Clustering de Données : Approches par Partitionnement et Hiérarchiques

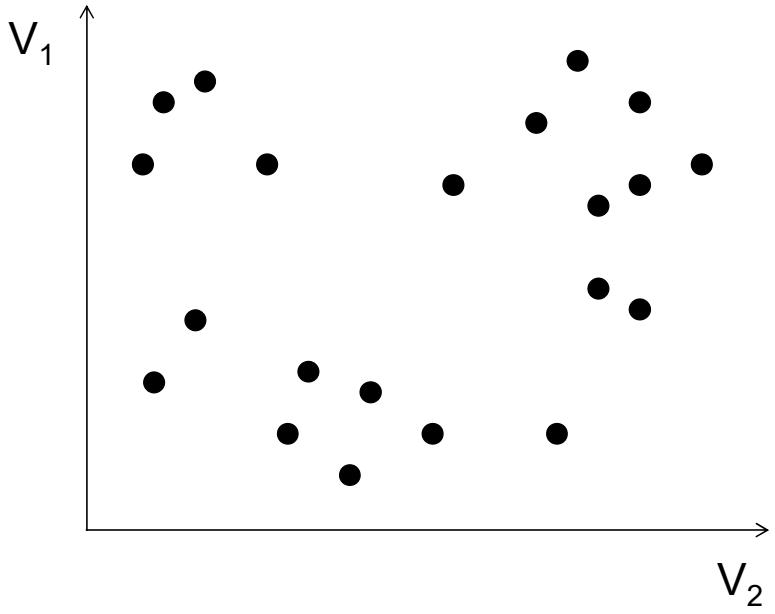
Nicolas PASQUIER
Laboratoire I3S (UMR-7271 UCA/CNRS)
Département Informatique
Université Côte d'Azur
http://www.i3s.unice.fr/~pasquier

Clustering par Partitionnement

- Principe : partitionner les instances de l'ensemble de données en K groupes où K est un paramètre défini par l'utilisateur
- Algorithme de référence : K-moyennes ou K-means (H. Steinhaus, 1957)
  1. Initialisation : choisir aléatoirement K instances en tant que centres initiaux des clusters, appelés centroïdes, (pour générer une partition initiale de l'ensemble de données)
  2. Boucle d'itérations :
     a. Calculer la distance entre chaque instance et les centroïdes de chaque cluster
     b. Si nécessaire, réaffectez chaque instance au cluster dont le centroïde est le plus proche
     c. Recalculer le centroïde de chaque cluster comme le barycentre du cluster
  3. Répétez la boucle d'itération si certaines instances ont été réaffectées

