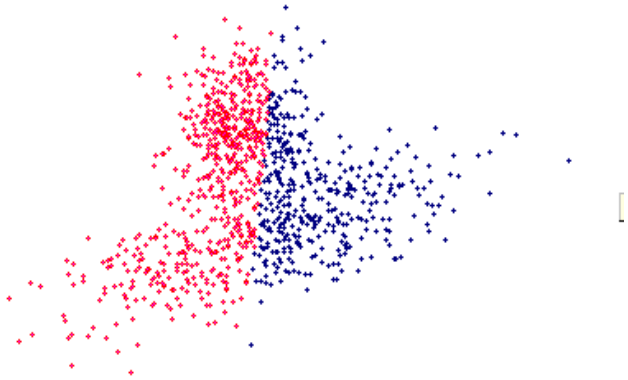
(b) Clustering avec *K*-means

Comportement de *K*-means

- Problème avec les clusters de formes allongées



(a)



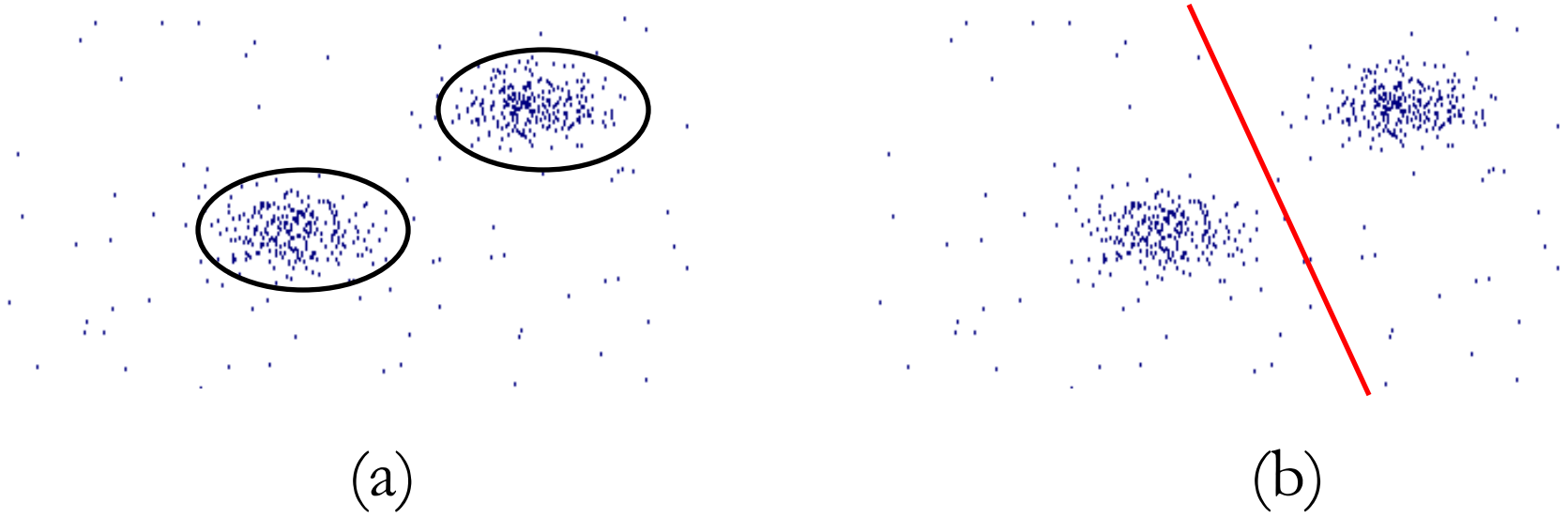
(b)

(a) Clustering réel

(b) Clustering avec *K*-means

Comportement de *K*-means

- Problème avec les données bruitées



(a) (b)

(a) Clustering réel

(b) Clustering avec *K*-means

Caractéristiques de *K*-means

- **Avantages**

  → Relativement efficace (rapide)

  → Converge souvent

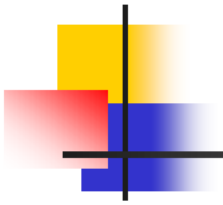
- **Faiblesses**

  → Besoin de spécifier *K* à l'avance

  → Ne gère pas le bruit

  → Ne trouve que des clusters de forme sphérique

  → Sensible à la sélection initiale des centres de clusters



Plan

1 – Mise en contexte

2 – Clustering avec K-means

3 – Clustering hiérarchique

4 – Clustering basé sur la densité

5 – Clustering des graphes

Clustering hiérarchique

- Un algorithme de clustering hiérarchique ne produit pas une seule partition, mais une hiérarchie de partitions emboîtées.
- Dans ce contexte, un cluster est défini comme un nœud d'arbre, auquel est associé l'ensemble des objets qui le composent, ainsi leurs caractéristiques.

- Il existe deux grandes catégories d'algorithmes hiérarchiques :
  1. Méthodes ascendantes ou agglomeratives
  2. Méthodes descendantes

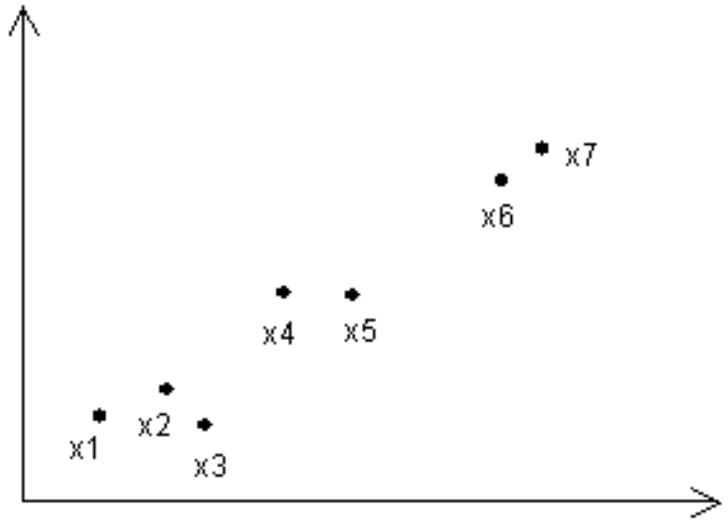
Clustering hiérarchique

- **Méthodes ascendantes ou agglomératives**

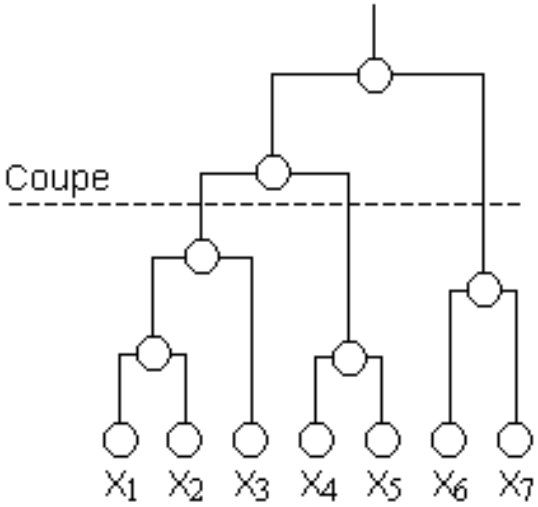
- La partition initiale contient autant de clusters que d'objets (K = n).
- A chaque étape, on cherche un couple (C_i ,C_j) de clusters candidats à la fusion qui maximise (reps. minimise) une certaine mesure de similarité (resp. de dissimilarité).
- On réitère ce processus jusqu'à l'obtention d'un seul cluster contenant tous les éléments.
- A fin de déterminer le nombre de clusters, on coupe la hiérarchie à un certain niveau de détail.
- La hiérarchie de partitions est représentée sous forme appelée dendrogramme.

Clustering hiérarchique

Méthodes ascendantes ou agglomératives - Exemple



Un ensemble d'objets à classer



Dendogramme de la partition

Clustering hiérarchique

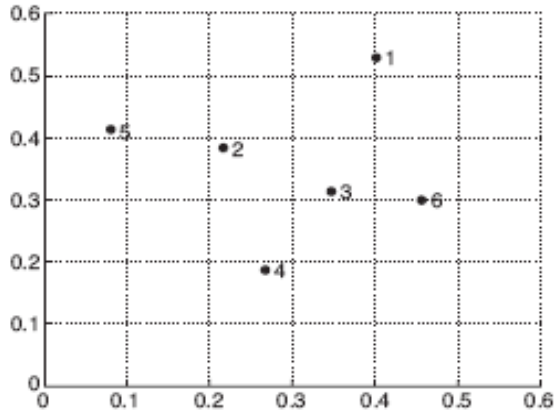
Méthodes descendantes.

- Commencer avec un cluster contenant toutes les données
- Séparer les groupes en plus petits groupes jusqu'à ce que chaque groupe ne contienne plus qu'un objet
- Dans cette approche on a besoin de décider qu'elle est le cluster qu'on doit le diviser, à qu'elle étape et comment faire la division

⇨ Les approches agglomératives sont les plus utilisées dans la pratique

Clustering hiérarchique

- Une approche de clustering hiérarchique utilise une matrice de distance



Ensemble de six points dans R^2

Point	x Coordinate	y Coordinate
p1	0.40	0.53
p2	0.22	0.38
p3	0.35	0.32
p4	0.26	0.19
p5	0.08	0.41
p6	0.45	0.30

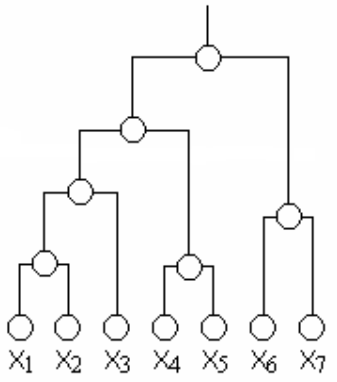
cordonnées

	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

Matrice de distance

- Ne nécessite pas de spécifier le nombre de clusters

Approches agglomératives

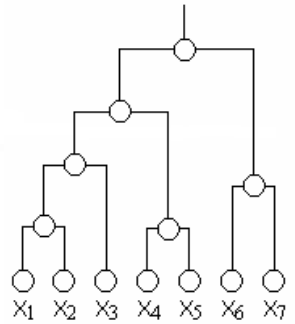


- Algorithme de base

1. Initialement, considérer chaque objet comme cluster;
2. Parmi tous les clusters, identifier deux clusters qui sont les plus proches l'un de l'autre;
3. Fusionner ces deux clusters;
4. Répéter l'étape 2 et 3 jusqu'à l'obtention d'un seul cluster contenant tous les éléments.

- L'étape clé dans cet algorithme est la mesure de la similarité entre les clusters

Approches agglomératives



- Comment mesurer la distance entre les clusters?

