Apprentissage non supervisé : Méthodes de Clustering

M.-J. Huguet

INSA TOULOUSE

https://homepages.laas.fr/huguet

2018-2019

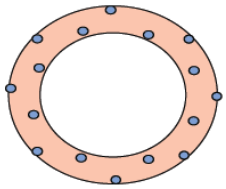
LAAS CNRS

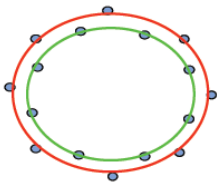
Plan

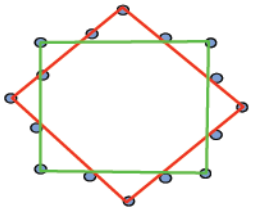
1. Contexte : l'Intelligence Artificielle
2. Contexte : l'apprentissage automatique
3. **Problème de clustering**
4. **Premières méthodes**
5. **Méthodes basées voisinage (densité) et basées graphes**
6. **Boite à outils**
7. Fouille de données
8. Réduction de dimensions (Analyse en Composantes principales)

Rappel : Types d'apprentissage

- **Différents types d'apprentissage**
  - **Apprentissage non supervisé**
    - Le système ne dispose que d'exemples :
      - Données X sans étiquette
    - Nombre et nature des classes inconnu
  - Rechercher une structure dans les données
    - Partitionner les exemples en clusters/classes
      - Clustering (segmentation, partitionnement)
      - Partitionner les exemples en clusters/classes
    - **Un bon clustering ?**
      - Homogènes : les éléments d'un même cluster sont similaires
      - Séparés : les éléments de différents clusters sont différents







Plan – section 3

3. **Problème de clustering**
   1. Partition d'un ensemble
   2. Position du problème de clustering
   3. Distance
   4. Type de méthodes de partitionnement
   5. Evaluation d'une solution de partitionnement
4. **Premières méthodes**
5. **Méthodes basées voisinage (densité) et basées graphes**
6. **Boite à outils**

Partition d'un ensemble

- **Le problème :**
  - Décomposer un ensemble X en sous-ensembles non vides tel que chaque élément $x \in X$ se retrouve dans un et un seul sous ensemble

  - Soit P une famille d'ensembles : P est une **partition** de X ssi
    - L'ensemble vide n'est pas dans P : $\emptyset \notin P$
    - L'union des ensembles de P vaut X : $\bigcup_{A \in P} A = X$
    - Les ensembles de P sont deux à 2 disjoints : $\forall A, B \in P : A \neq B \Rightarrow A \cap P = \emptyset$

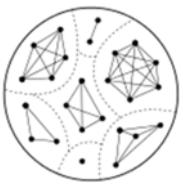
    - **Exemple**
      - $X = \{a, b, c\}$. Il existe 5 partitions :
        - $\{\{a\},\{b\},\{c\}\}$; $\{\{a,b\},\{c\}\}$; $\{\{a,c\},\{b\}\}$; $\{\{b,c\},\{a\}\}$; $\{\{a,b,c\}\}$
      - Ne sont pas des partitions de X :
        - $\{\{\ \},\{a,b\},\{c\}\}$; $\{\{a,b\},\{b,c\}\}$; $\{\{a\},\{b\}\}$;

Partitions et relations d'équivalence

- **Relations d'équivalence**
  - Regrouper des éléments d'un ensemble (à partir d'une relation binaire)
    - Sont considérés comme similaires par rapport à une propriété (ex : couleur)
  - Propriétés : réflexive, symétrique et transitive
    - Ex : relation « est égal à »

  - Classe d'équivalence
    - Avec une relation d'équivalence ➔ éléments regroupés dans un ensemble de classes d'équivalences
    - L'ensemble des classes d'équivalence est une partition

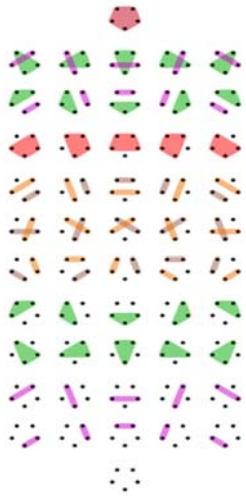
  - Représentation graphe non orienté d'une relation d'équivalence
    - Classes d'équivalence : composantes connexe formées de cliques



  - A toute partition on peut associer une relation d'équivalence

Dénombrer le nombre de partitions

- **Nombre de partitions d'un ensemble en K sous-ensembles :**
  - *Ex : pour un ensemble $X = \{a, b, c\}$ de taille 3 : il existe 5 partitions différentes*
    - $\{\{a\},\{b\},\{c\}\}$; $\{\{a,b\},\{c\}\}$; $\{\{a,c\},\{b\}\}$; $\{\{b,c\},\{a\}\}$; $\{\{a,b,c\}\}$
    - k=3; k=2 ; k=1

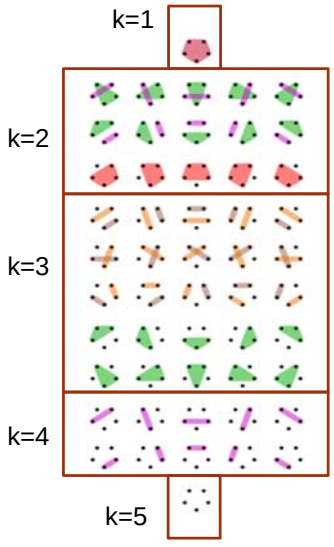


  - Nombre de Stirling : $S(n,k)$
    - **Equations de récurrence :**
      - $S(n,k) = S(n-1,k-1) + k \times S(n-1,k)$
      - $S(0,0) = 1$ et $\forall n > 0, S(n,0) = S(0,n) = 0$
    - Formulation explicite : $S(n,k) = \frac{1}{k!}\sum_{j=0}^{k}(-1)^{k-j} {C_j}^k j^n$
    - Où ${C_j}^k$ est le nombre de combinaisons de j parmi k
- **Nombre total de partitions**
  - Nombre de Bell : $B(n) = \sum_{k=1}^{n} S(n,k)$

Dénombrer le nombre de partitions

- Nombre de Stirling $S(n,k)$ et nombre de Bell $B(n) = \sum_{k=1}^{n} S(n,k)$

		k										
		1	2	3	**4**	5	6	6	8	9	10	Nb Bell
n	1	1										1
	2	1	1									2
	3	1	3	1								5
	4	1	7	6	1							15
	5	**1**	**15**	**25**	**10**	**1**						**52**
	6	1	31	**90**	**65**	15	1					203
	7	1	63	301	**350**	140	21	1				877
	8	1	127	966	1701	1050	266	27	1			4139
	9	1	255	3025	7770	6951	2646	428	35	1		21112
	10	1	511	9330	34105	42525	22827	5214	708	44	1	115326



Plan

3. **Problème de clustering**
   1. Partition d'un ensemble
   2. Position du problème de clustering
   3. Distance
   4. Type de méthodes de partitionnement
   5. Evaluation d'une solution de partitionnement
4. **Premières méthodes**
5. **Méthodes basées voisinage (densité) et basées graphes**
6. **Boite à outils**

Caractérisation du clustering

- **Cluster = un regroupement de données**
  - Regrouper des données proches
  - Eloigner des données différentes

Données

Clustering

Clustering

Clustering

- Qu'est-ce qu'un bon clustering ?
- Problème mal posé ….
- L'objectif dépend du problème considéré

Images N. Baskiotis, LIP6 - 2017

Exemples de clustering

- **Quelques applications**
  - Identifier des communautés dans des réseaux sociaux
  - Identifier des clients avec un profil similaire
  - Analyser des logs d'applications
  - Analyser des textes, des emails
  - Segmenter des images
  - ….

Position du problème (1)

- **Définition**
  - Un ensemble $X = \{x_i\}$ de n exemples / observations
  - Une observation
    - est composée de d attributs

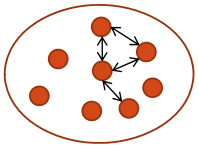
$$x_i = \begin{pmatrix} x_{i,1} \\ x_{i,2} \\ \vdots \\ x_{i,d} \end{pmatrix}$$

  - Déterminer K clusters tel que
    - Chaque cluster regroupe des observations similaires

- **Définir la similarité**
- **Déterminer le nombre de clusters**
- **Evaluer un résultat de clustering**

Position du problème (2)

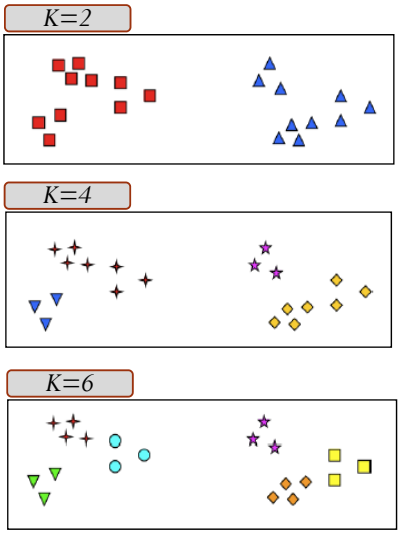
- **Distance et similarité**
  - Distance entre points $d(x, y)$: mesure de di-similarité
    - **Minimiser distance intra-cluster**
      - Plus la distance est élevée moins les points sont similaires
      - Ex : similarité : $sim(x, y) = \frac{1}{1+d(x,y)}$
      - dépend de la nature des données



  - Déterminer une **partition** $\pi : X \rightarrow \{C_1, \dots C_K\}$ de **taille *K*** telle que :
    - $\bigcup_{i=1}^{K} C_i = X$, et $C_i \bigcap_{i \neq j} C_j = \emptyset$
    - pour chaque cluster/classe C_i, $\forall\, x, y \in C_i$ et $z \notin C_i$: on veut vérifier
      - $sim(x, y) > sim(x, z)$ et $sim(x, y) > sim(y, z)$
    - **Partition optimale** π^*: $argmin_\pi f(\pi)$ où f dépend de la fonction de similarité

Position du problème (3)

- **Difficulté algorithmique**
  - **Partition optimale** : π^*: $argmin_\pi f(\pi)$
  - Nombre partitions possibles en fonction de X et de K ➔ nombre de Bell
  - **Valeur de *K***
    - Fixée (méthodes paramétriques)
    - Non fixée (méthodes non paramétriques)
  - **Prendre en compte la distance entre clusters**
    - **Maximiser** distance inter-cluster
    - Plus la distance est élevée plus les clusters sont séparés