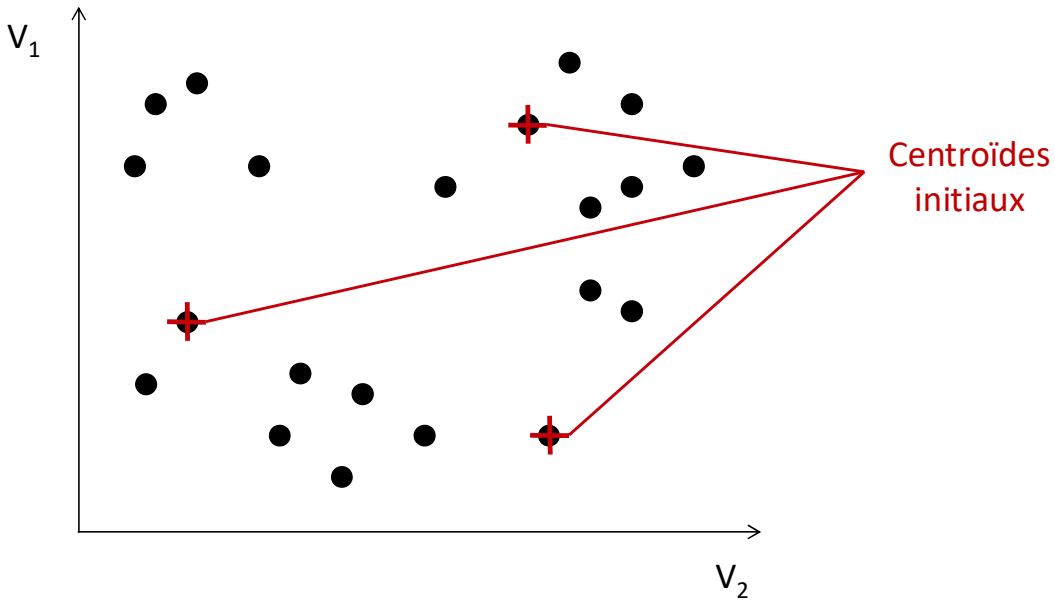
Algorithme des K-means : Exemple

- Espace des données bi-dimensionnel à partitionner en 3 clusters:
  Paramètre K = 3



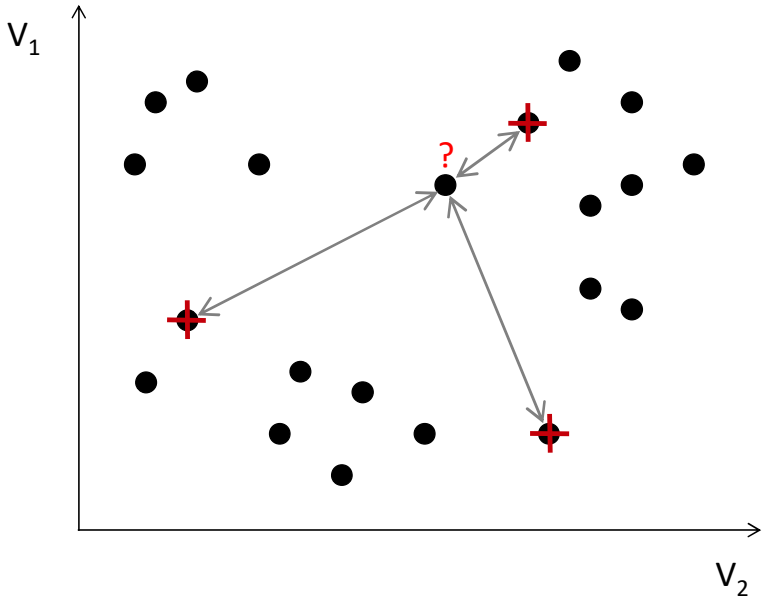
Algorithme des K-means : Exemple

- Initialisation aléatoire : 3 instances sont choisies aléatoirement comme centroïdes initiaux



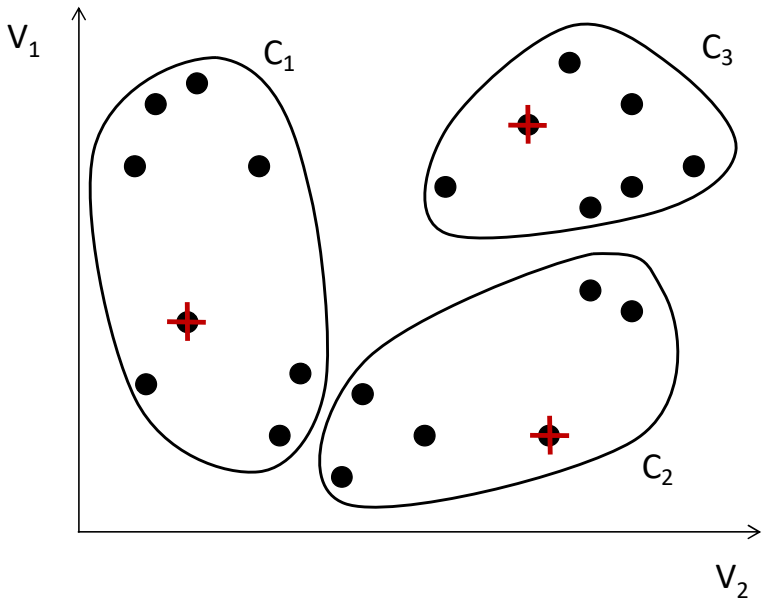
Algorithme des K-means : Exemple

- Calcul de mesure de distance : pour chaque instance non centroïde, sa distance par rapport à chaque centroïde est calculée