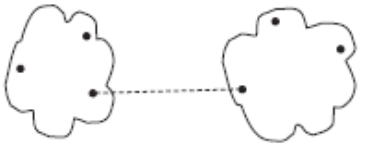
Trois mesures de similarité se distinguent en clustering hiérarchique:

1. Lien unique (Single link)
2. Lien complet (Complete link)
3. Lien moyen (Average link)

Mesures de distance entre les clusters

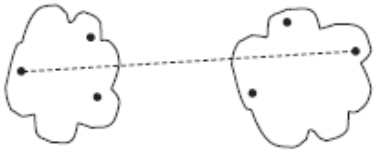
1. Lien unique : la similarité entre deux clusters est le minimum des distances entre toutes les paires de données entre deux clusters.

$$dist_{lien\_unique}(C_i, C_j) = \min_{\substack{x \in C_i \\ y \in C_j}} (\|x - y\|)$$



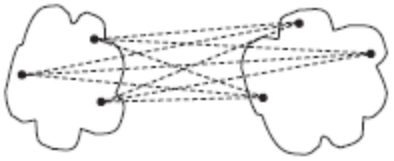
2. Lien complet : la similarité entre deux clusters est le maximum des distances entre toutes les paires de données entre deux clusters.

$$dist_{lien\_complet}(C_i, C_j) = \max_{\substack{x \in C_i \\ y \in C_j}} (\|x - y\|)$$



3. Lien moyen : la similarité est définie par la moyenne de ces distances

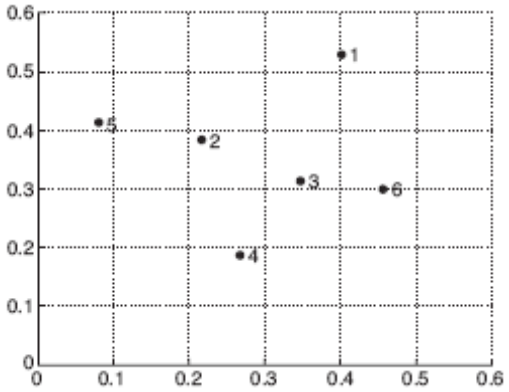
$$dist_{lien\_moyen}(C_i, C_j) = \frac{1}{n_i n_j} \sum_{x \in C_i} \sum_{y \in C_j} \|x - y\|$$



n_i et n_j représentent la taille de C_i et C_j respectivement

Approches agglomératives

Dans ce qui suit on va illustrer le comportement des différentes mesures de similarité entre les clusters sur l'ensemble de données D suivants :



Ensemble de six points dans R^2

Point	x Coordinate	y Coordinate
p1	0.40	0.53
p2	0.22	0.38
p3	0.35	0.32
p4	0.26	0.19
p5	0.08	0.41
p6	0.45	0.30

cordonnées

- Identifier les clusters dans D et ce en utilisant comme mesure de similarité
  - Lien unique
  - Lien complet
  - Lien moyen

Lien unique (Single link) - Exemple

- Étape 1: estimer la matrice de la distance

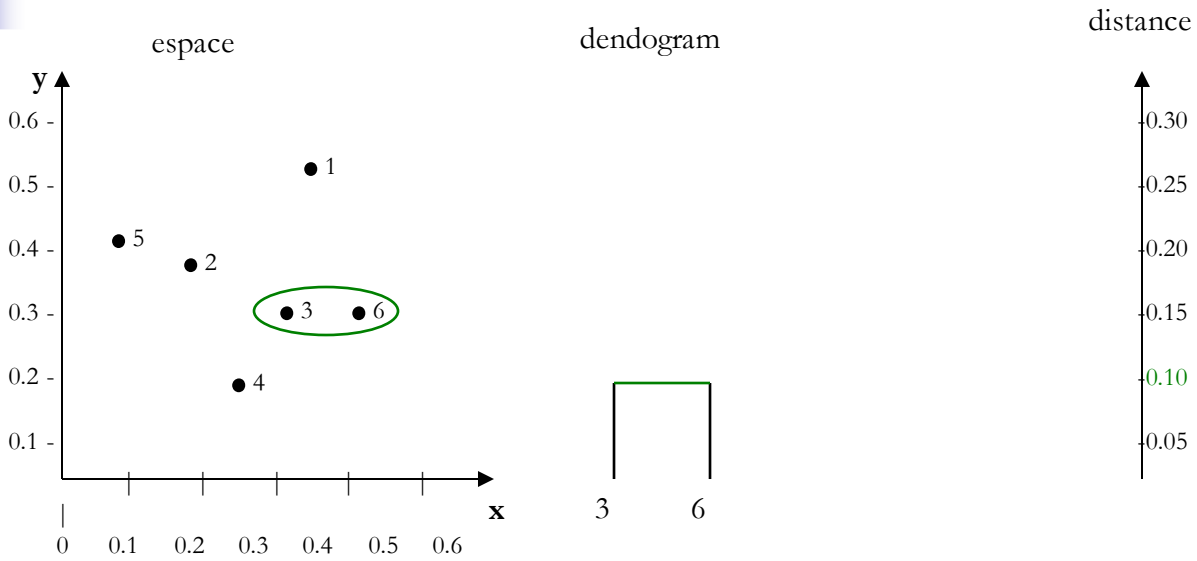
	p1	p2	p3	p4	p5	p6
p1	0					
p2	0.24	0				
p3	0.22	0.15	0			
p4	0.37	0.20	0.15	0		
p5	0.34	0.14	0.28	0.29	0	
p6	0.23	0.25	0.11	0.22	0.39	0

- Étape 2:
  - À partir de la matrice de distance, identifiez les deux clusters avec la plus petite distance
  - Fusionner ces deux clusters
  - Mettre à jour la matrice de distance

- Les points p3 et p6 sont les plus proches

	p1	p2	p3	p4	p5	p6
p1	0					
p2	0.24	0				
p3	0.22	0.15	0			
p4	0.37	0.20	0.15	0		
p5	0.34	0.14	0.28	0.29	0	
p6	0.23	0.25	0.11	0.22	0.39	0

Lien unique (Single link) - Exemple



Mise à jour de la matrice de distance

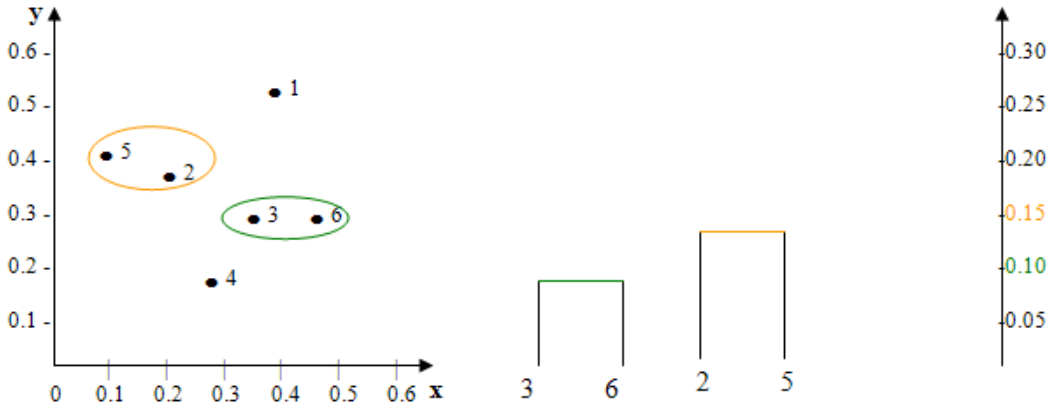
	p1	p2	(p3,p6)	p4	p5
p1	0				
p2	0.24	0			
(p3, p6)	**0.22**	**0.15**	0		
p4	0.37	0.20	**0.15**	0	
p5	0.34	0.14	**0.28**	0.29	0

$$dist((p3,p6),\ p1) = \min(dist(p3,p1),\ dist(p6,p1))$$
$$= \min(0.22,\ 0.23)$$
$$= 0.22$$

Lien unique (Single link) - Exemple

➢ Étape 3 : Répéter l'étape 2 de l'algorithme jusqu'à ce que tous les clusters soient fusionnés

- À partir de la nouvelle matrice de distance, on remarque que la plus petite distance est entre p2 et p5 donc on doit fusionner ces deux clusters et on doit recalculer la matrice de distance



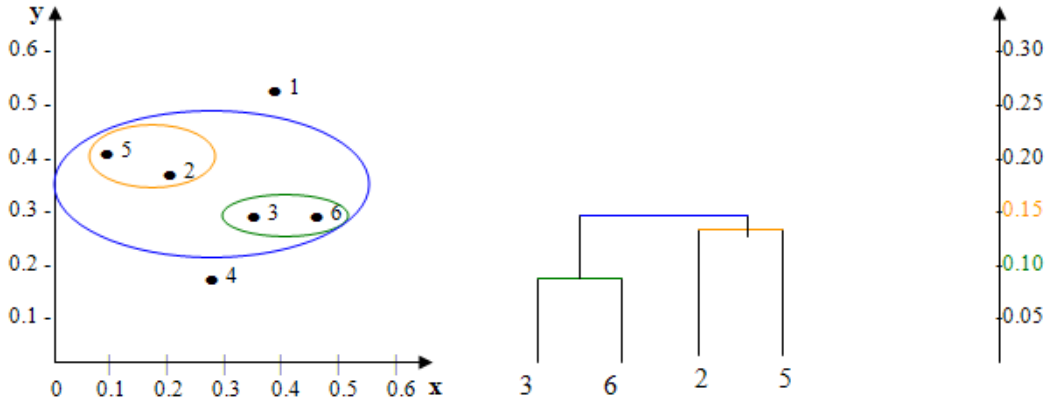
Mise à jour de la matrice de distance

	p1	(p2, p5)	(p3, p6)	p4
p1	0			
(p2, p5)	**0.24**	0		
(p3, p6)	0.22	**0.15**	0	
p4	0.37	**0.20**	0.15	0

dist((p3, p6), (p2, p5)) = min (*dist*(p3, p2) , *dist*(p6, p2), *dist*(p3, p5), *dist*(p6, p5))
= min (0.15 , 0.25, 0.28, 0.39)
= 0.15

Lien unique (Single link) - Exemple

- À partir de la nouvelle matrice de distance, on remarque que la plus petite distance est entre (p3, p6) et (p2, p5). Le même phénomène et aussi observé avec p4 et (p3,p6).
- Pour la fusion on doit choisir un seul cas parmi ces deux cas,
- On choisit (p3, p6) et (p2, p5)

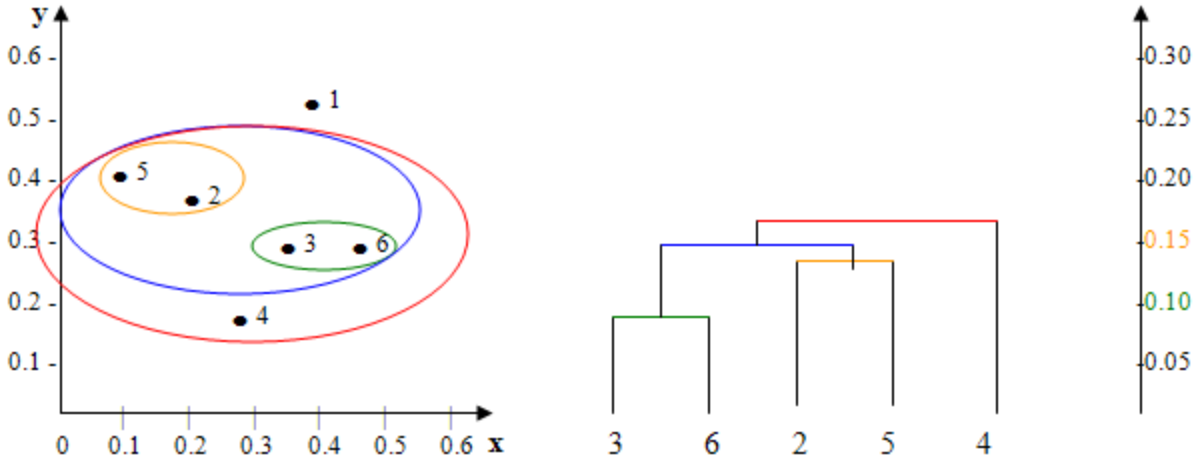


Mise à jour de la matrice de distance

	p1	(p2, p5, p3, p6)	p4
p1	0		
(p2, p5, p3, p6)	**0.22**	0	
p4	0.37	**0.15**	0

Lien unique (Single link) - Exemple

- À partir de la nouvelle matrice de distance, on remarque que la plus petite distance est entre p4 et (p2, p5, p3, p6) donc on doit fusionner ces deux clusters et on doit recalculer la matrice de distance