Plan

3. **Problème de clustering**
   1. Partition d'un ensemble
   2. Position du problème de clustering
   3. Distance
   4. Type de méthodes de partitionnement
   5. Evaluation d'une solution de partitionnement
4. **Premières méthodes**
5. **Méthodes basées voisinage (densité) et basées graphes**
6. **Boite à outils**

Distances (1)

- **Distance : une fonction d : $\mathbb{R} \rightarrow \mathbb{R}^+$ vérifiant :**
  - Symétrie : $d(x,y) = d(y,x)$
  - Séparation : $d(x,y) = 0 \iff x = y$
  - Inégalité triangulaire : $d(x,y) \leq d(x,z) + d(z,y) = d(y,x)$

- Distance de Minkowski ou Norme $\boldsymbol{L_q}$
  - $d(x_1, x_2) = ||x_2 - x_1||_q = \sqrt[q]{\sum_{j=1}^{d} |x_{1,j} - x_{2,j}|^q}$
  - Si $q = 2$, distance euclidienne
    - $d(x_1, x_2) = ||x_2 - x_1|| = \sqrt{\sum_{j=1}^{d} |x_{1,j} - x_{2,j}|^2}$
  - Si $q = 1$, distance de Manhattan

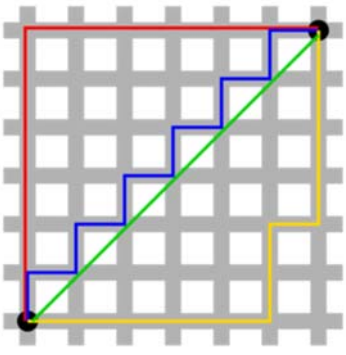


Image:Wikipedia

Distances (2)

- **Distance de Hamming**
  - Mesurer différence entre deux séquences de symboles
    - Traitement du signal
  - Soit x_i et y_i deux observations de dimension d
  - Hamming : $h(x_i, y_i) = Card(\{j : x_{ij} \neq y_{ij}\})$

$$x_i = \begin{pmatrix} x_{i,1} \\ x_{i,2} \\ \vdots \\ x_{i,d} \end{pmatrix}$$

  - Exemple
    - Entre 1011101 et 1001001 ➔ distance de Hamming = 2
    - Entre 2143896 et 2233796 ➔ distance de Hamming = 3
    - Entre ramer et cases ➔ distance de Hamming = 3

Distances (3)

- **Distance de Levenshtein (distance d'édition)**
  - Mesurer la différence entre deux chaines de caractères
    - Nombre d'opérations élémentaires (insérer/supprimer/remplacer) pour passer d'une chaine source à une chaine destination
      - Passer de "a " vers "ab" : distance = 1 (insérer 'b')
  - Autres : compter des n-grammes
    - Sous séquences de longueur n présentes dans une séquence
    - Comparer des séquences à partir des n-grammes communs

Plan

3. **Problème de clustering**
   1. Partition d'un ensemble
   2. Position du problème de clustering
   3. Distance
   4. Type de méthodes de partitionnement
   5. Evaluation d'une solution de partitionnement
4. **Premières méthodes**
5. **Méthodes basées voisinage (densité) et basées graphes**
6. **Boite à outils**

Méthodes de partitionnement (1)

- **Principe**
  - Initialisation : création d'une partition de K clusters
  - Itérations : déplacer un objet entre clusters pour optimiser la fonction objectif
- **Méthodes exactes**
  - Enumération et évaluation de toutes les partitions possibles
- **Méthodes approchées ou heuristiques**
  - Très nombreuses dans la littérature
  - Méthodes générales ou spécifiques pour un domaine d'application
  - Exploitent une fonction objectif +/- complexe

Méthodes de partitionnement (2)

- **Idéalement**
  - Prendre en compte différents types de données
    - numériques, symboliques, …
  - Générer des formes quelconques de clusters
    - Pas seulement des formes convexes
  - Facilité de paramétrages
  - Insensibilité à l'ordre de traitement des données
  - Robustesse / anomalies et aux bruits
  - Passage à l'échelle (volume données et dimension des données)
  - Résultats cohérents / utilisateurs
  - Enrichissement par des contraintes / clusters

Plan

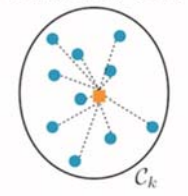
3. **Problème de clustering**
   1. Partition d'un ensemble
   2. Position du problème de clustering
   3. Distance
   4. Type de méthodes de partitionnement
   5. Evaluation d'une solution de partitionnement
4. **Premières méthodes**
5. **Méthodes basées voisinage (densité) et basées graphes**
6. **Boite à outils**

Evaluation d'un clustering

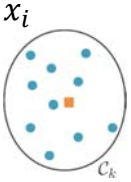
- **Forme des clusters**
  - Évaluation de la **qualité** des clusters / distance
  - Clusters resserrés sur eux-mêmes et éloignés entre eux
- **Stabilité des clusters**
  - Insensibilité à l'ordre des traitement des données
  - Les mêmes points sont-ils toujours dans le même cluster ?
  - Aide pour fixer le nombre de clusters
- **Cohérence / expertise**
  - Évaluation par expert humain …

Qualité d'un clustering (1)

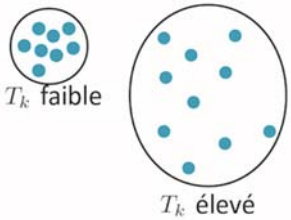
- **Indice de Davies Bouldin (DB)**
  - Combinaison mesures d'homogénéité et de séparation
  - **Cohésion / homogénéité** = qualité intra-cluster (**Diamètre moyen**)
    - pour un cluster k : moyenne des distances entre chaque point et le centre μ_k :
      - $H_k = \frac{1}{n_k} \sum_{x \in C_k} d(x, \mu_k)$



Centre d'un cluster :
$\mu_k = \frac{1}{n_k} \sum_{x_i \in C_k} x_i$
avec $n_k = |C_k|$

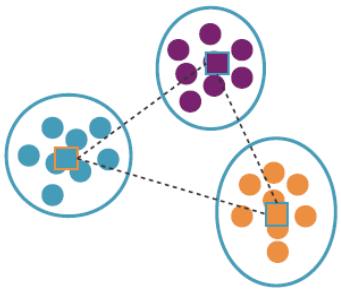


- Un cluster k homogène a une valeur H_k faible



Qualité d'un clustering (2)

- **Indice de Davies Bouldin (DB)**
  - Combinaison mesures d'homogénéité et de séparation
  - **Séparation** = qualité inter-cluster
    - Pour 2 clusters k et l : distance entre leurs centres
    - $S_{k,l} = d(\mu_k, \mu_l)$



  - **Indice DB**
    - $DB = \frac{1}{K}\sum_{k=1}^{K} DB_k$, avec $DB_k = max_{l \neq k}(\frac{H_k + H_l}{S_{k,l}})$
    - Valeur faible si les clusters sont homogènes (numérateur petit) et s'ils sont bien séparés (dénominateur grand)
    - Minimiser DB ➔ aide pour déterminer le nombre de clusters

Qualité d'un clustering (3)

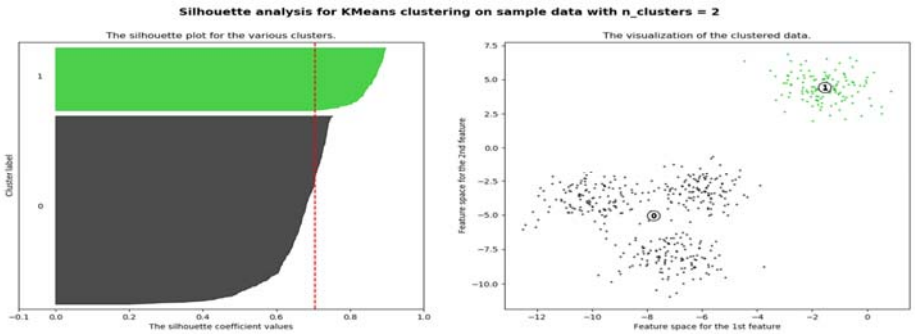
- **Coefficient de silhouette**
  - Combinaison de deux mesures
  - **Cohésion (proximité)** : appartenance au « bon » cluster
    - pour chaque point $x \in C_k$: est-il proche des points du cluster auquel il appartient ?
      - distance moyenne aux autres points du même cluster
      - $a(x) = \frac{1}{n_k - 1}\sum_{y \in C_k, y \neq x} d(x, y)$
  - **Séparation** : éloignement des autres clusters
    - pour chaque point $x \in C_k$: est-il loin des points des autres clusters C_l ?
      - Distance moyenne **minimale** par rapport aux points des clusters C_l c'est à dire au cluster le plus proche
      - $b(x) = min_{l \neq k} \frac{1}{n_l}\sum_{y \in C_l} d(x, y)$

Qualité d'un clustering (4)

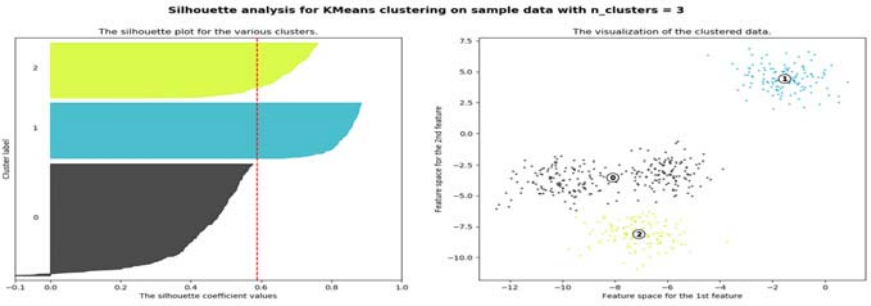
- **Coefficient de silhouette**
  - Combinaison de deux mesures
  - **Cohésion** : appartenance au « bon » cluster
    - $a(x) = \frac{1}{n_k - 1} \sum_{y \in C_k, y \neq x} d(x, y)$
  - **Séparation** : éloignement des autres clusters
    - $b(x) = min_{l \neq k} \frac{1}{n_l} \sum_{y \in C_l} d(x, y)$
  - **Silhouette** : $s(x) = \frac{b(x) - a(x)}{\max(a(x), b(x))}$; compris dans [-1, 1])
    - Si le point x est dans le bon cluster : $a(x) < b(x)$ et $s(x)$ ➔ 1
  - Pour tous les points : $S = \frac{1}{n} \sum s(x)$
  - Minimiser S ➔ aide pour déterminer le nombre de clusters

Exemple (scikitlearn - silhouette)