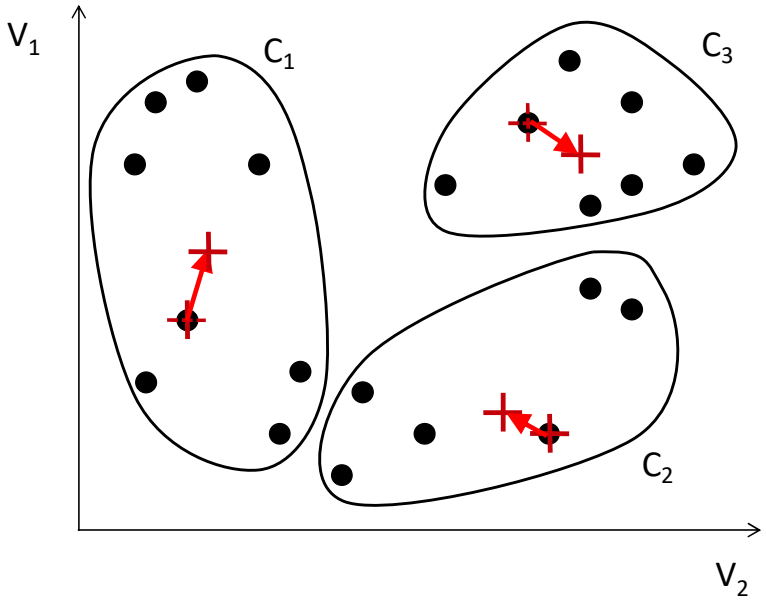
Algorithme des K-means : Exemple

- Affectation des instances : chaque instance non centroïde est assignée au cluster dont le centroïde est le plus proche



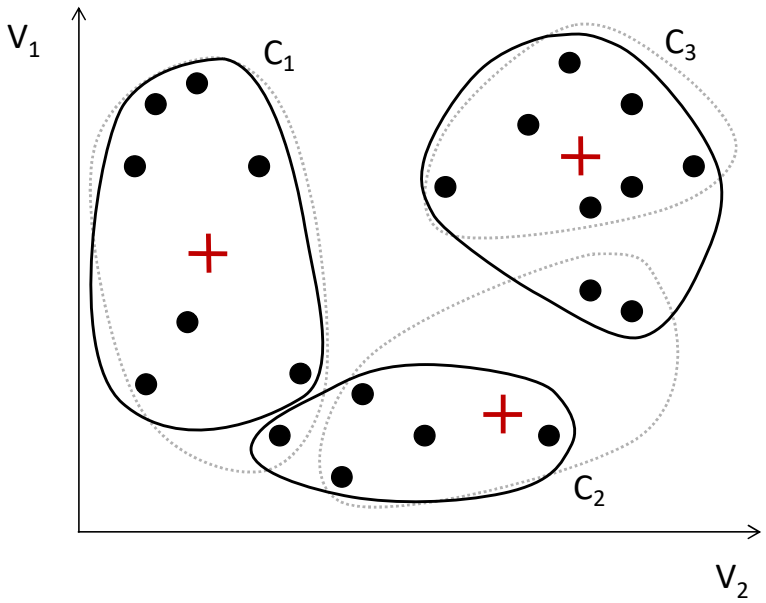
Algorithme des K-means : Exemple

- Mise à jour des centroïdes : le centroïde de chaque cluster est recalculé en tant que barycentre des instances du cluster