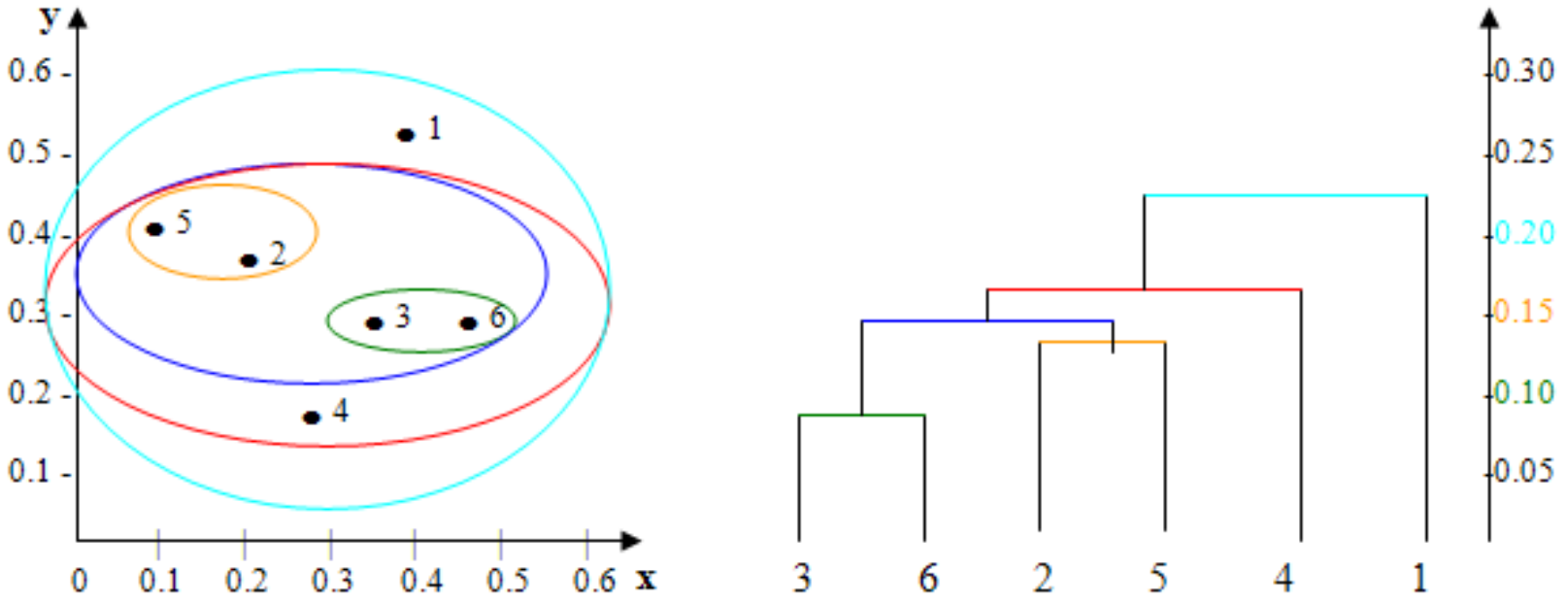


Mise à jour de la matrice de distance

	p1	(p2, p5, p3, p6, p4)
p1	0	
(p2, p5, p3, p6, p4)	0.22	0

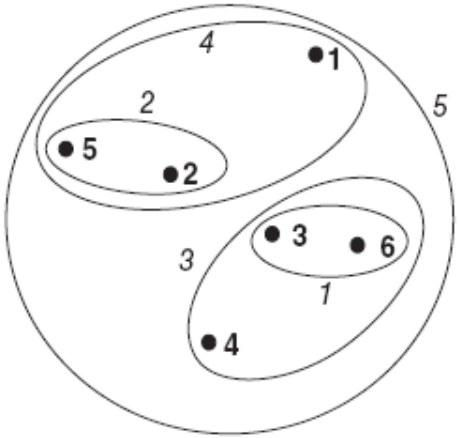
Lien unique (Single link) - Exemple

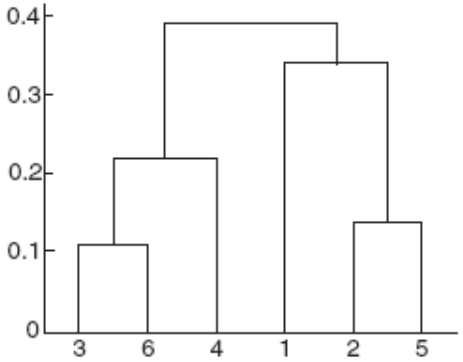
- À partir de la nouvelle matrice de distance, on remarque que la plus petite distance est entre p1 et (p2, p5, p3, p6, p4) donc on doit fusionner ces deux clusters et on doit recalculer la matrice de distance



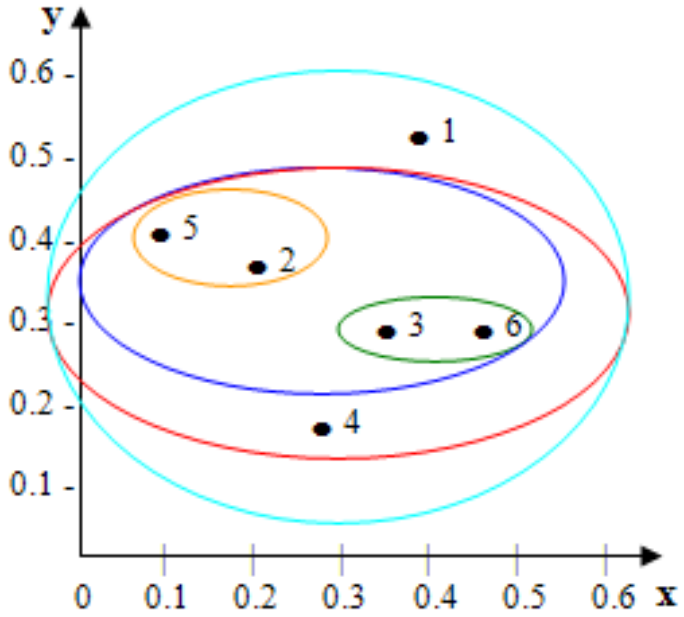
Lien complet (Complete link) - Exemple

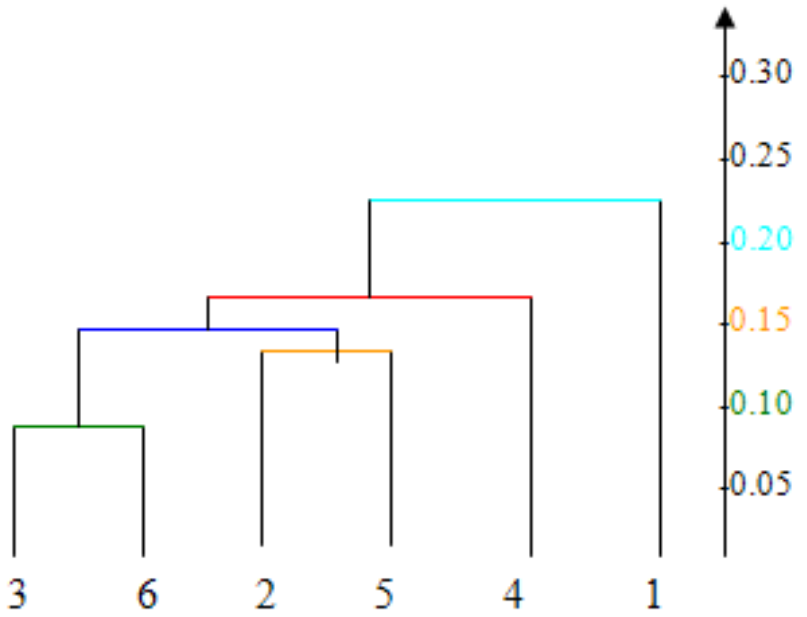
- On adopte le même processus décrit précédemment, sauf on utilise la mesure lien complet comme mesure de similarité

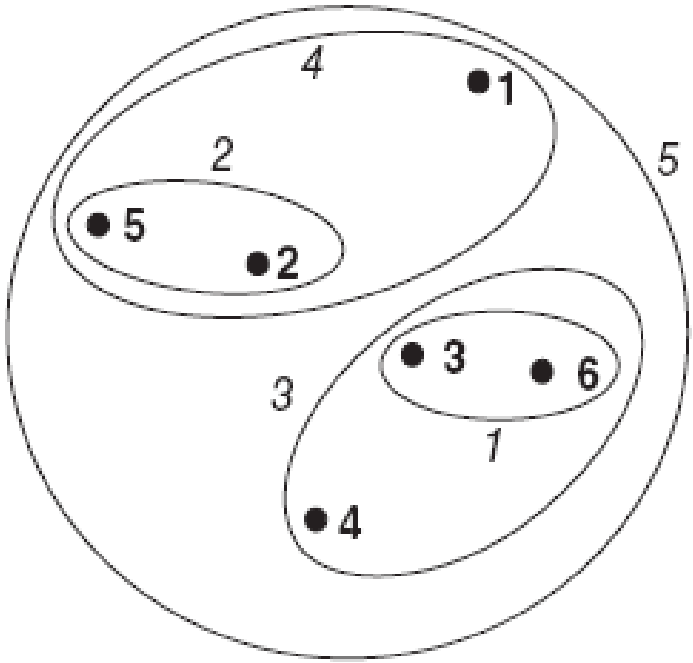


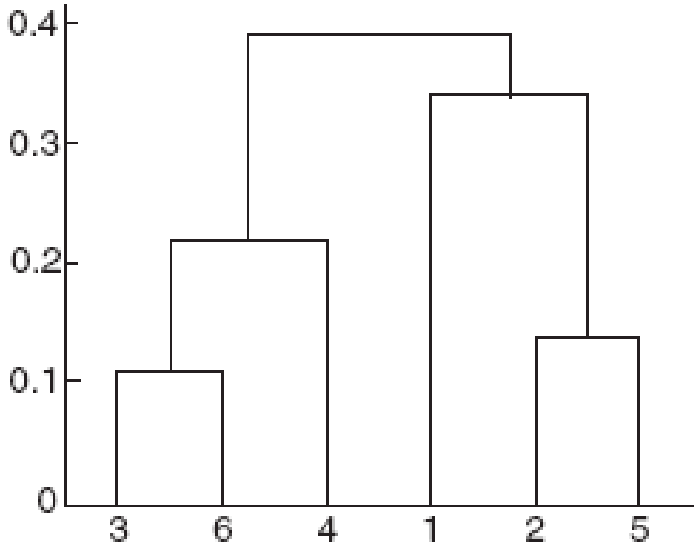


$$
\begin{aligned}
dist(\{3,6\},\{4\}) &= \max(dist(3,4), dist(6,4)) \\
&= \max(0.15, 0.22) \\
&= 0.22. \\
dist(\{3,6\},\{2,5\}) &= \max(dist(3,2), dist(6,2), dist(3,5), dist(6,5)) \\
&= \max(0.15, 0.25, 0.28, 0.39) \\
&= 0.39. \\
dist(\{3,6\},\{1\}) &= \max(dist(3,1), dist(6,1)) \\
&= \max(0.22, 0.23) \\
&= 0.23.
\end{aligned}
$$