- 2 clusters, silhouette (moyenne) = 0,705

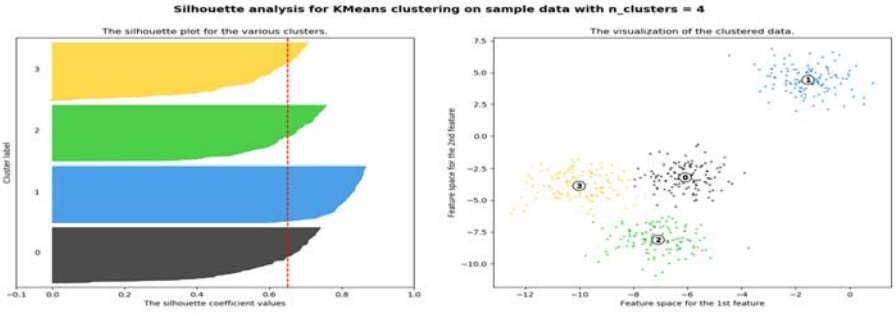


- 3 clusters, silhouette (moyenne) = 0,588



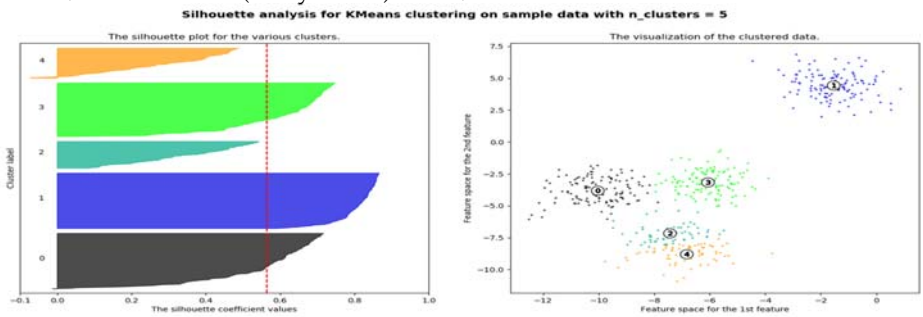
Exemple (scikitlearn - silhouette)

- 4 clusters, silhouette (moyenne) = 0,650



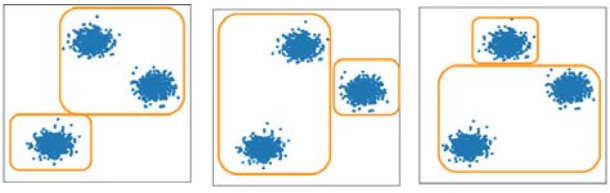
- 5 clusters, silhouette (moyenne) = 0,563

➔ **Choix K=2 ou K=4**

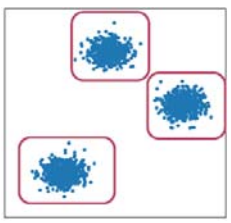


Stabilité d'un clustering

- **Problème :**
  - Méthodes non déterministes : résultats différents si plusieurs exécutions
    - lancer la méthode plusieurs fois avec initialisation différente, avec des sous-ensembles différents,
    - Est-ce que les points sont regroupés de manière similaire ?
    - Ex : problème de stabilité pour K=2



    - Stable pour K=3



    - **Indice de Rand** : comparer 2 solution de clustering en ignorant les permutations
      - Nombre de paires dans le même cluster / nb de paires dans des clusters différents

Evaluation d'un clustering

- **Forme des clusters**
- **Stabilité des clusters**
  - Différentes mesures …
  - Dépend des données, des distances, …
  - Avoir l'esprit critique sur les résultats et se documenter
- **Cohérence / expertise**
  - Évaluation par expert humain …
  - Évaluation « manuelle »
  - Vérification sur un sous-ensemble de données

Plan

1. Contexte : l'Intelligence Artificielle
2. Contexte : l'apprentissage automatique
3. Problème de clustering
4. **Premières méthodes**
5. **Méthodes basées voisinage (densité) et basées graphes**
6. **Boite à outils**
7. Fouille de données
8. Réduction de dimensions (Analyse en Composantes principales)

Plan – section 4

3. Problème de clustering
4. **Premières méthodes**
   1. Méthode par partitionnement : k-means
   2. Méthodes hiérarchiques : ascendant et descendant
5. Méthodes basées voisinage (densité) et basées graphes
6. Boite à outils

Méthode k-Means (1)

- **Principe**
  - Le nombre de clusters K est fixé
  - Pour chaque cluster : son centre de gravité
  - Placer toutes les données dans les clusters / centres de gravité courants
  - Mettre à jour les centres au fur et à mesure de l'évolution des clusters
    - Méthode k-means ou méthodes des centres mobiles
  - Méthode approchée
  - Méthode très populaire

Méthode k-Means (2)

- **Algorithme**
  - Soit K le nombre de clusters
  - Choisir K centres (aléatoirement, ++)
  - Itérations
    - Affecter chaque donnée au centre le plus proche (similarité)
    - Recalculer les nouveaux centres
  - Arrêt : stabilité

  - Similarité : **erreur quadratique / inertie / variance**
    - minimiser la **variance intra-cluster** : $\sum_{k=1}^{K} \frac{1}{n_k} \sum_{i \in C_k} d^2(x_i, \mu_k)$

Méthode k-Means (3)

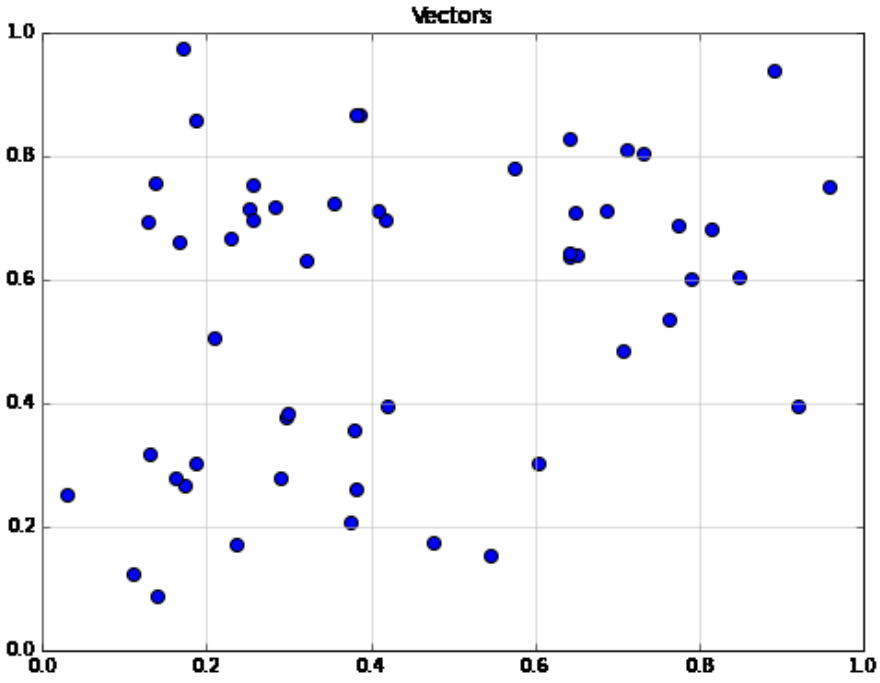
- **Similarité :**
  - $\sum_{k=1}^{K} \frac{1}{n_k} \sum_{i \in C_k} d^2(x_i, \mu_k)$ (min Squared Error)
- **Algorithme :**
  - Initialisation
    - Choisir k éléments (centres) : $\{\mu_1, \dots, \mu_k\}$
    - Les placer dans un cluster : $C_i \leftarrow \mu_i$
  - Répéter
    - Affecter chaque élément au centre le plus proche
      - $C_l = C_l \cup \{x_i\}$ tel que $l = argmin_k(d^2(x_i, \mu_k))$
    - Re-évaluer le centre de gravité de chaque cluster
      - $\mu_k = \frac{1}{n_k} \sum_{i \in C_k} x_i$
  - Jusqu'à : // Conditions d'arrêt //

Méthode k-Means (3)

- **Conditions d'arrêt :**
  - Nombre d'itérations
  - Plus de changement sur les centres de gravité (ou changements limités)
  - Pas de changement dans la composition des clusters
- **Convergence** :
  - Diminution de la fonction objectif
- **Complexité** : $O(KndI)$
  - où : I le nombre d'itérations