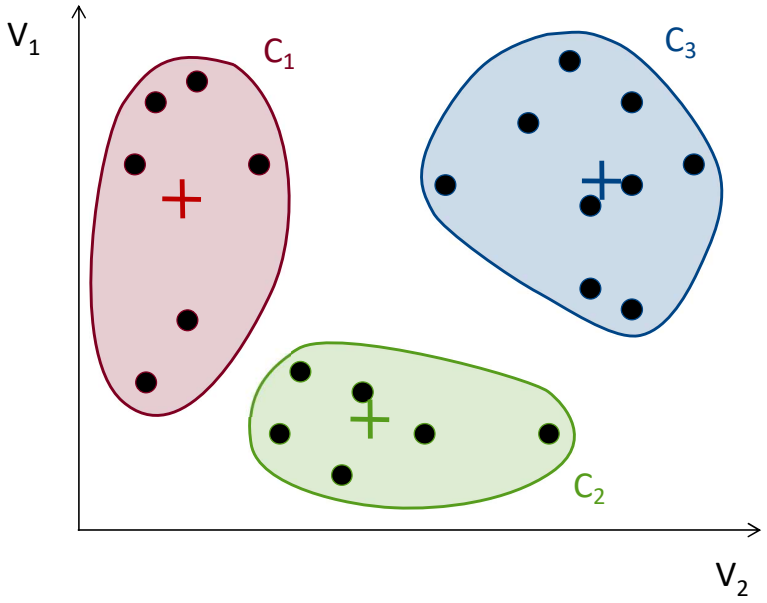
Algorithme des K-means : Exemple

- Mise à jour des clusters : les distances entre les instances et les centroïdes sont recalculées et une instances est réaffectée à un autre cluster si son centroïde est plus proche



Algorithme des K-means : Exemple

- Les itérations s'arrêtent lorsque la stabilité est atteinte, c'est-à-dire qu'aucune réaffectation d'instance n'est requise



Algorithme des K-means : Propriétés

- Forces
  - Efficace en terme de temps de calcul : complexité en temps O(K.N.T) pour K clusters, N instances et T itérations (généralement K «T« N)
  - Bonnes propriétés de passage à l'échelle par rapport à la taille de l'ensemble de données
- Faiblesses
  - Sensible aux données bruitées et valeurs aberrantes
  - Peut générer uniquement des clusters convexes
  - Nécessite que l'utilisateur définisse le nombre K de clusters
  - Peut traiter uniquement les variables pour lesquelles la moyenne est calculable (impossible de traiter les variables discrètes)
  - Non-déterministe : le résultat dépend du choix des centroïdes initiaux
    - Certaines implémentations tentent d'améliorer le choix des centres initiaux (hypothèses sur les distributions des valeurs)

Variantes de l'Algorithme des K-means

- Algorithme des K-medoïdes, ou *Partitioning Around Medoids* (L. Kaufman & P.J. Rousseeuw, 1987)
  - Le centre de chaque cluster est représenté par son instance la plus centrale, son médoïde, plutôt que par son barycentre
  - Permet le traitement de variables discrètes (catégorielles, binaires, etc.)
- Variante des Nuées dynamiques : chaque cluster est représentée par un noyau
  - Noyau : ensemble d'instances réelles les plus centrales
  - Plus robuste aux données bruitées et valeurs aberrantes
  - Approche plus coûteuse en temps car elle nécessite davantage de calculs, dépendamment de la taille du noyau

Clustering Hiérarchique

- Ces algorithmes créent une décomposition hiérarchique des différents clusters possibles
- Cette hiérarchie est représentée graphiquement dans une structure arborescente appelée dendrogramme
- Approches par agglomération (ex : AGNES, ROCK, UPGMA)
  - Initialisation : chaque instance est considérée comme un cluster
  - Itérations : regrouper successivement les clusters les plus proches
  - Arrêt : toutes les instances sont regroupées dans un unique cluster
- Approches par division (ex : DIANA, BIRCH, CURE)
  - Initialisation : un unique cluster regroupe toutes les instances
  - Itérations : divisions successives des clusters les moins compacts
  - Arrêt : condition d'arrêt atteinte (ex : mesure < seuil ou chaque instance constitue un cluster)