y
0.6
0.5
0.4
0.3
0.2
0.1
0
0.1
0.2
0.3
0.4
0.5
0.6
x
1
2
3
4
5
6

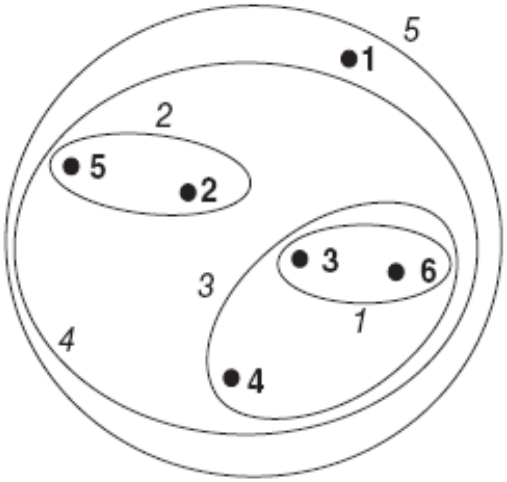

0.30
0.25
0.20
0.15
0.10
0.05
3
6
2
5
4
1

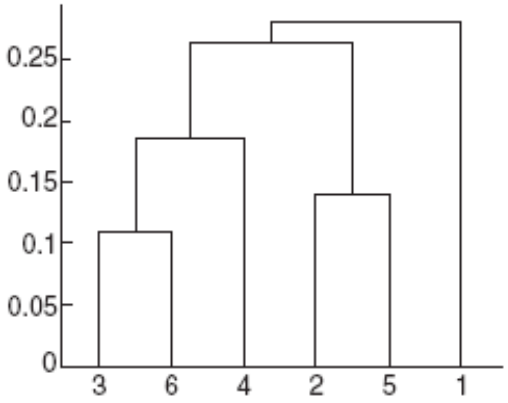

4
1
2
5
5
2
3
6
3
1
4


0.4
0.3
0.2
0.1
0
3
6
4
1
2
5

Lien moyen (Average link) - Exemple

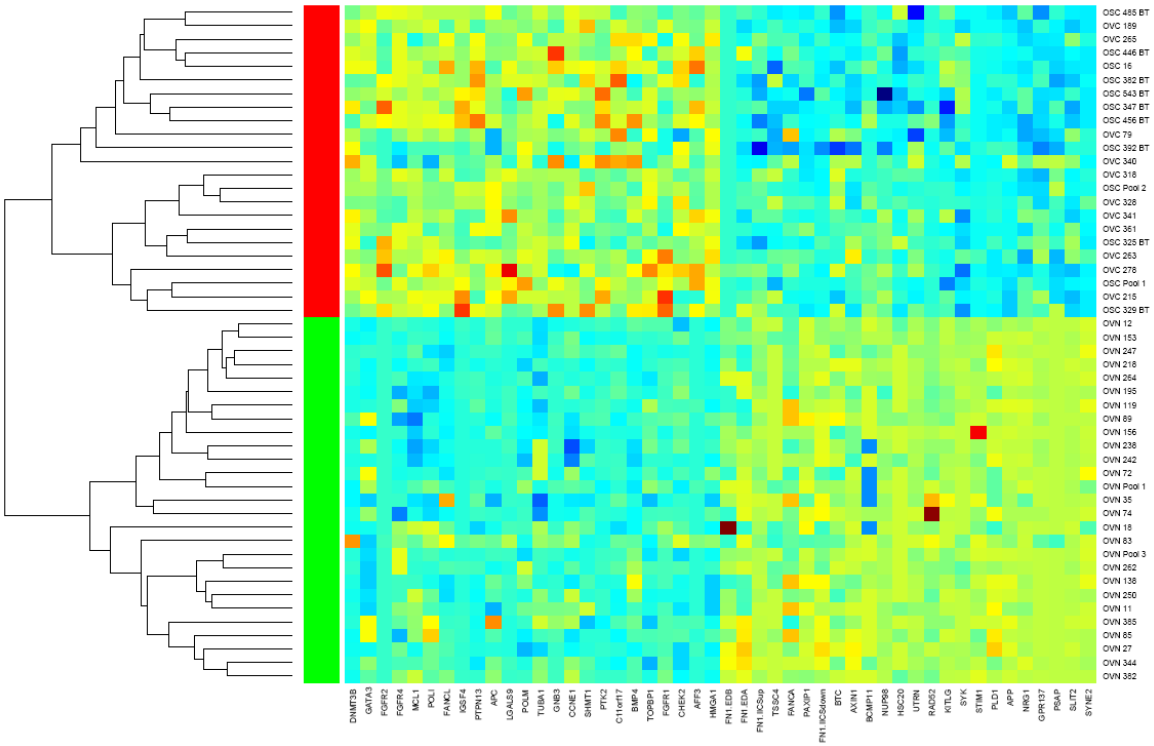
- On adopte le même processus décrit précédemment, sauf on utilise la mesure lien moyen comme mesure de similarité

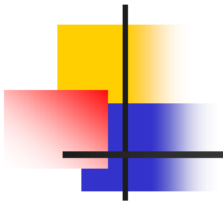




Discussion

- Les approches de clustering hiérarchique sont pertinentes dans les domaines où il y a une relation naturelle entre les clusters. Un exemple concret : la biologie
- La taxonomie des plantes est des animaux peut-être vus comme une hiérarchie de clusters
- La relation entre les gènes





Plan

1 – Mise en contexte

2 – Clustering avec K-means

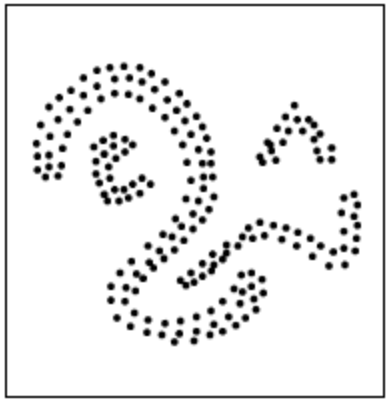
3 – Clustering hiérarchique

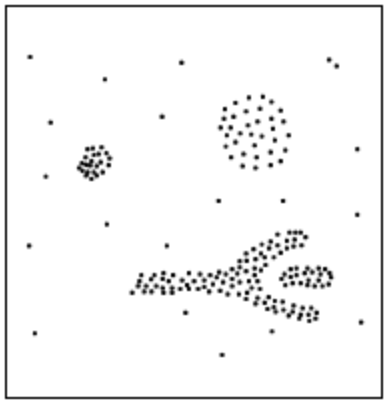
4 – Clustering basé sur la densité

5 – Clustering des graphes

Clustering basé sur la densité

- Les techniques de clustering vu jusqu'à maintenant ne permettent pas d'identifier des clusters de forme : étiré, linéaire, allongé, etc.
- Les données spatiales (ex. images satellites) contiennent des clusters de forme variables.
- Des exemples d'applications pratiques :
  - Le regroupement des maisons le long d'une rivière
  - L'extraction des autoroutes

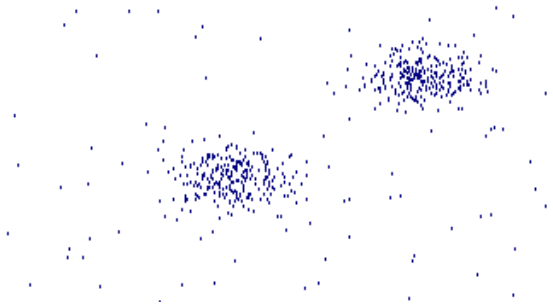




➢ DBSCAN (Density-Based Spatial Clustering of Applications with Noise) est capable d'identifier ce type de clusters

Clustering basé sur la densité

- Un cluster est région de grande densité entourée par des points avec une densité relativement faible
- Un bruit appartient à une région de très faible densité (*éparse*)

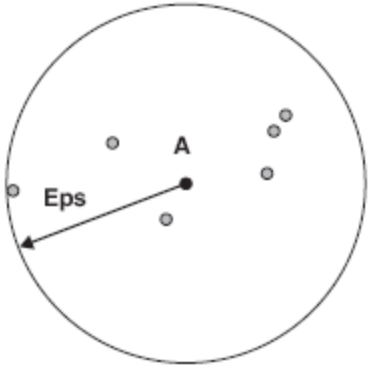


Comment identifier les régions denses?

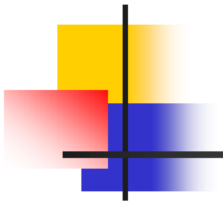
Clustering basé sur la densité

Idée

- L'idée clé du clustering basé sur la densité est que pour chaque point d'un cluster, ses environs pour un rayon donné Eps doit contenir un nombre minimum de points MinPts.



- On dit un point appartient à une région de forte densité si la cardinalité de son voisinage dépasse un certain seuil.



DBSCAN

Cette notion de densité permet de classifier un point comme

1. un point central (*a core point*) :
   - un point qui existe à l'intérieur d'une région de forte densité
2. un point de bordure (*a border point*) :
   - un point qui est très proche d'une région dense
3. un bruit (*noise, outlier*) :
   - un point qui existe dans une région de très faible densité

DBSCAN

- p est un point central si

$| N_{EPS}(p) | \geq MinPts$ avec $N_{EPS}(p) = \{q \in X \mid dist(p, q) \leq Eps \}$

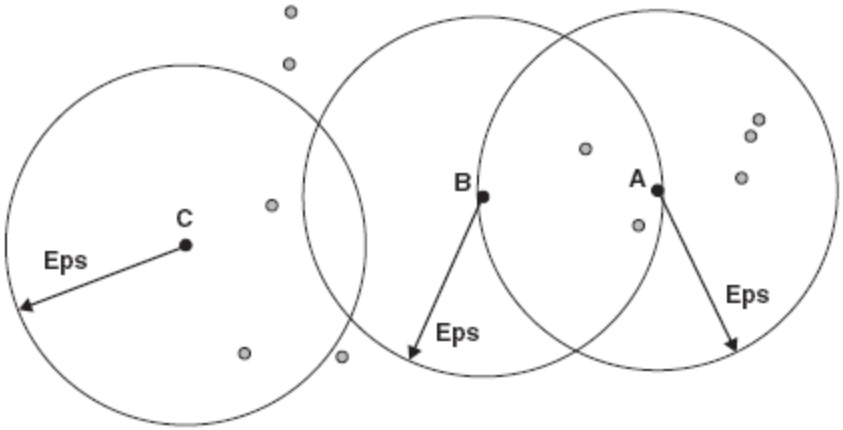
- Un point de bordure n'est pas un point central, mais appartient au voisinage d'un point central

- Un bruit n'est ni un point central ni un point de bordure

DBSCAN

- Illustration

MinPts = 7

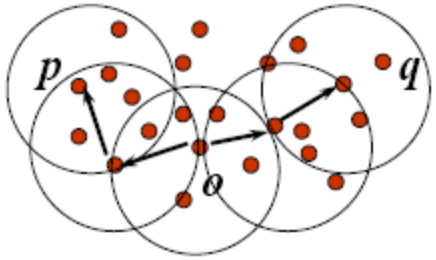


A est un point central

B est un point de bordure

C est un bruit

DBSCAN



Entrée : ensemble de données X, Eps, MinPts

Sortie : un ensemble de clusters C, l'ensemble des points qui représentent le bruit

1 : Identifier les points centraux, les points de bordures et les points qui représentent le bruit;

2 : Éliminer le bruit;

3 : Grouper les points centraux :

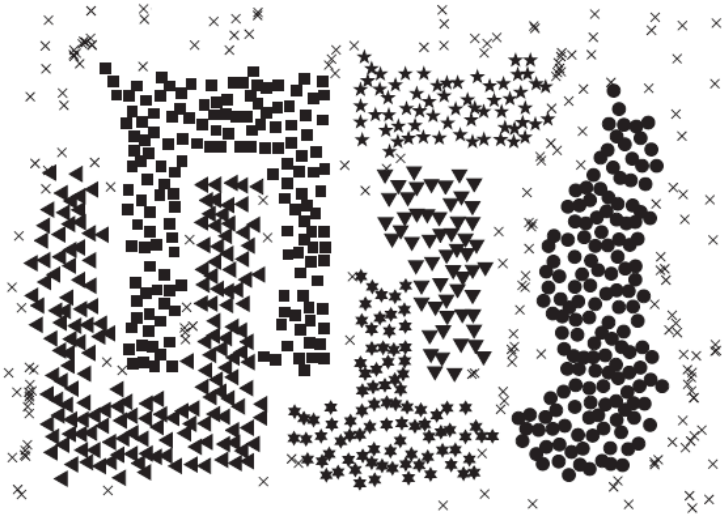
- Examiner le voisinage d'un point central p1 (c.-à-d.. les points qui se trouvent à l'intérieur d'un rayon Eps)
- Si parmi les points qui appartiennent à son voisinage existe un point central p2 alors mettre un arc entre ces deux-points
- répéter le même processus pour tous les points centraux

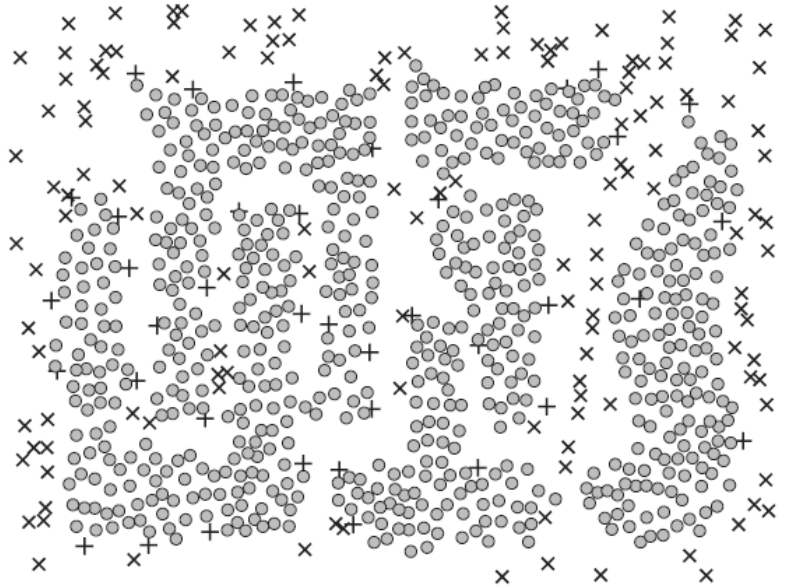
4 : Étiqueter chaque groupe identifié comme cluster

5 : Assigner chaque point de bordure à l'un des clusters qui contient son point central associé.