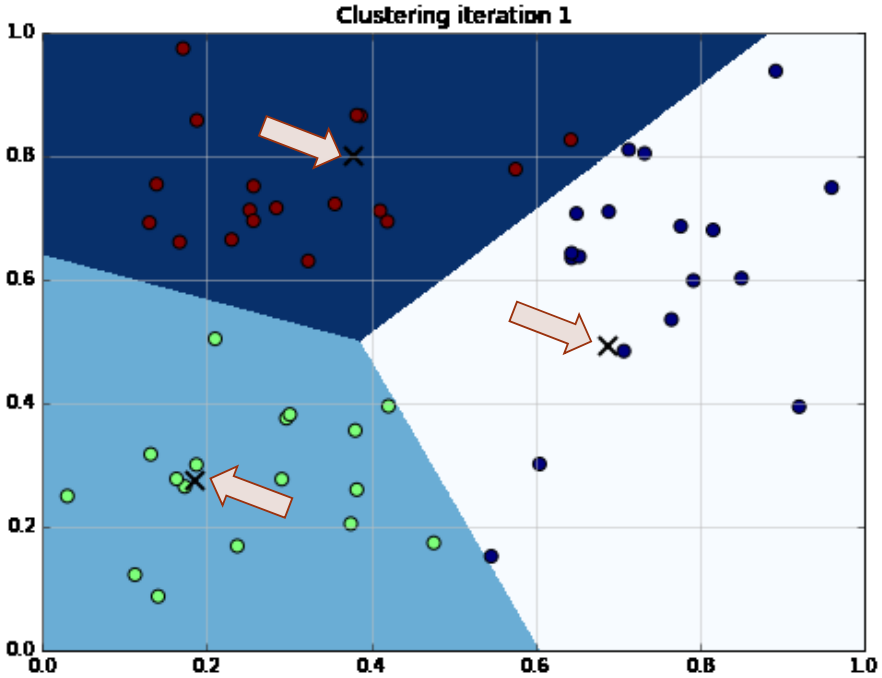
Exemple (1) – Données initiales

- Données initiales
- Déterminer 3 clusters



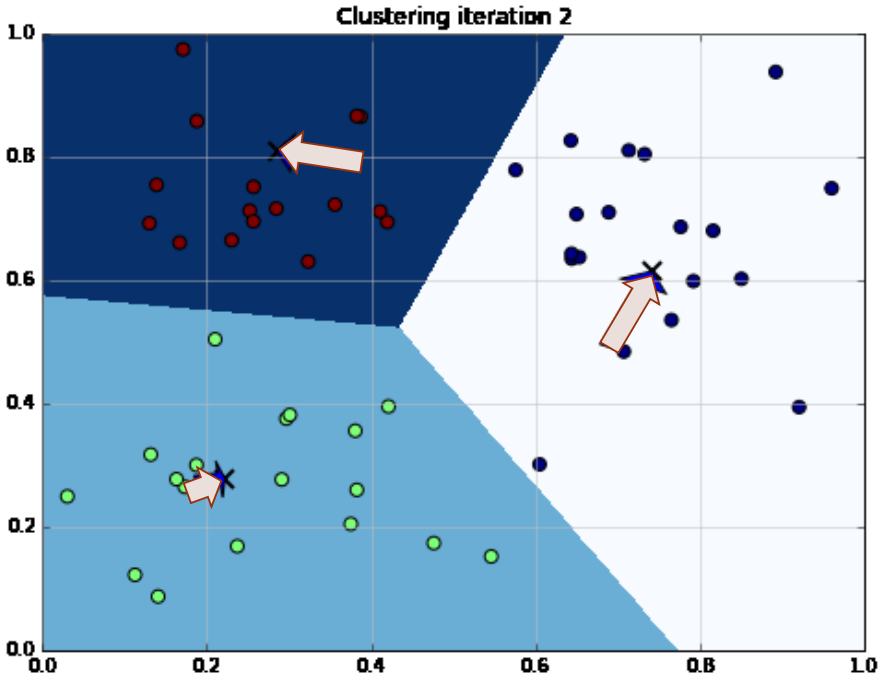
Exemple (2) – Choix des centres et Allocation

- Choisir 3 centres
- Allocation des points au centre le plus proche



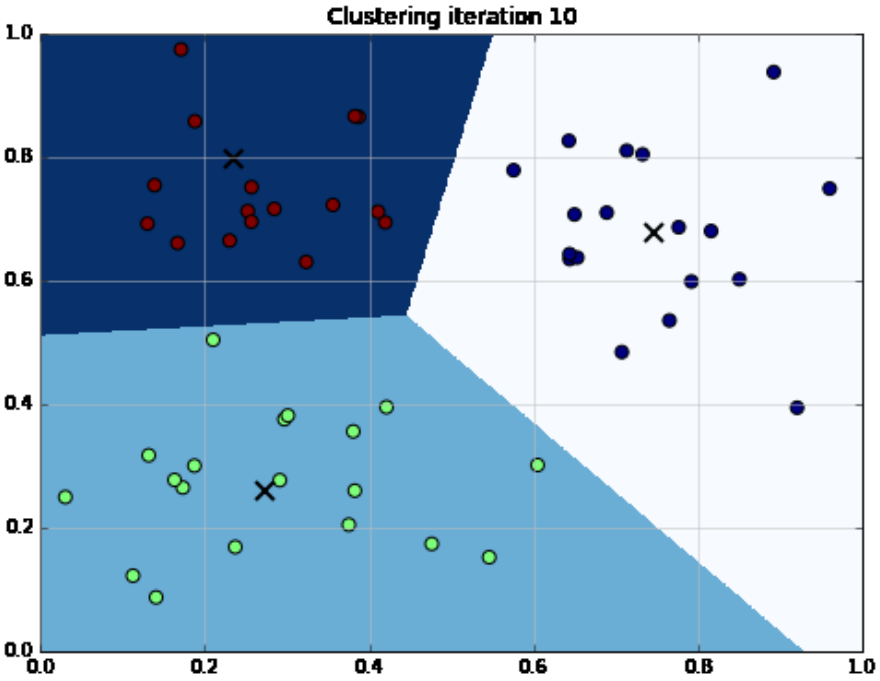
Exemple (3) – Re-calcul des centres

- Calcul des nouveaux centres pour les 3 clusters
- Nouvelle allocation des points aux centres



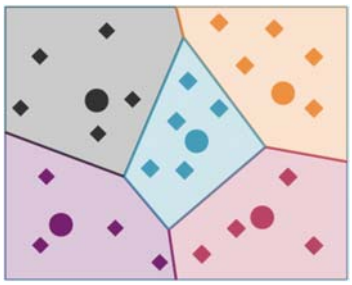
Exemple (4) – Dernière itération

- Solution obtenue après 10 itérations



Caractéristiques k-means (1)

- **Stratégie gloutonne :**
  - Obtention d'un minimum local / min erreur
  - Faible complexité
  - Passage à l'échelle
  - Compréhension simple de la méthode
- **Choix des points initiaux**
  - Fort impact sur le résultat
- **Forme des clusters**
  - Formes convexes
    - Chaque point d'un cluster est plus proche de son centre de gravité que des autres centres



Caractéristiques k-means (2)

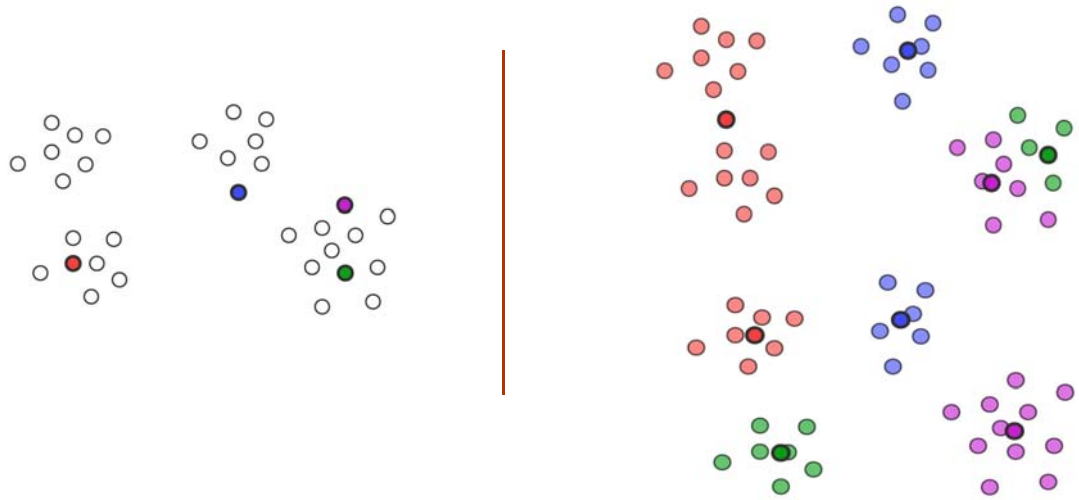
- **Points d'attention**
  - Nécessite de fixer le nombre de clusters
  - Nécessite l'existence d'une distance
  - Reste bloqué dans un optimum local

  - **Sensibilité à l'initialisation** :
  - **Sensibilité aux bruits et aux anomalies**
    - Tous les points sont inclus dans un cluster
    - K-means - -
      - Calcul de K clusters et détection de l anomalies
      - *SIAM International Conference on Data Mining 2013*
  - **Sensibilité à la densité des points, à la taille ou forme des clusters**

Initialisation (1)

- Aléatoire
  - Faire plusieurs exécutions avec différentes initialisation et conserver la meilleure solution
- Identification
  - Utiliser une méthode de clustering hiérarchique pour déterminer des centres
- Sélection
  - Fixer un nombre supérieur de centres et sélectionner parmi ceux-ci les centres conduisant à des clusters les plus séparables
- Post-processing : Split & Merge
  - Découper un cluster quand sa variance est supérieure à un seuil
  - Regrouper deux clusters quand la distance entre leurs centres est inférieure à un seuil

Initialisation (2)



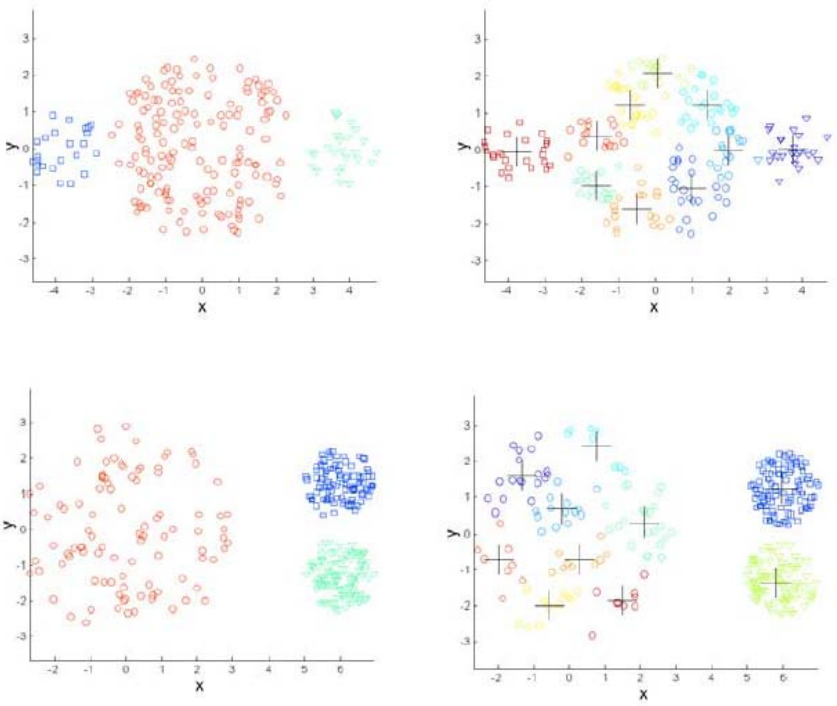
- **Initialisation** : K-means++
  - Choix des centres **avec une probabilité** liée à la distance au carré aux autres centres
  - Garanties / qualité du résultat par rapport à l'aléatoire (article 2007)

Nombres de clusters ?