Clustering Hiérarchique : Exemple

- Considérons la matrice de distance ci-dessous générée à partir d'un ensemble de données constitué de 5 instances D = {a, b, c, d, e}

Matrice de distance

	a	b	c	d	e
a	0.00				
b	0.18	0.00			
c	0.39	0.32	0.00		
d	0.43	0.34	0.25	0.00	
e	0.39	0.41	0.27	0.21	0.00

- Les clusters sont construits en utilisant l'approche par agglomération

Clustering Hiérarchique : Exemple

- Initialisation : chaque instance constitue un cluster

Matrice de distance

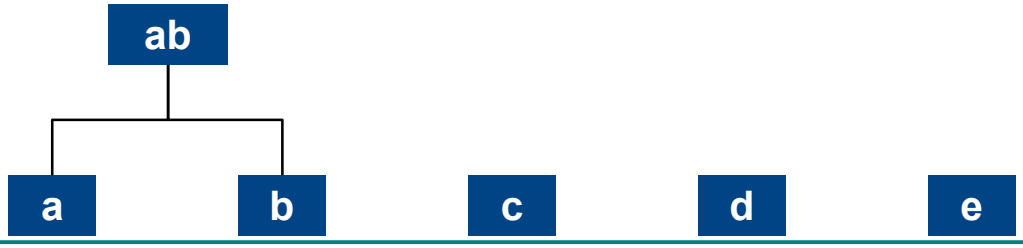
	a	b	c	d	e
a	0.00				
b	0.18	0.00			
c	0.39	0.32	0.00		
d	0.43	0.34	0.25	0.00	
e	0.39	0.41	0.27	0.21	0.00

Clustering Hiérarchique : Exemple

- Itération : fusionner les deux clusters les plus proches
- Ils sont identifiés par la distance minimale dans la matrice de distances
- La hauteur de l'arc les reliant est proportionnelle à cette distance

Matrice de distance

	a	b	c	d	e
a	0.00				
b	**0.18**	0.00			
c	0.39	0.32	0.00		
d	0.43	0.34	0.25	0.00	
e	0.39	0.41	0.27	0.21	0.00



Clustering Hiérarchique : Exemple

- Itération : fusionner les deux clusters les plus proches
- Les distances d(ab,c), d(ab,d), d(ab,e), d(c,d), d(c,e) et d(d,e) sont comparées à partir de la matrice de distance

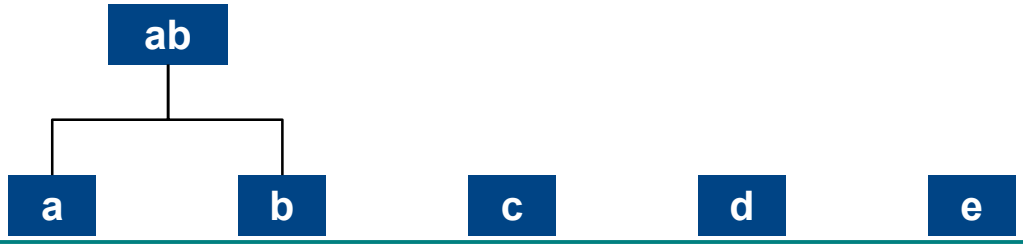
Matrice de distance

	a	b	c	d	e
a	0.00				
b	0.18	0.00			
c	0.39	0.32	0.00		
d	0.43	0.34	0.25	0.00	
e	0.39	0.41	0.27	0.21	0.00