DBSCAN

Les clusters identifiés par DBSCAN →





- bruit
- point de bordure
- point central

DBSCAN

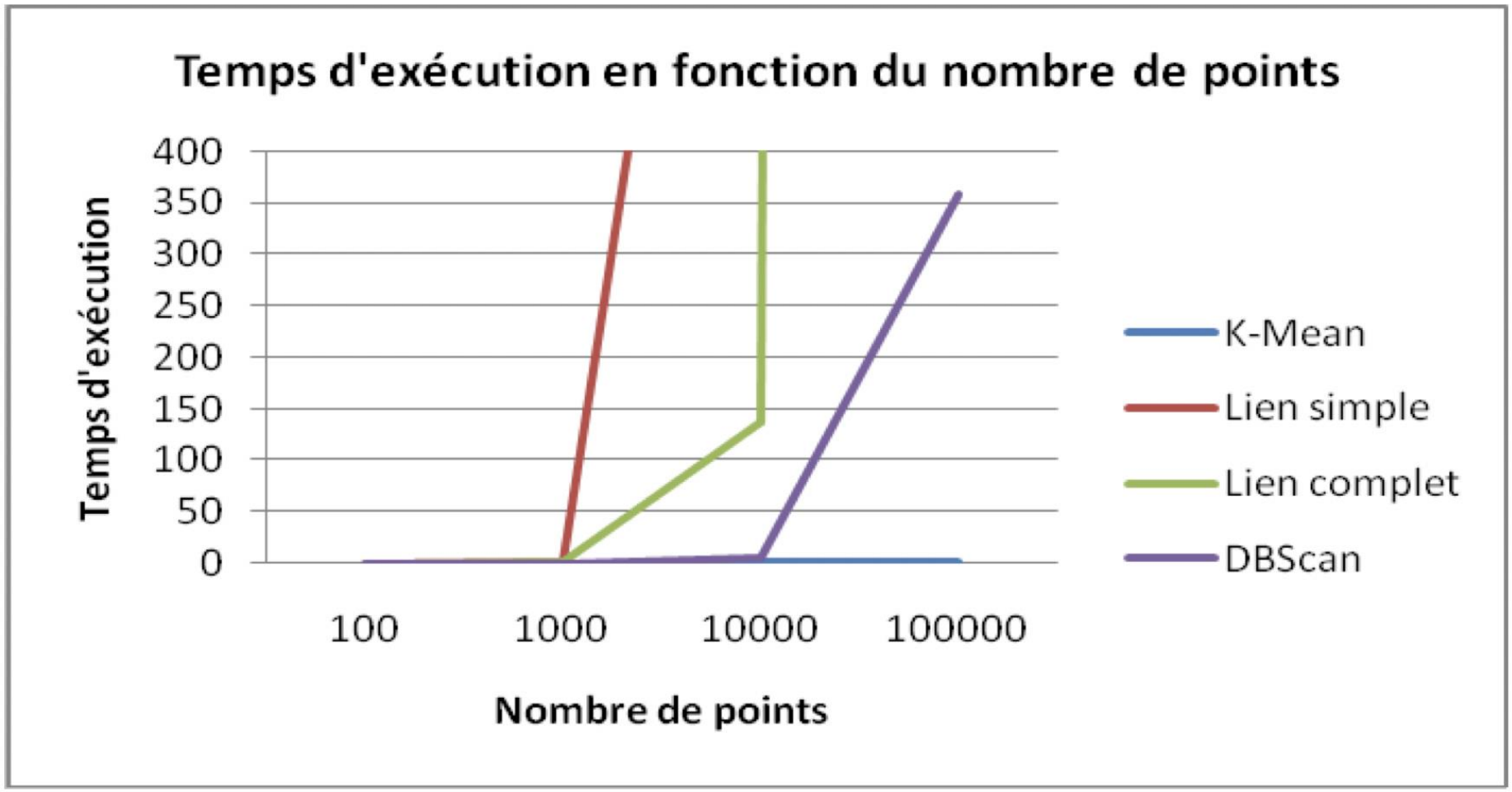
Discussion

- Capable d'identifier des clusters avec des formes arbitraires
- Résiste aux bruits
- Peut identifier les clusters identifiés par K-means
- Rencontre des difficultés lorsque l'ensemble de données contient des clusters avec des densités très variées
- Rencontre des difficultés avec les données de grande dimension (il n'est pas toujours évident de mesurer la densité avec les données multidimensionnelles)

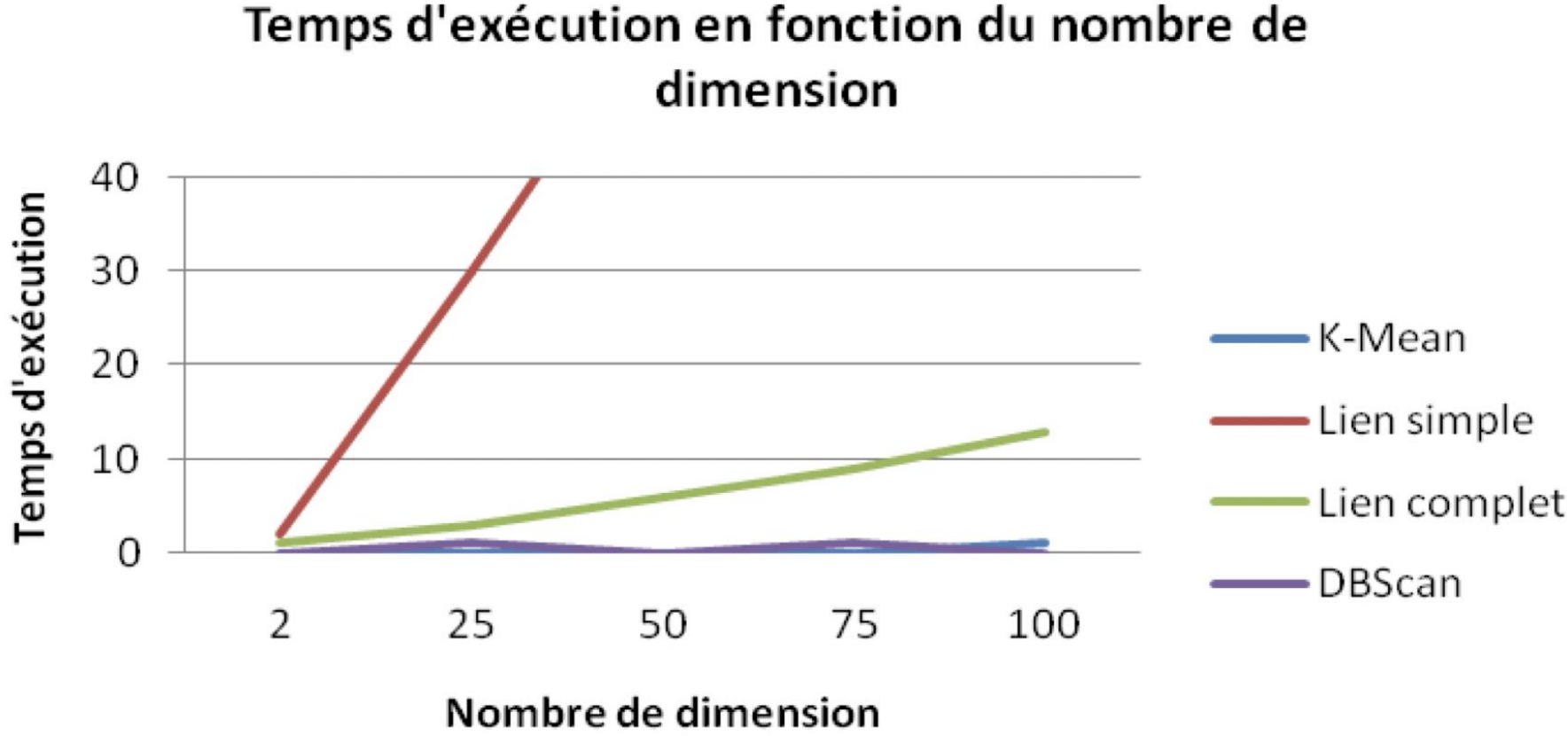
Comparaison

K-means vs hiérarchique vs DBSCAN

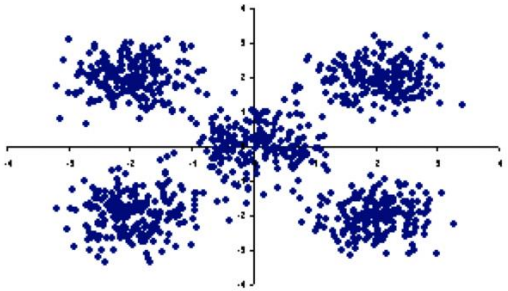
Temps d'exécution (en seconde)



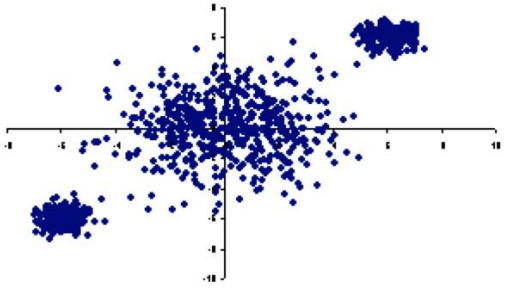
Temps d'exécution (en seconde)