- **Choix : problème difficile**
  - Fixé : segmenter en K (contraintes du problème)
  - Itérer sur différentes valeurs de K
    - Evaluer la qualité de chaque clustering
  - Imposer des contraintes sur le volume ou la densité des clusters
- **Stabilité des clusters**
  - Répéter la méthode
  - Regrouper dans un même cluster final les éléments qui se retrouvent toujours dans les mêmes clusters intermédiaires

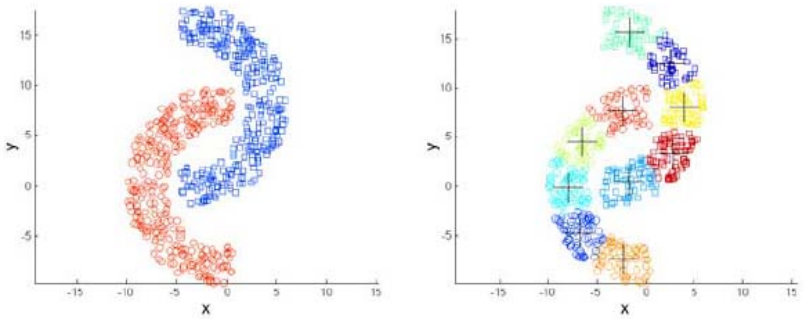
Variantes (1)

- **Augmenter le nombre de clusters**
  - Taille
  - Densité



Variantes (2)

- **Augmenter le nombre de clusters**
  - Forme



Variante

- **Méthode applicable en séquentiel**
  - Prise en compte de l'arrivée de nouveaux exemples
    - A chaque arrivée d'un exemple
      - Le placer dans le cluster le plus proche
      - Recalculer le centre de ce cluster
      - Les autres clusters restent inchangés

Plan

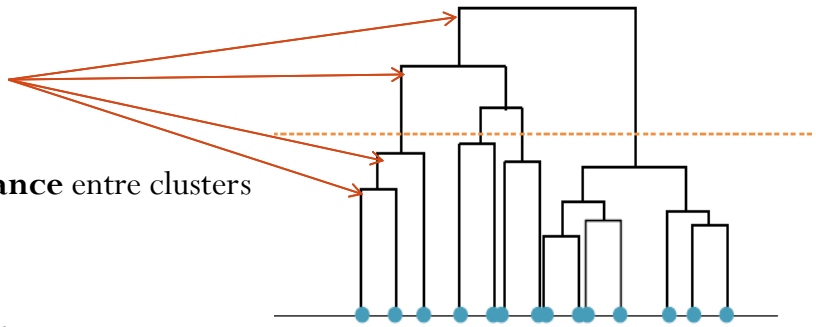
3. Problème de clustering
4. **Premières méthodes**
   1. Méthode par partitionnement : k-means
   2. Méthodes hiérarchiques : ascendant et descendant
5. Méthodes basées voisinage (densité) et basées graphes
6. Boite à outils

Méthodes hiérarchiques : Principe général

- **Deux types de méthodes hiérarchiques**
  - **Clustering ascendant (agglomératif)**
    - Initialement chaque observation (point) est un cluster
    - Fusionner les observations proches : mesure de similarité (ressemblance)
    - Itérer jusqu'à 1 seul cluster
  - **Clustering descendant (divisif)**
    - Initialement toutes les observations sont dans le même cluster
    - Le diviser jusqu'à séparer toutes les observations

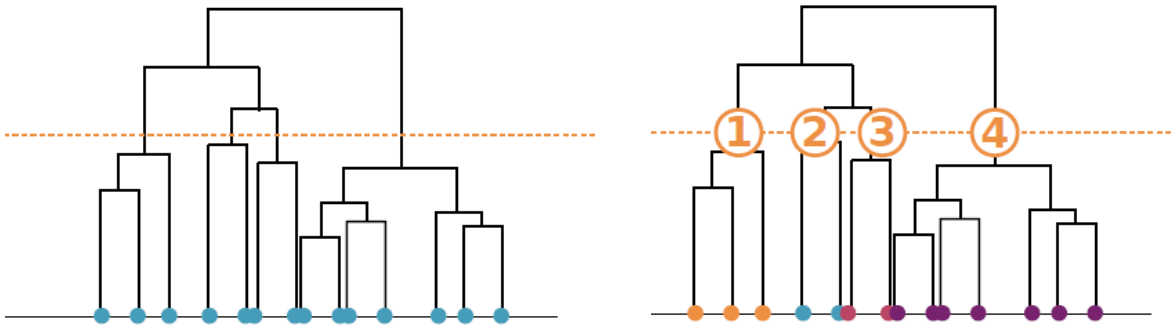
Dendrogramme (1)

- **Représentation du résultat**
  - Dendrogramme = arbre
    - Feuilles = échantillons
    - Nœuds = cluster
  - Hauteur des branches
    - Proportionnelle **distance** entre clusters
  - Représentation partielle
    - Le « haut » de l'arbre

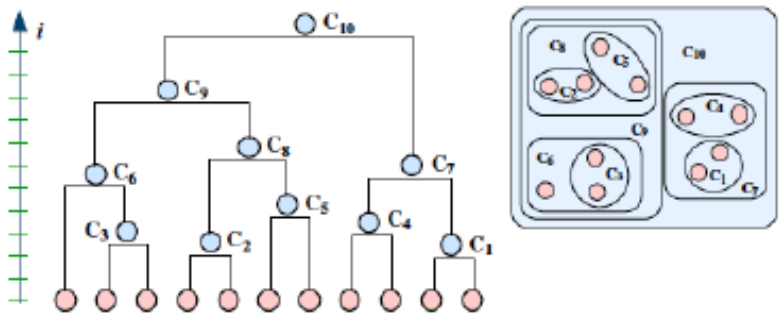


Dendrogramme (2)

- **Représentation du résultat**
  - Couper un dendrogramme
    - Un ensemble de clusters



Exemple



- Propriété : monotonie
  - Quand on fusionne deux clusters, la similarité avec un autre cluster n'augmente pas
  - Les fusions se font dans l'ordre croissant de similarité (distance)
  - Les barres horizontales (fusion/cluster) ne croisent pas les verticales

Clustering hiérarchique ascendant (CHA)

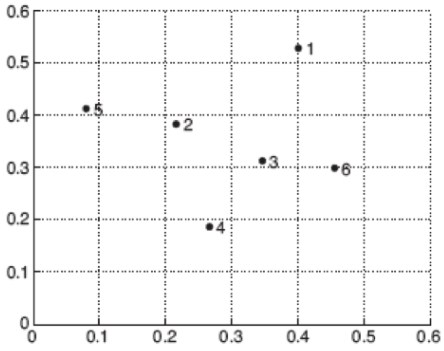
- **Principe**
  - Chaque point ou cluster est fusionné avec le cluster le plus proche
- **Algorithme**
  - Initialisation
    - Chaque point est placé dans son propre cluster
    - Calcul de la matrice M de « ressemblance »
  - Itérations
    - Sélection dans M des 2 clusters les plus proches : C_i et C_j
    - Fusion de C_i et C_j pour former un cluster C_k
    - Mise à jour de M en calculant « ressemblance » entre cluster C_k et autre clusters
  - Arrêt : fusion des 2 derniers clusters

Point clé : calcul de la similarité (ressemblance)

- Complexité : n^3 ($n-1$ recherche de minimum dans la matrice)
  - Variantes selon le calcul de similarité au moins n^2

Application (1)

- **Données**



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

- **Distance Euclidienne : mesure de « ressemblance »**

	p1	p2	p3	p4	p5	p6
p1	0					
p2	0,23	0				
p3	0,22	0,14	0			
p4	0,37	0,19	0,16	0		
p5	0,34	0,14	0,28	0,28	0	
p6	0,24	0,24	0,10	0,22	0,39	0

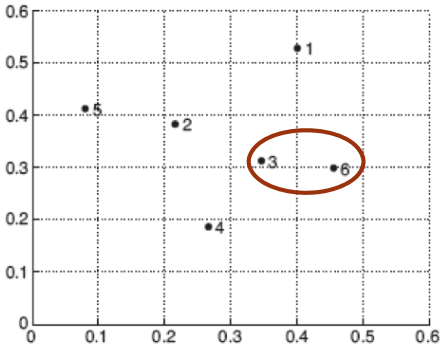
Application (2)

- **Saut minimal (single linkage)**

	p1	p2	p3	p4	p5	p6
p1	0					
p2	0,23	0				
p3	0,22	0,14	0			
p4	0,37	0,19	0,16	0		
p5	0,34	0,14	0,28	0,28	0	
p6	0,24	0,24	0,10	0,22	0,39	0

- Sélection Min ➔ 0,10
- Cluster (p3, p6)
- Mise à jour de la matrice de ressemblance
  - **Distance minimale**

	p1	p2	(p3,p6)	p4	p5
p1	0				
p2	0,23	0			
(p3,p6)	0,22	0,14	0		
p4	0,37	0,19	0,16	0	
p5	0,34	0,14	0,28	0,28	0



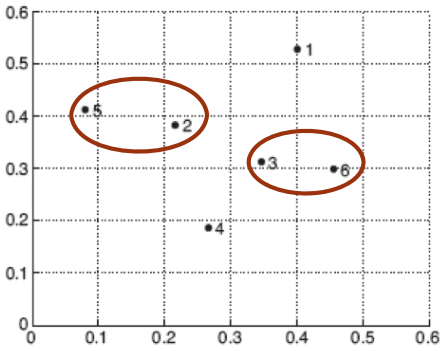
Application (3)

- **Saut minimal (single linkage)**

	p1	p2	(p3,p6)	p4	p5
p1	0				
p2	0,23	0			
(p3,p6)	0,22	0,14	0		
p4	0,37	0,19	0,16	0	
p5	0,34	0,14	0,28	0,28	0

- Sélection Min ➔ 0,14
- Cluster (p2, p5)
- Mise à jour de la matrice de ressemblance
  - **Distance minimale**

	p1	(p2,p5)	(p3,p6)	p4
p1	0			
(p2,p5)	0,23	0		
(p3,p6)	0,22	0,14	0	
p4	0,37	0,19	0,16	0

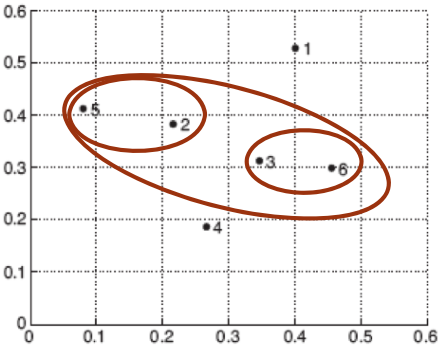


Application (4)

- **Saut minimal (single linkage)**

	p1	(p2,p5)	(p3,p6)	p4
p1	0			
(p2,p5)	0,23	0		
(p3,p6)	0,22	0,14	0	
p4	0,37	0,19	0,16	0

- Sélection Min ➔ 0,14
- Cluster (p2, p5, p3, p6)
- Mise à jour de la matrice de ressemblance
  - **Distance minimale**



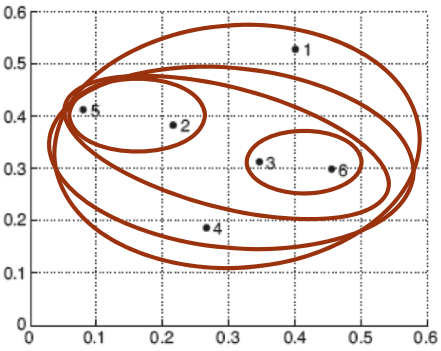
	p1	(p2,p5,p3,p6)	p4
p1	0		
(p2,p5,p3,p6)	0,22	0	
p4	0,37	0,16	0