d(ab,c) = avg(0.39,0.32) = 0.355

d(ab,d) = avg(0.43,0.34) = 0.385

d(ab,e) = avg(0.39,0.41) = 0.40

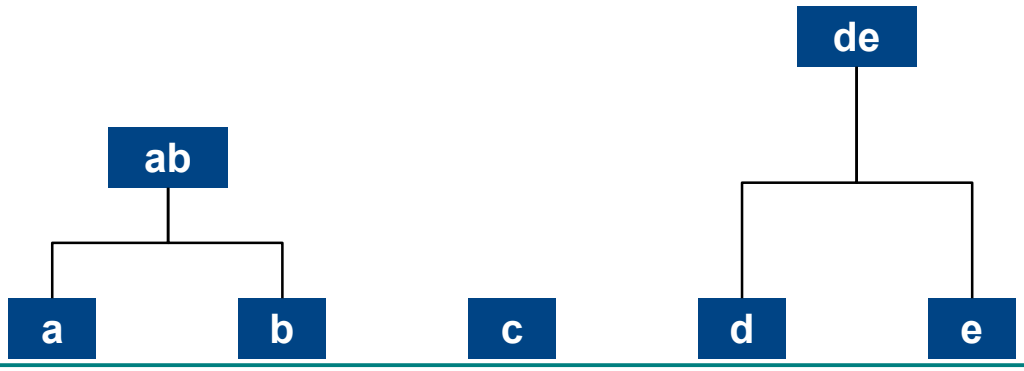


Clustering Hiérarchique : Exemple

- Itération : fusionner les deux clusters les plus proches
- Les clusters {d} et {e} sont fusionnés car la distance d(d,e) est la distance minimale

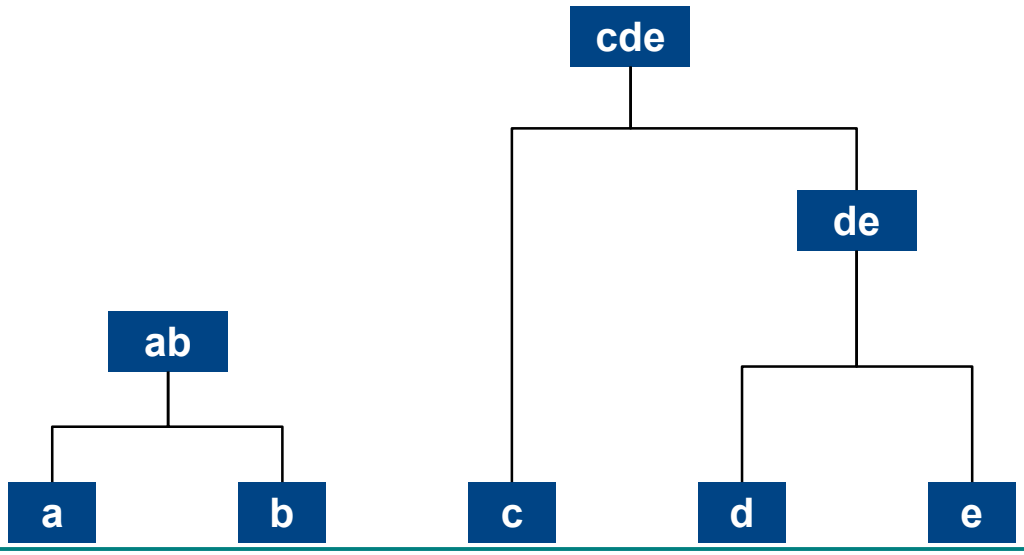
Matrice de distance

	a	b	c	d	e
a	0.00				
b	0.18	0.00			
c	0.39	0.32	0.00		
d	0.43	0.34	0.25	0.00	
e	0.39	0.41	0.27	**0.21**	0.00