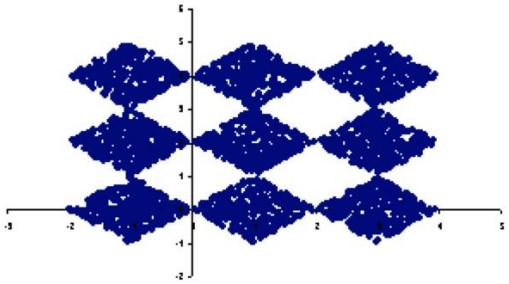
Comparaison sur une variété de données



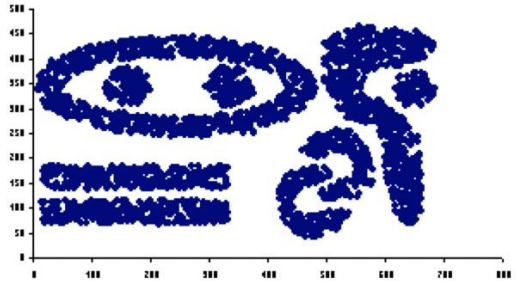
Data1_1



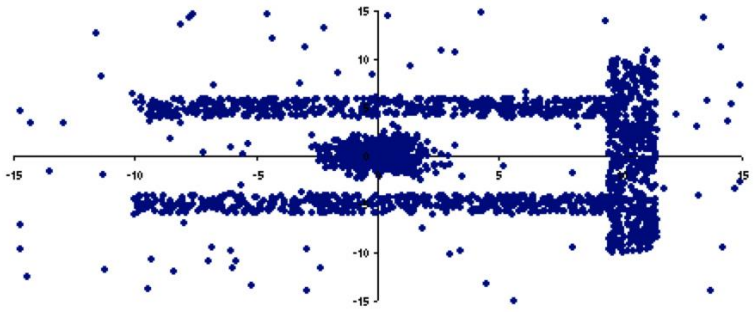
Data1_2



Data1_3

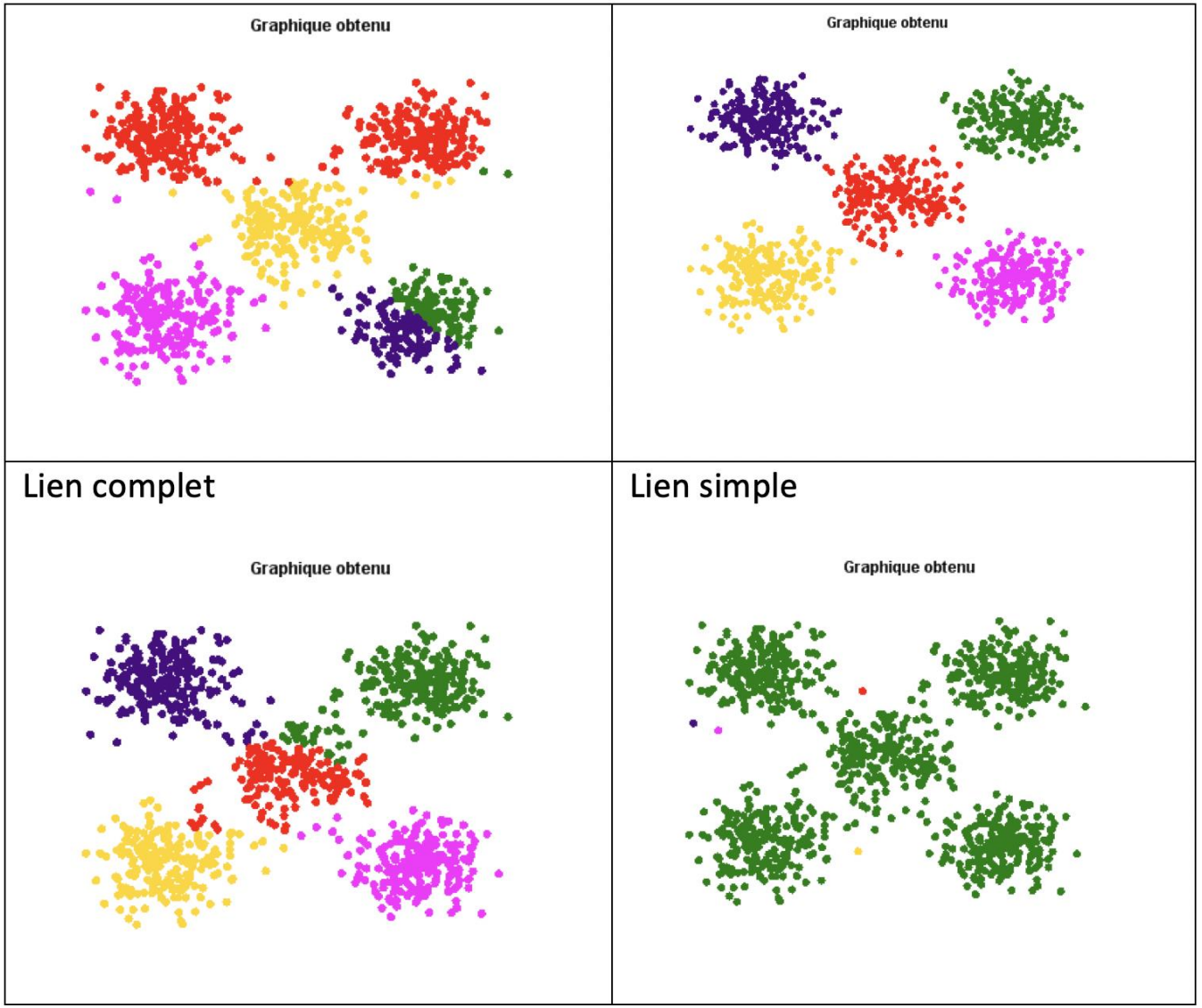


Data1_4



Data1_5

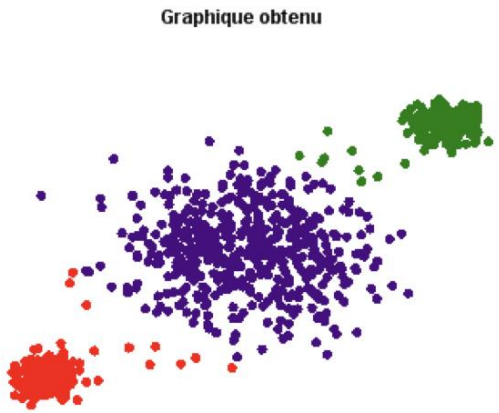
Comparaison sur une variété de données



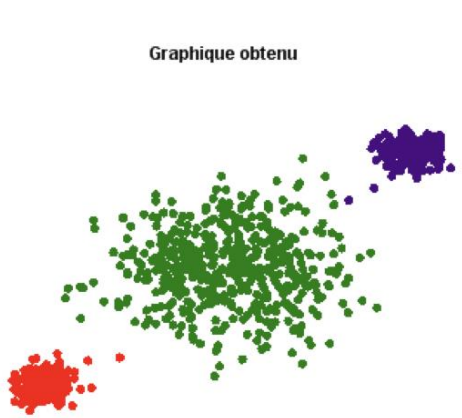
Data1_2

Comparaison sur une variété de données

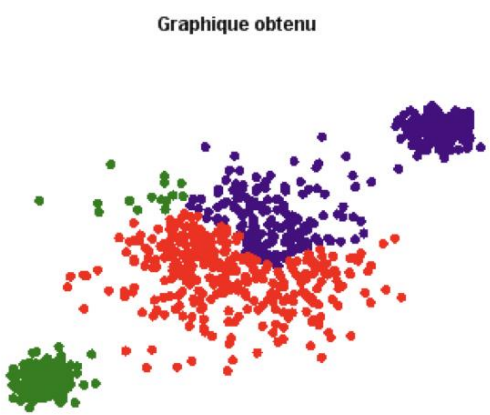
K-Means



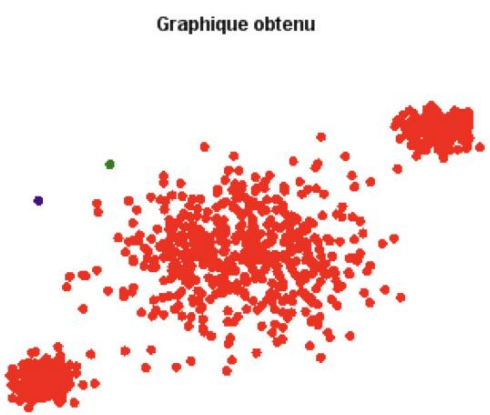
DBSCAN



Lien complet



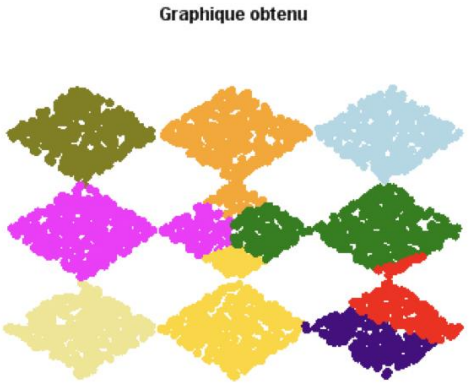
Lien simple



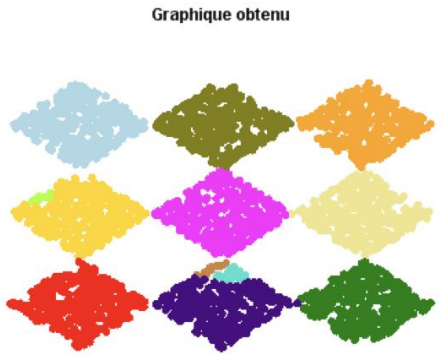
Comparaison sur une variété de données

Data1_3

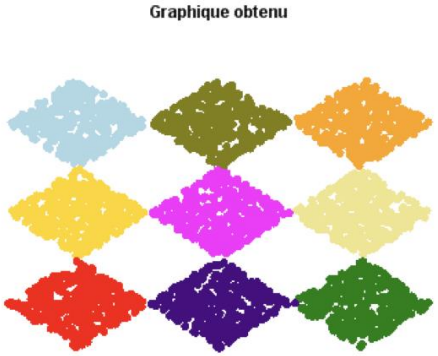
K-Means



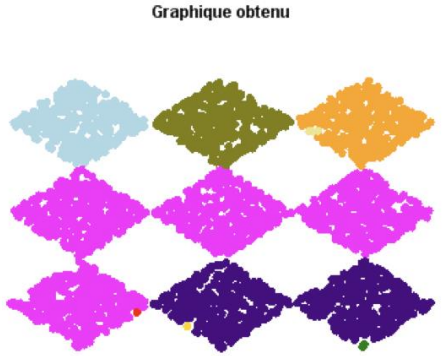
DBSCAN



Lien complet



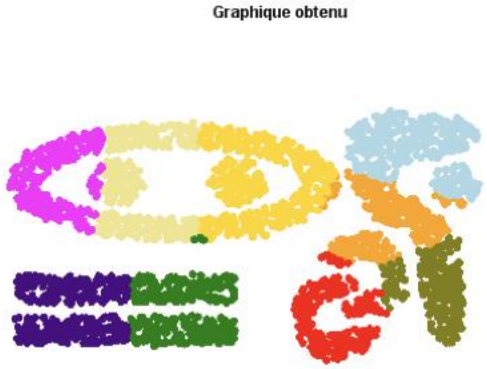
Lien simple



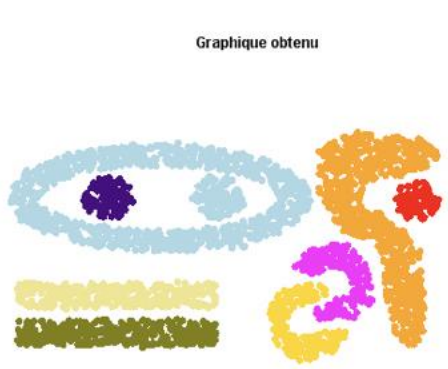
Comparaison sur une variété de données

Data1_4

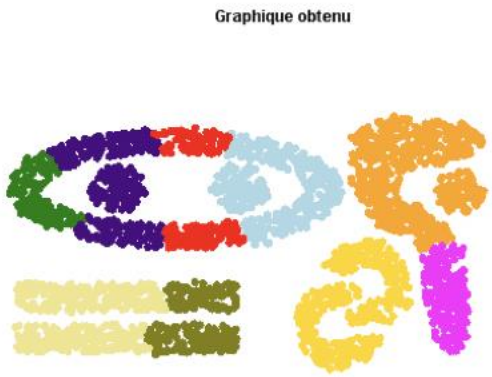
K-Mean



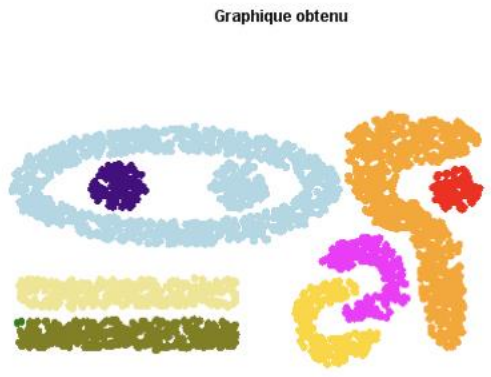
DBScan



Lien complet



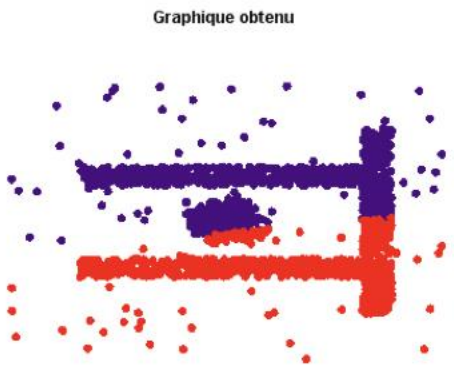
Lien simple



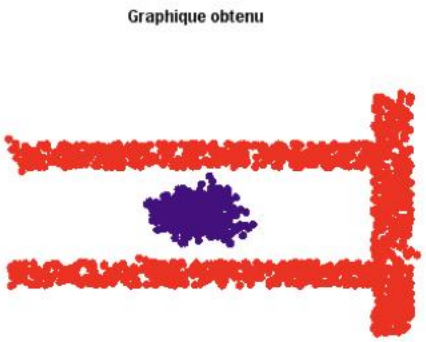
Comparaison sur une variété de données

Data1_5

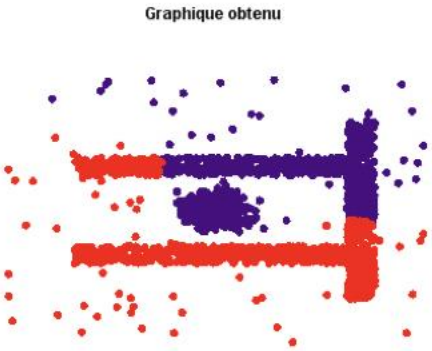
K-Mean



DBScan



Lien complet



Lien simple

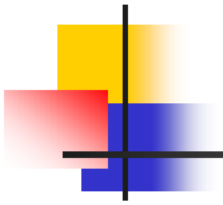
Défis

Deux images avec des représentations de pixels similaires peuvent appartenir à des clusters différents.