Application (4)

- **Saut minimal (single linkage)**

	p1	(p2,p5,p3,p6)	p4
p1	0		
(p2,p5,p3,p6)	0,22	0	
p4	0,37	0,16	0

- Sélection Min ➔ 0,16
- Cluster (p2, p5, p3, p6, p4)
- Mise à jour de la matrice de ressemblance
  - **Distance minimale**

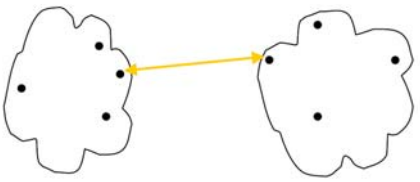


	p1	(p2,p5,p3,p6,p4)
p1	0	
(p2,p5,p3,p6,p4)	0,22	0

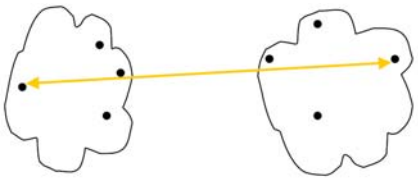
- Sélection Min ➔ 0,22
- Cluster (p2, p5, p3, p6, p4,p1)
- Arrêt

Calcul de similarité (1)

- **Difficulté :**
  - Trouver une métrique entre les clusters
- **Différentes possibilités**
  - **Valeur minimale**
    - Distance entre les 2 points les plus proches
    - $D_{min}(C_i, C_j) = \min(d(x_i, x_j), x_i \in C_i; x_j \in C_j)$
      - Classes assez « générales »
      - Sensibilité aux anomalies et aux données bruitées

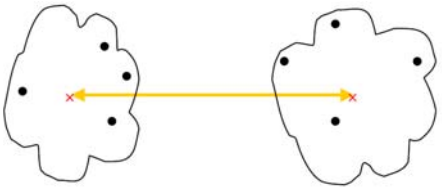


  - **Valeur maximale**
    - Distance entre les 2 points les plus éloignés
    - $D_{max}(C_i, C_j) = \max(d(x_i, x_j), x_i \in C_i; x_j \in C_j)$
      - Classes plus spécifiques (points regroupés très proches)
      - Sensibilité aux anomalies et aux données bruitées

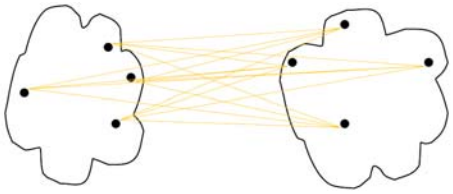


Calcul de similarité (2)

- **Valeur entre centres**
  - Distance entre les centres de chaque cluster
  - $D_{cg}(C_i, C_j) = d(\mu_i, \mu_j)$
    - Moins sensible aux anomalies et données bruitées



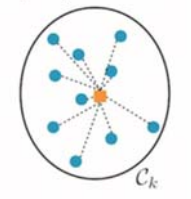
- **Valeur moyenne**
  - Distance moyenne entre toute paire de points
  - $D_{moy}(C_i, C_j) = \frac{\sum_{x_i \in C_i} \sum_{x_j \in C_j} d(x_i, x_j)}{n_i \times n_j}$
    - Classes homogènes
    - Moins sensible aux anomalies et données bruitées



Autre approche

- **Clustering de Ward**
  - Méthode hiérarchique
  - Maximiser l'homogénéité des clusters
    - Les méthodes précédentes visent à la séparation
  - Deux clusters sont fusionnés :
    - l'augmentation de la variance intra-cluster est minimale
      - moyenne des distances au carré entre chaque point et le centre μ_k

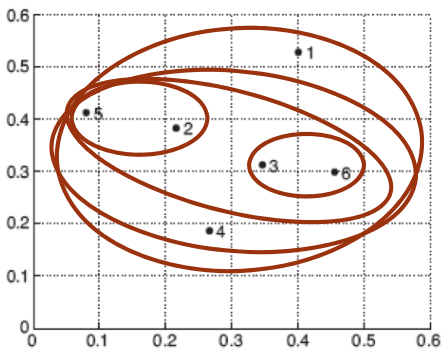
$$I_k = \frac{1}{n_k} \sum_{x \in C_k} d^2(x, \mu_k)$$

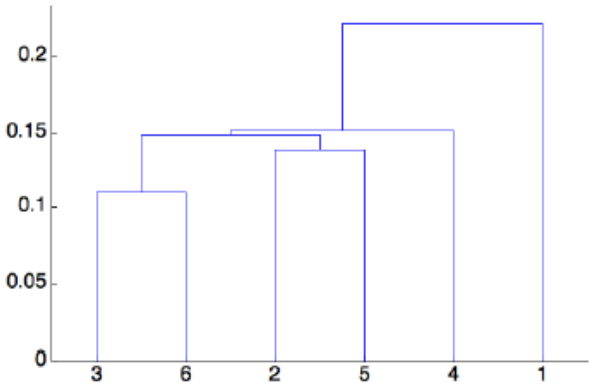


Résultats différents (1)

- **Saut minimal (single linkage)**

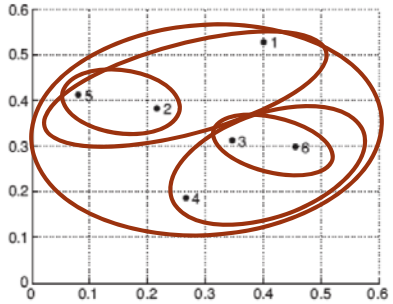
	p1	p2	p3	p4	p5	p6
p1	0					
p2	0,23	0				
p3	0,22	0,14	0			
p4	0,37	0,19	0,16	0		
p5	0,34	0,14	0,28	0,28	0	
p6	0,24	0,24	0,10	0,22	0,39	0



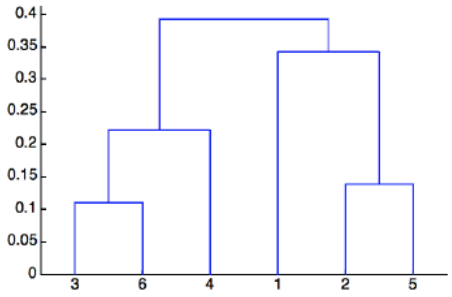


Résultats différents (2)

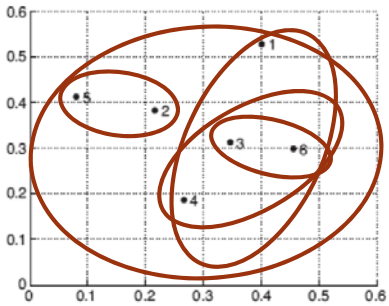
- **Saut maximal (complete linkage)**



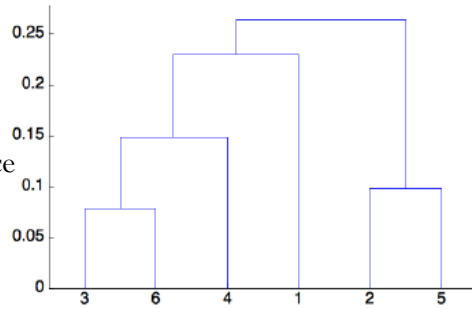
- Sélection Min
- Mise à jour de la matrice
  - Distance maximale



- **Saut moyen (average linkage)**



- Sélection Min
- Mise à jour de la matrice
  - Distance moyenne



Synthèse clustering hiérarchique ascendant

- **Méthode flexible**
  - Nombre de clusters non fixé
    - A établir en fonction du dendrogramme
    - Evaluer les différentes partitions en utilisant les mesures de qualité d'un clustering

- **Caractéristiques :**
  - Complexité : au moins n^2 (calcul de distance)
  - Passage à l'échelle difficile
  - Pas de remise en cause des classes fusionnées
  - Sensible aux anomalies (outliers)

Plan

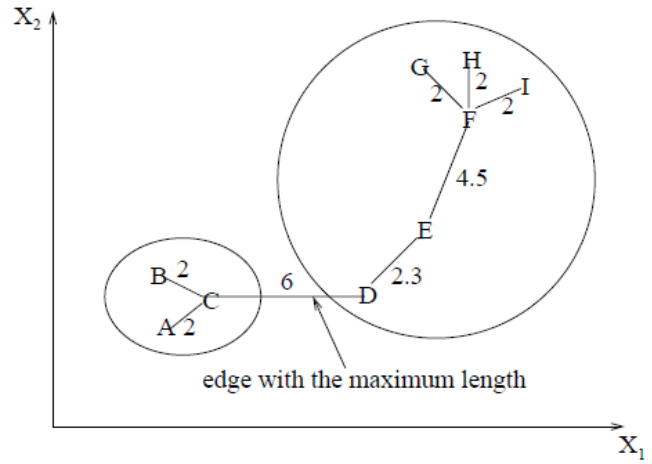
3. Problème de clustering
4. **Premières méthodes**
   1. Méthode par partitionnement : k-means
   2. Méthodes hiérarchiques : ascendant et descendant
5. Méthodes basées voisinage (densité) et basées graphes
6. Boite à outils

Clustering hiérarchique descendant

- **Principe**
  - Clustering descendant (divisif)
    - Initialement tous les points sont dans le même cluster
    - Le diviser jusqu'à séparer tous les points
      - Sélectionner les points les moins similaires
  - Assez peu de méthodes ?
    - Nombre de possibilités pour diviser en 2 : $2^{n-1} - 1$
    - Approche ascendante : nombre de possibilités pour regrouper : $\frac{n(n-1)}{2}$
  - Approches heuristiques
    - Ascendante : regrouper les observations les plus proches
    - Descendante : séparer les observations les plus éloignées
    - basées sur calculs de distance

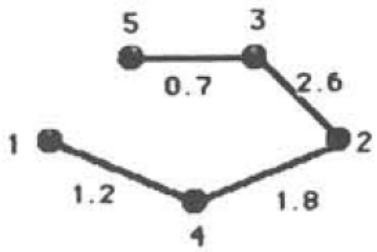
Méthode basée sur un calcul d'arbre couvrant

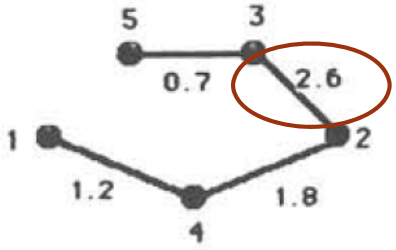
- **Calcul de l'arbre couvrant minimal**
  - Minimal Spanning Tree (MST)
    - Distance « saut minimal » (single link)