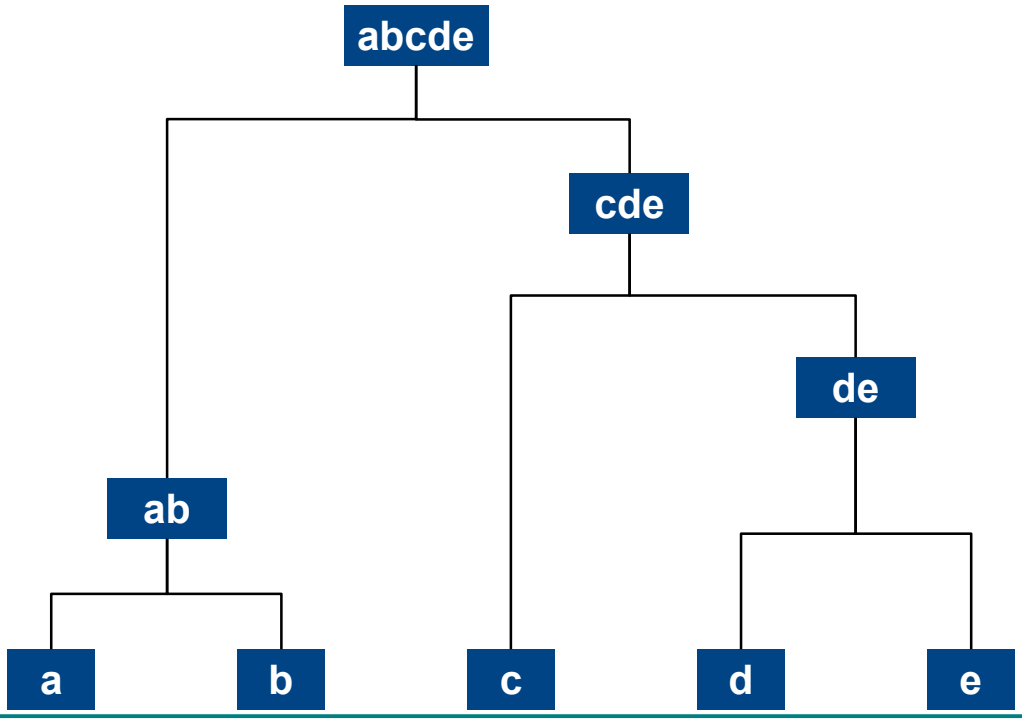
Clustering Hiérarchique : Exemple

- Itération : fusionner les deux clusters les plus proches
- Les distances d(ab,c), d(ab,de) et d(c,de) sont comparées à partir de la matrice de distance : les clusters {c} et {de} sont fusionnés