Défis

Deux images avec des représentations de pixels différentes peuvent appartenir au même Cluster.

Solution possible : “Deep clustering” pour la recherche des similarités sémantiques entre les données.



Plan

1 – Mise en contexte

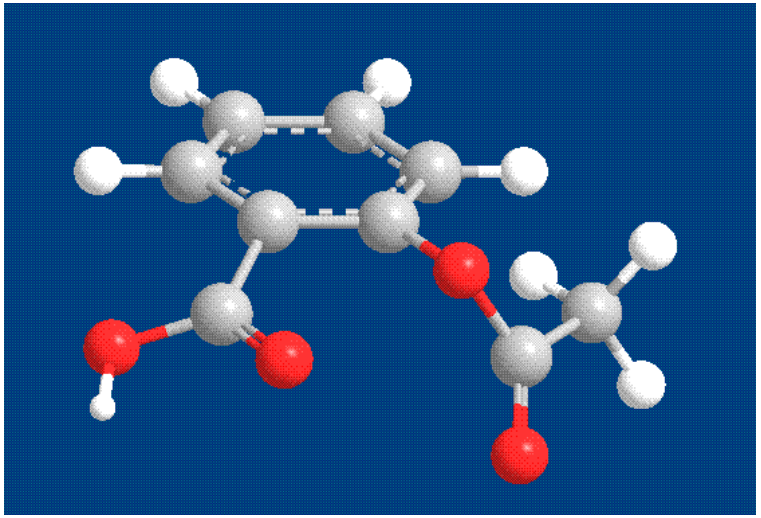
2 – Clustering avec K-means

3 – Clustering hiérarchique

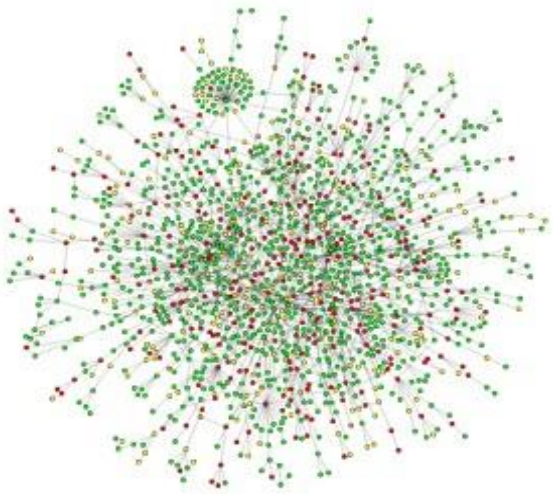
4 – Clustering basé sur la densité

5 – Clustering des graphes

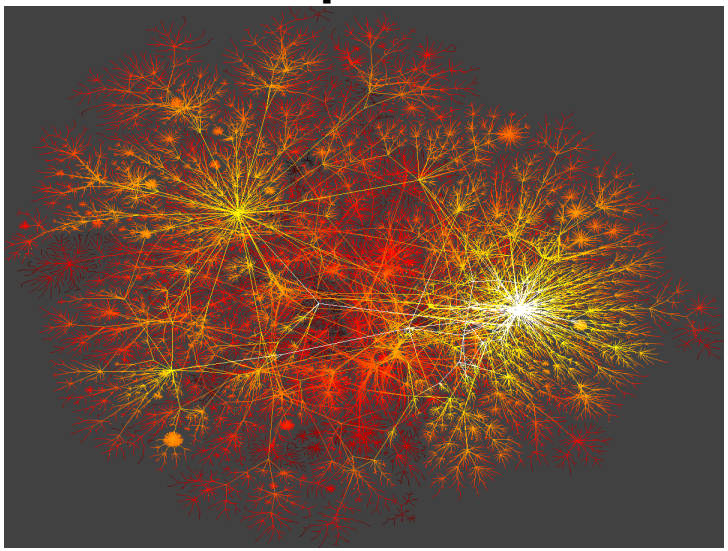
Représentation sous forme de graphe



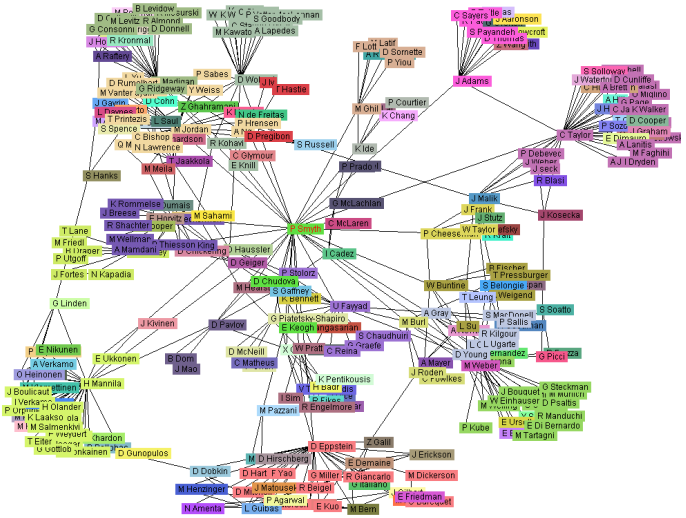
Aspirin



Yeast protein interaction network



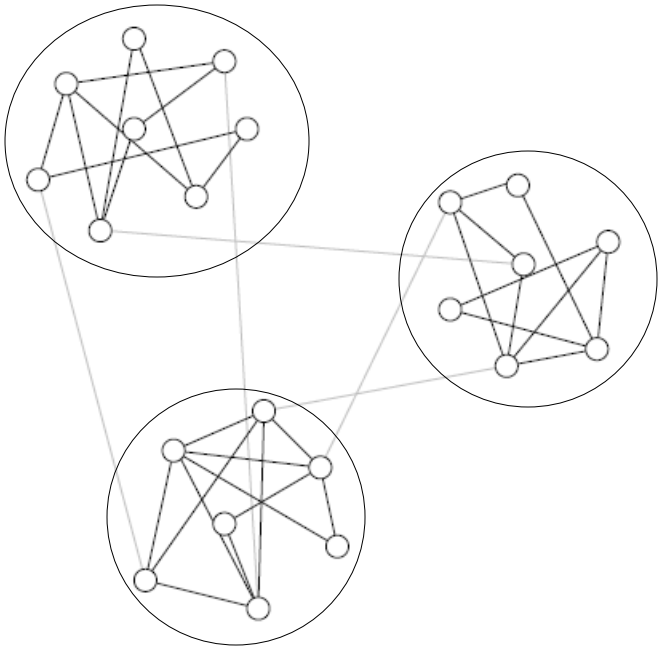
Internet



Co-author network

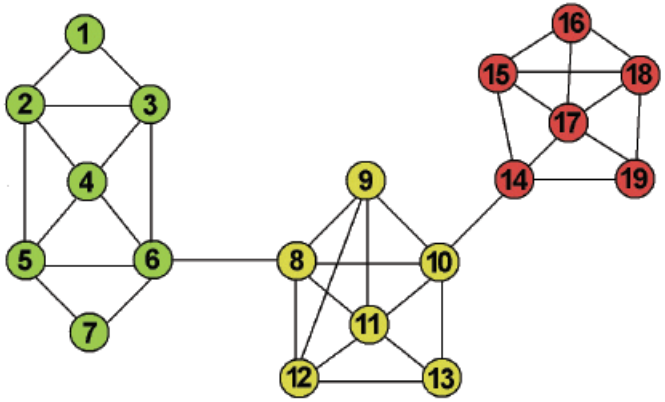
Clustering des graphes

- But : regrouper les nœuds densément connectés



- Une approche simple consiste à représenter un graphe comme une matrice binaire de telle sorte 1 indique la présence d'une connexion entre deux nœuds et 0 indique son absence. Puis appliquer des techniques de clustering sur la matrice pour grouper les vecteurs binaires.

Illustration



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

Méthodes ascendantes

- Une façon pour grouper les nœuds densément connectés dans un graphe consiste à mesurer la similarité entre chaque pair de nœuds.
- La distance de Czekanovski-Dice peut être utilisée pour mesurer la similarité entre deux nœuds dans un graphe. Cette distance entre deux nœuds N1 et N2 est définie comme suit :

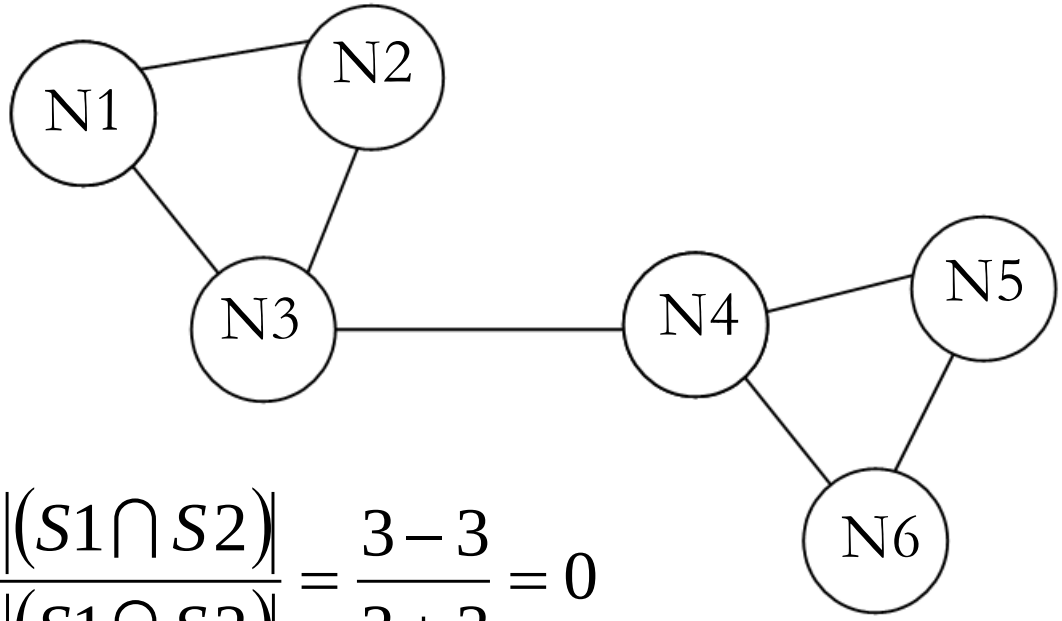
$$dist(N1, N2) = \frac{|(S1 \cup S2)| - |(S1 \cap S2)|}{|(S1 \cup S2)| + |(S1 \cap S2)|}$$

- S1 représente l'ensemble des nœuds qui sont connectés à N1 plus N1 lui-même. De même pour S2

➢ Plus la valeur de la distance est petite plus les nœuds sont fortement connectés entre eux.

La distance de Czekanovski-Dice

- **Exemple**



- dist(N1, N2) = ?

S1 = {N1, N2, N3}

S2 = {N2, N1, N3}

$$dist(N1,N2) = \frac{|(S1 \cup S2)| - |(S1 \cap S2)|}{|(S1 \cup S2)| + |(S1 \cap S2)|} = \frac{3-3}{3+3} = 0$$

- dist(N3, N4) = ?