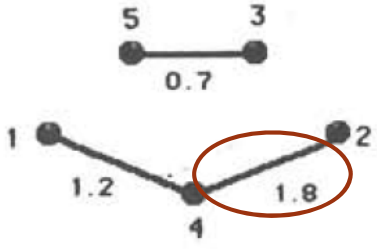


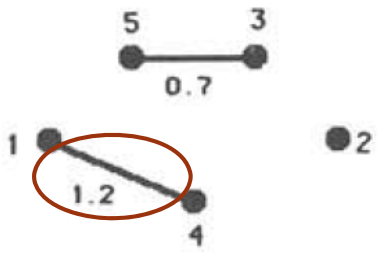
Exemple

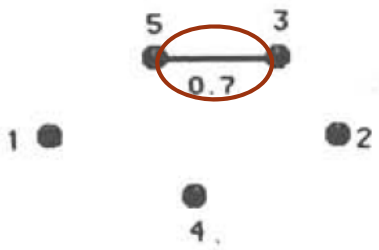
- **Clustering descendant et arbre couvrant**











Plan

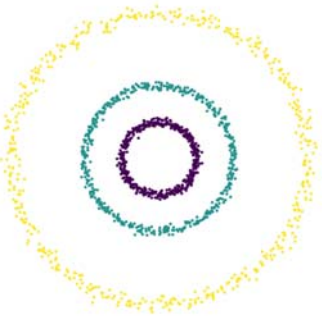
1. Contexte : l'Intelligence Artificielle
2. Contexte : l'apprentissage automatique
3. Problème de clustering
4. Premières méthodes
5. **Méthodes basées voisinage (densité) et basées graphes**
6. **Boite à outils**
7. Fouille de données
8. Réduction de dimensions (Analyse en Composantes principales)

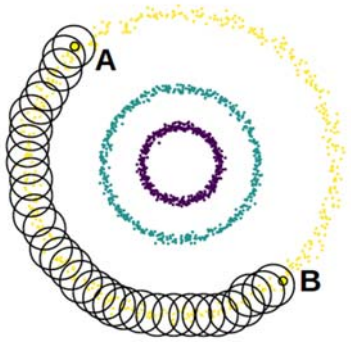
Plan – section 5

3. Problème de clustering
4. Premières méthodes
   1. Méthode par partitionnement : k-means
   2. Méthodes hiérarchiques : ascendant et descendant
5. **Méthodes basées voisinage (densité) et basées graphes**
   1. Clustering basé densité : DBSCAN
   2. Clustering basé graphe
6. Boite à outils

Méthode DBSCAN (1)

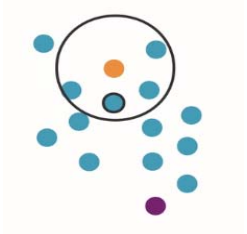
- **Objectif**
  - Obtenir des formes non convexes
  - Principe
    - Pour associer les 2 points A et B
      - créer un chemin pour passer de l'un à l'autre *en restant à l'intérieur du même cluster*
      - Notion de voisinage
  - **DBSCAN : *Density-Based Spatial Clustering of Applications with Noise***
  - **Densité :**
    - Nombre de points compris dans un rayon donné

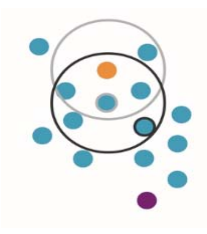


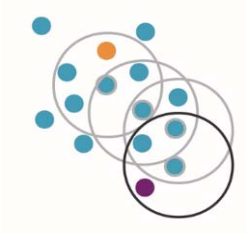


Méthode DBSCAN (2)

- **Voisinage**
  - Pour une observation x_i, et une valeur ε
    - Epsilon voisinage : $N_\varepsilon(x_i) = \{x_j \in X \mid d(x_i, x_j) < \varepsilon\}$
  - **Point intérieur** (core point) x_i : si $|N_\varepsilon(x_i)| \geq min_{pt}$
  - **Points connectés** par densité : x_i et x_j sont connectés si
    - Il existe une suite de points intérieurs $y_1, y_2, \ldots, y_m$ tels que
      - $y_1 \in N_\varepsilon(x_i),\ y_2 \in N_\varepsilon(y_1),\ \ldots., x_j \in N_\varepsilon(y_m)$

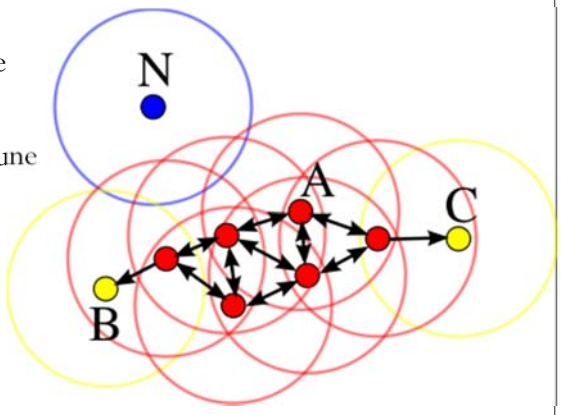






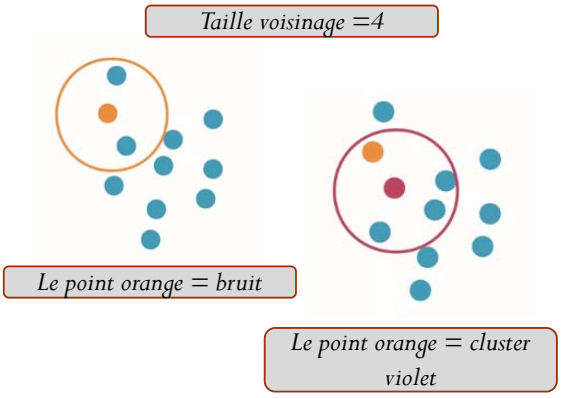
Méthode DBSCAN (3)

- **Exemple :**
  - Seuil min_{pt}=4
    - **Points intérieurs** : tous les points en rouge
      - Même cluster que A
    - **Points atteignables** : tous les points en jaune
      - Ne sont pas des points intérieurs
      - Taille de voisinage trop faible
      - Mais dans voisinage de points intérieurs
      - Même cluster que A
    - **Points atypiques** : point en bleu
      - Ne sont pas atteignables par les points intérieurs existant
      - Ne sont pas eux-mêmes des points intérieurs



Méthode DBSCAN (4)

- **Principe**
  - Maintenir une liste de points visités
  - Répéter
    - Sélectionner un point x non visité
    - Construire N le voisinage de x
    - Si $|N| < min_{pt}$ alors marquer $x \leftarrow$ bruit
    - Sinon // x *est un point intérieur*
      - Initialiser un cluster $C \leftarrow \{x\}$
      - Agrandir le cluster C par voisinage
      - Ajouter C à la liste des clusters
      - Marquer les points de C comme visités
  - Arrêt : tous les points sont visités



Méthode DBSCAN (5)

- **Algorithme**
  - **DBSCAN**(D, eps, MinPts)

```
k = 0
pour chaque point P non visité des données D
    marquer P comme visité
    PtsVoisins = epsilonVoisinage(D, P, eps)
    si tailleDe(PtsVoisins) < MinPts
        marquer P comme BRUIT
    sinon
        k ← k+1
        etendreCluster(D, P, PtsVoisins, k, eps, MinPts)
```

  - **etendreCluster**(D, P, PtsVoisins, k, eps, MinPts)

```
ajouter P au cluster k
pour chaque point P' de PtsVoisins
    si P' n'a pas été visité
        marquer P' comme visité
        PtsVoisins' = epsilonVoisinage(D, P', eps)
        si tailleDe(PtsVoisins') >= MinPts
            PtsVoisins = PtsVoisins U PtsVoisins'
    si P' n'est membre d'aucun cluster
        ajouter P' au cluster k
```

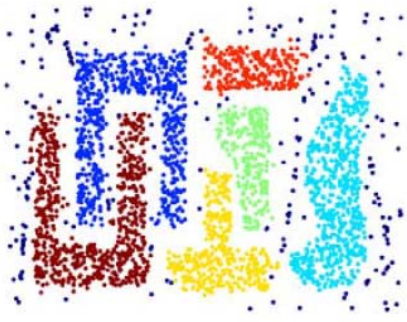
  - **epsilonVoisinage**(D, P, eps)

    retourner tous les points de D qui sont à une distance inférieure à epsilon de P

Méthode DBSCAN (6)

- **Exemple**

Original Points

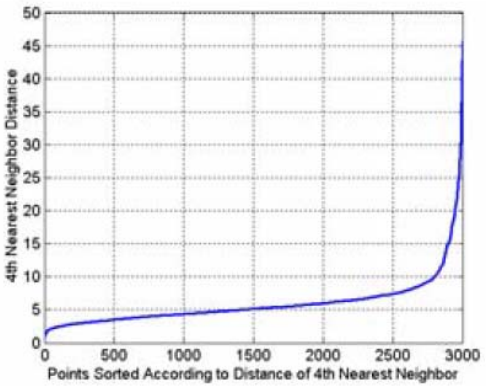


Clusters

Caractéristiques (1)

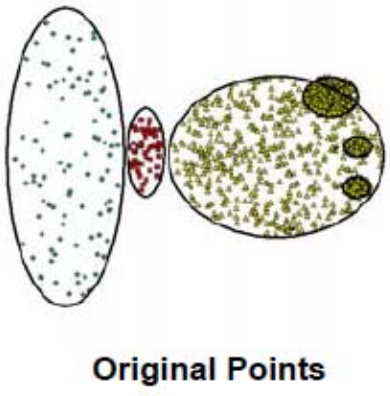
- **Intérêts**
  - Pas besoin de fixer le nombre de cluster
  - Peut déterminer des clusters non convexes
  - Est robuste au bruit et anomalies

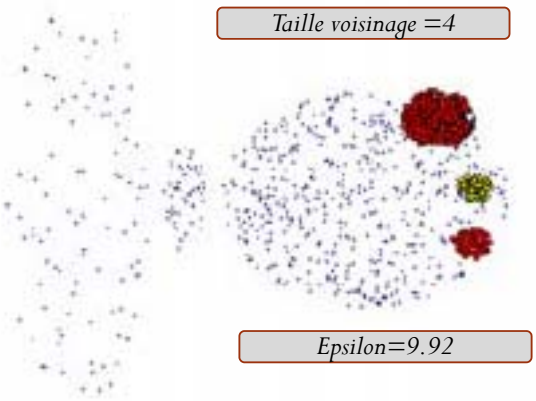
- **Difficulté**
  - Paramètres à déterminer
    - Fixer la taille du voisinage et le nombre de points à considérer
      - Lien valeur de k et MinPts
      - Déterminer le graphes des distances des k plus proches voisins de chaque point
      - Inflexion : guide pour la valeur epsilon