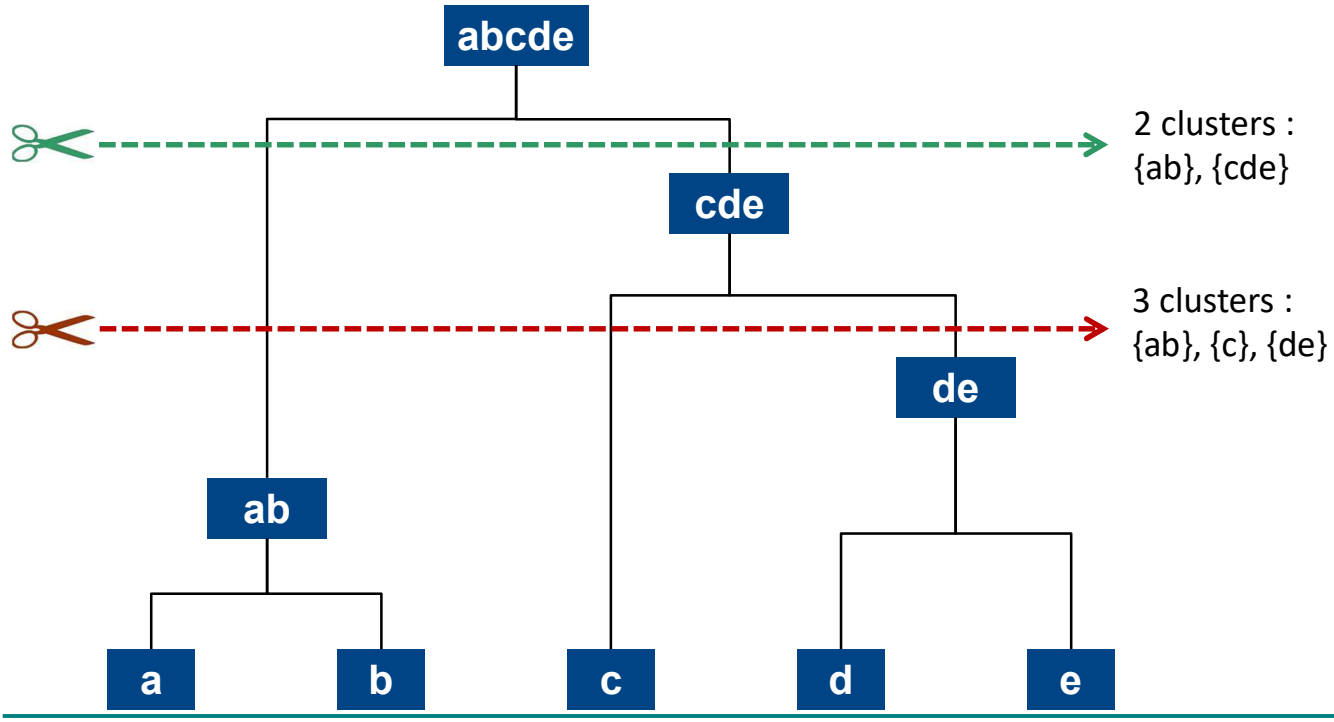
Clustering Hiérarchique : Exemple

- Arrêt : les clusters {ab} et {cde} sont fusionnés dans un unique cluster {abcde}



Clustering Hiérarchique : Exemple

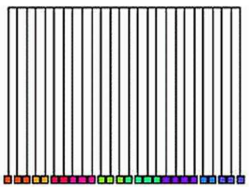
- Le nombre de clusters générés dépend de la hauteur à laquelle on « coupe » le dendrogramme



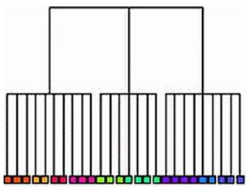
Interprétation du Dendrogramme

- Le dendrogramme peut identifier des regroupements (clusters) à différent niveaux de granularité (précision)
- Ex : clustering de groupes d'utilisateurs de réseaux sociaux

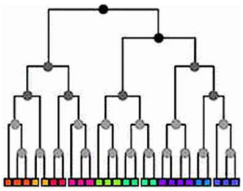
Dendrogramme



a) Un cluster

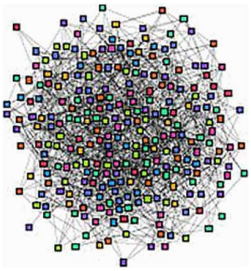


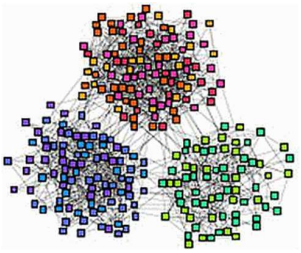
b) Trois clusters

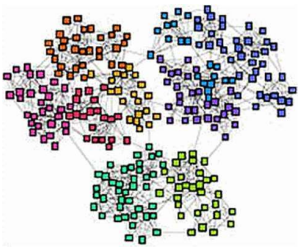


c) Neuf clusters

Graphes des utilisateurs et leurs relations







- Les niveaux représentent différent types de relations (proximité)

Clustering Hiérarchique : Propriétés

- Forces
  - Ne nécessite pas de définir a priori le nombre K de clusters
  - Fournit une décomposition hiérarchique des clusters
  - Fournit une représentation graphique arborescente qui représente également les distances entre clusters
- Faiblesses
  - Complexité en temps de $O(N^2.\log(N))$ pour N instances
  - Mauvaises propriétés de mise à l'échelle par rapport à la taille de l'ensemble de données
  - Le regroupement d'instances dans les clusters est définitif : les décisions erronées sont impossibles à corriger plus tard
  - Les clusters ont tendance à être de même taille