S3 = {N3, N1, N2, N4}

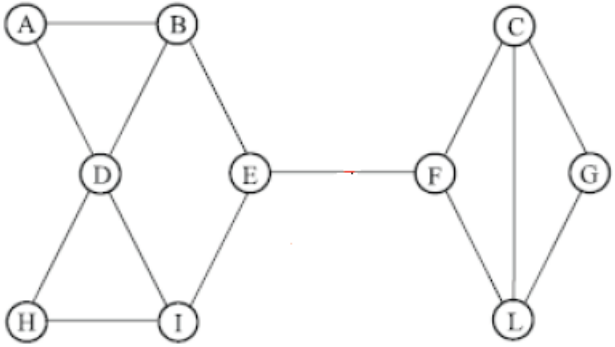
S4 = {N4, N3, N5, N6}

$$dist(N3,N4) = \frac{|(S3 \cup S4)| - |(S3 \cap S4)|}{|(S3 \cup S4)| + |(S3 \cap S4)|} = \frac{6-2}{6+2} = 0.5$$

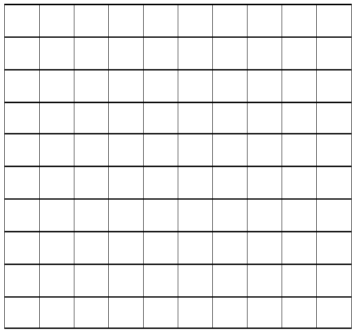
Algorithme

- Pour grouper les nœuds fortement connectés entre eux, on doit premièrement estimer une matrice de similarité en utilisant la distance de Czekanovski-Dice. Cette matrice est de dimension *n* x *n* (*n* est le nombre de nœuds).
- Une fois la matrice de similarité est estimée, les nœuds peuvent être regroupés à l'aide d'une procédure de classification hiérarchique ascendante.

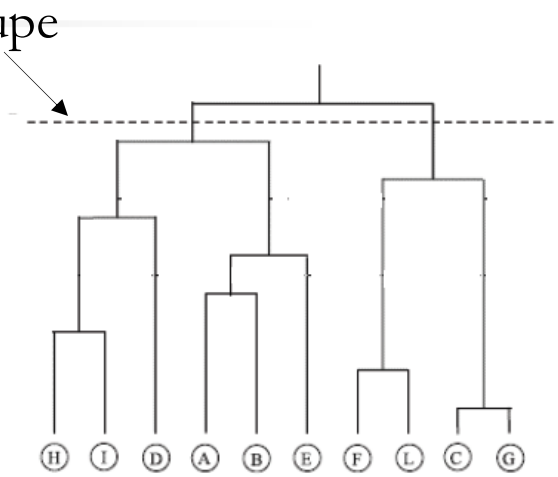
Illustration



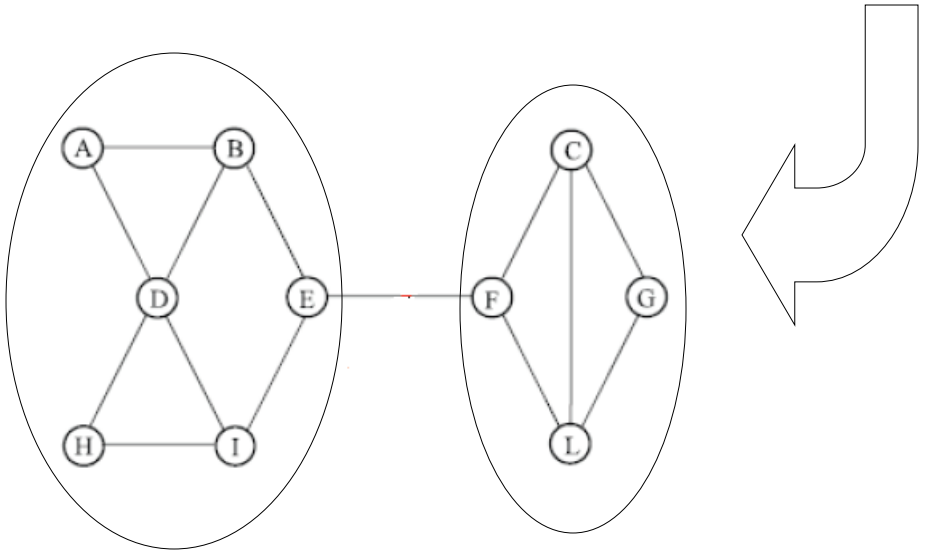
(a) Graphe



(b) Matrice de similarités



(c) Dendogramme



(d) Clustering

Référence

- Cette approche a été utilisée pour analyser les interactions entre les protéines. voir :

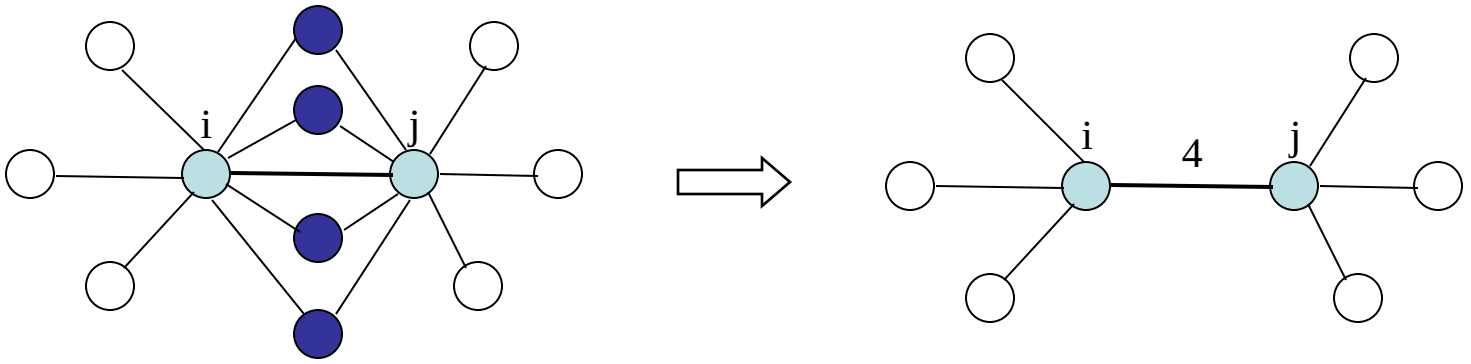
Brun, et al (2003).
Functional classification of proteins for the prediction of cellular function from a protein-protein interaction network.
Genome Biology, **5**, R6 1–13.

http://genomebiology.com/2003/5/1/R6

Une autre mesure de similarité

- Une autre mesure de similarité entre les nœuds d'un graphe consiste à assigner des poids entre un arc qui relie deux nœuds.
- Ce poids représente la similarité entre deux nœuds

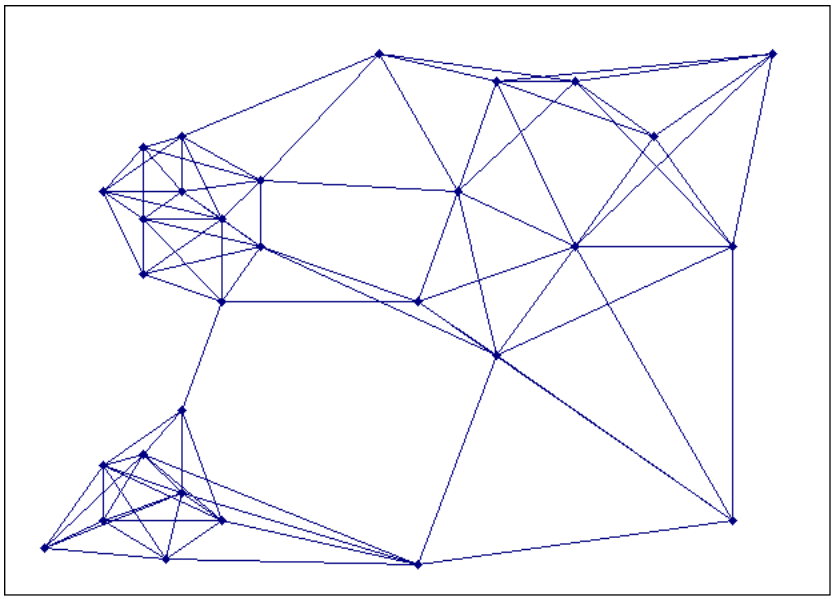
➢ Sa valeur est égale au nombre de nœuds voisins qui sont partagés et connectés aux deux nœuds qu'on veut mesurer leur similarité.



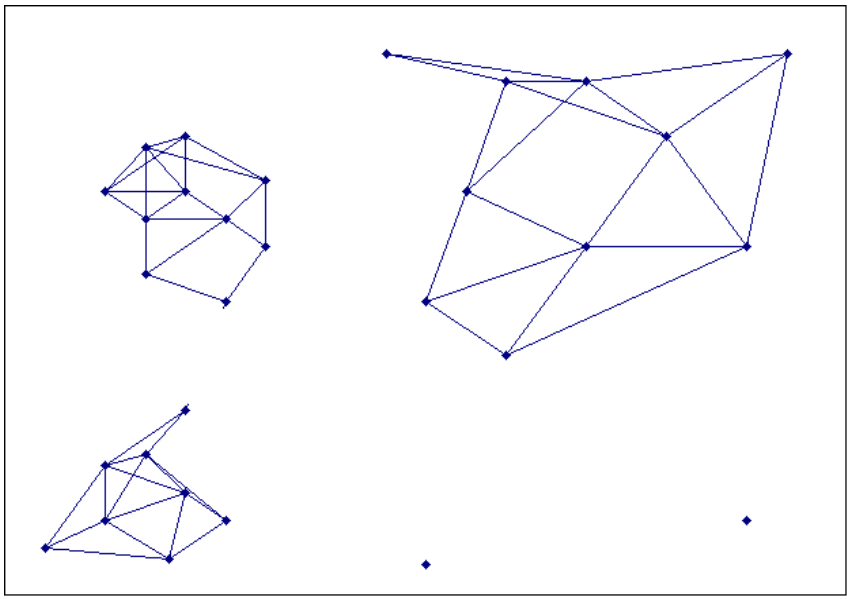
➢ Plus la valeur de la valeur du poids est grande plus les deux nœuds sont fortement connectés

Approche

- Avec cette approche, en fixant un seuil de similarité, il est possible de construire « Shared Near Neighbor Garph » qui relate des structures de clusters dans le graphe.



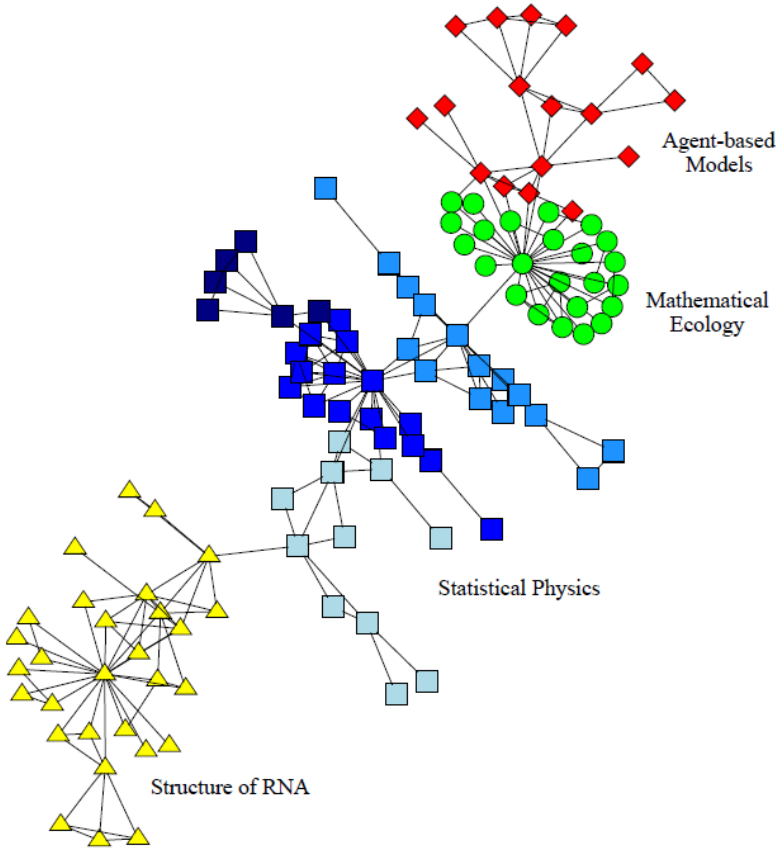
Graphe en entrée



Clustering

Exemple d'application

- The Santa Fe Institute collaboration network



Source : Grivan and Newman (2002), « Community structure in social and biological networks" PNAS, vol. 99, n0. 12

Exemple d'application

- **Enron email network** http://www.cs.cmu.edu/~enron/
- Juste une partie
  - 151 utilisateurs
  - Approximativement 200,399 messages