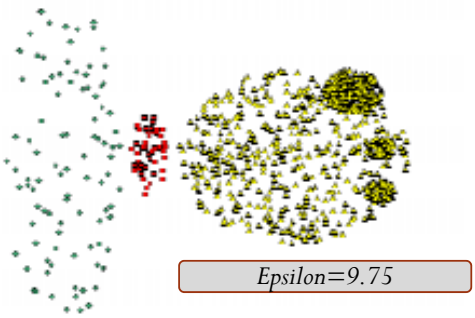


Caractéristiques (2)

- **Limites**
  - Densité variable dans les données







  - Grande dimension

Plan

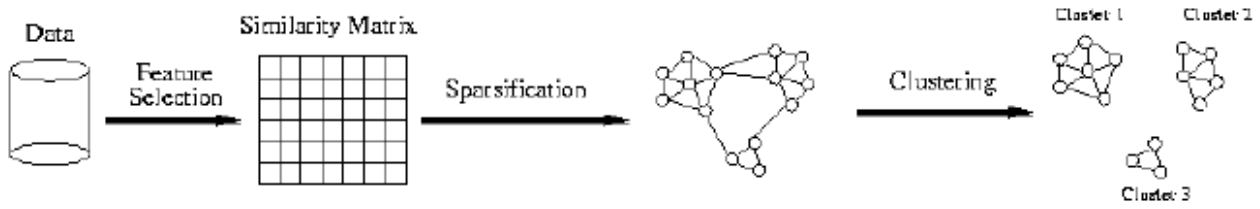
3. Problème de clustering
4. Premières méthodes
   1. Méthode par partitionnement : k-means
   2. Méthodes hiérarchiques : ascendant et descendant
5. **Méthodes basées voisinage (densité) et basées graphes**
   1. Clustering basé densité : DBSCAN
   2. Clustering basé graphe
6. Boite à outils

Clustering basé graphe (1)

- **Graphe de proximité**
  - Un point : un sommet du graphe
  - Chaque lien entre 2 sommets est valué par une pondération représentant la proximité
  - Déterminer la matrice de similarité / proximité
  - Graphe obtenu : graphe complet
- **Principe**
  - Filtrer des liens (sommets éloignés)
  - Clusters : « composantes connexes » du graphe
  - « Sparsification » du graphe

Clustering basé graphe (2)

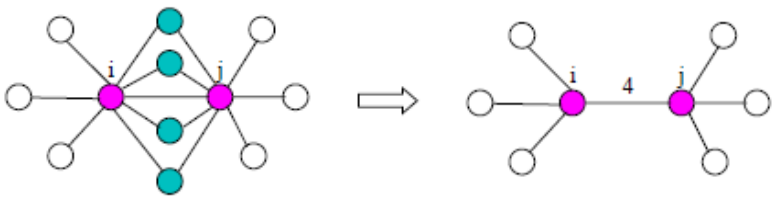
- **« Sparsification » du graphe**
  - Conserver un nombre réduit de voisins pour chaque sommet
    - Données proches : devraient être dans le même cluster
    - Réduire les effets du bruit et des anomalies



  - Réduire la quantité d'information à manipuler
    - Diminuer le temps de calcul pour le clustering
    - Augmenter la taille des problèmes pouvant être considérés

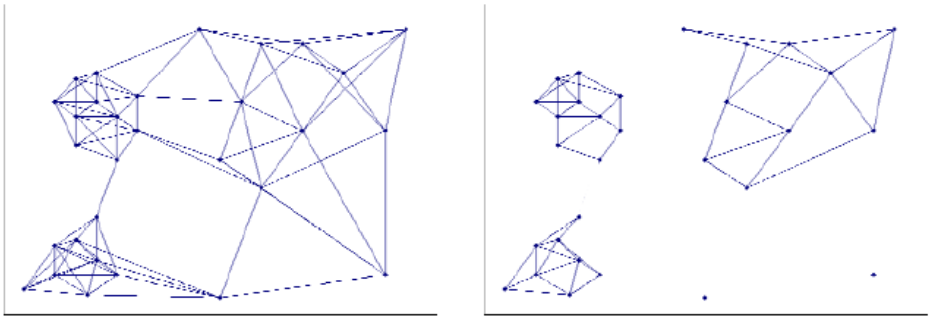
Clustering basé graphe (3)

- **Graphe de voisinage**
  - Un point : un sommet du graphe
  - Chaque lien entre 2 sommets est valué par leur nombre de voisins en commun (si les 2 sommets sont connectés)
  - **Graphe SNN** : Shared Nearest Neighbor



Clustering basé graphe (4)

- **Principe**
  - Partir du graphe de proximité (clairsemé)
  - Établir le graphe SNN



Sparse Graph

Shared Near Neighbor Graph

  - Méthodes de clustering sur le graphe SNN

Clustering basé graphe (5)

- **Algorithme SNN-DBSCAN**

1. Calculer la matrice de similarité point à point
2. Filtrer la matrice pour ne conserver que k voisins les plus similaires
3. Construire le graphe SNN
4. Appliquer le principe de DBSCAN
   - Paramètres : epsilon et MinPts
     - Calculer le epsilon-voisinage de chaque point x ➔ densité SNN(x)
     - Déterminer les points intérieurs (voisinage au moins de taille MinPts) et construire les clusters associés
     - Retirer les points atypiques

Variantes (1)

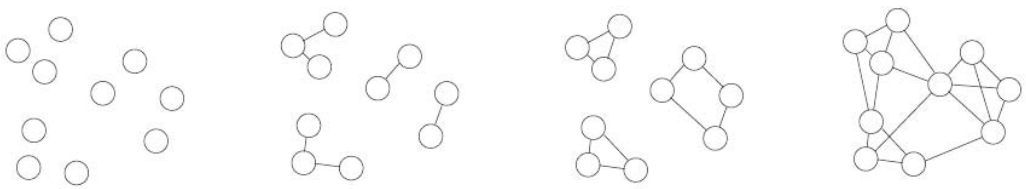
- **Autres méthodes utilisant le graphe SNN**
  - Jarvis-Patrick, Chameleon, Rock,

  - Jarvis-Patrick

  1. Calculer la matrice de similarité point à point
  2. Filtrer la matrice pour ne conserver que k voisins les plus similaires
  3. Construire le graphe SNN
     - Appliquer un seuil de similarité sur la matrice
     - appliquer une recherche de composantes connexes pour obtenir les clusters

Variantes (2)

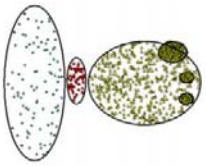
- **Méthode Chameleon**
  - Pré-processing :

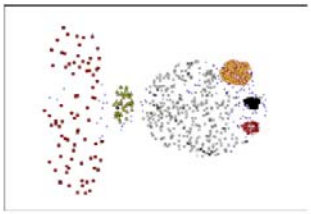


  - Deux phases
    - Déterminer des sous clusters initiaux (partition du graphe des k-ppv)
    - Approche hiérarchique ascendante :
      - Mesures spécifiques pour regrouper des clusters (inter-connectivité et proximité)

Méthode SNN-DBSCAN

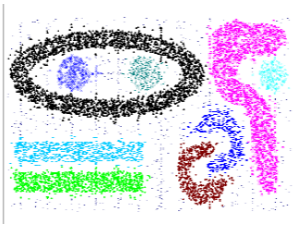
- **Intérêt**
  - Densité variable des données
  - Formes complexes

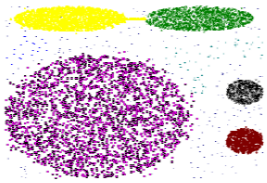




Original Points

SNN Clustering





- **Limites**
  - Tous les points ne sont pas placés dans un cluster
  - Paramétrage (lié à DBSCAN)

Plan

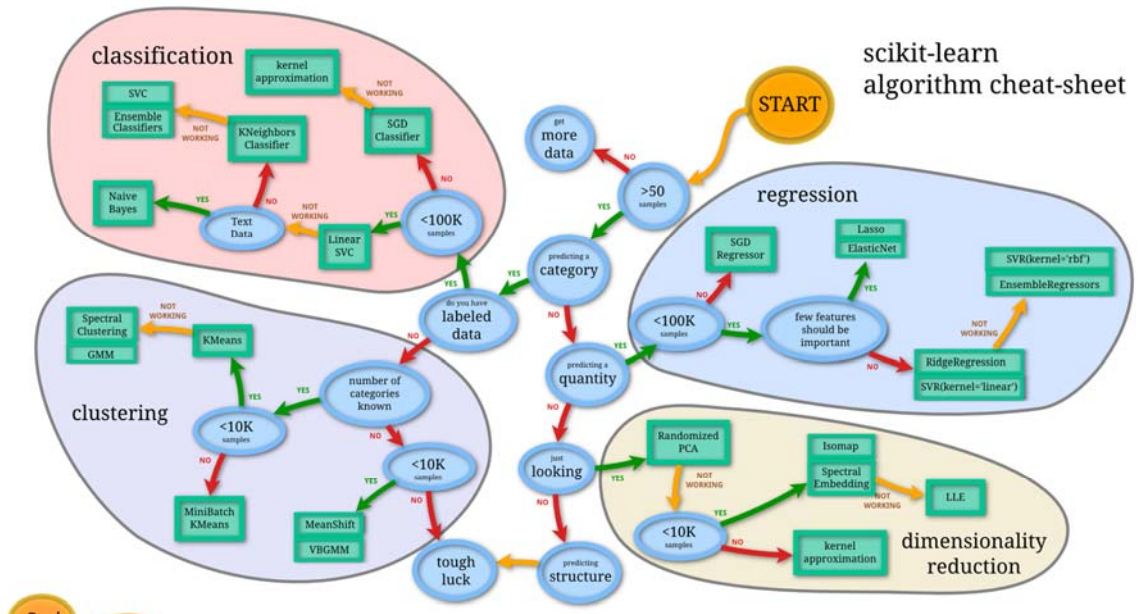
1. Contexte : l'Intelligence Artificielle
2. Contexte : l'apprentissage automatique
3. Problème de clustering
4. Premières méthodes
5. Méthodes basées voisinage (densité) et basées graphes
6. **Boite à outils**
7. Fouille de données
8. Réduction de dimensions (Analyse en Composantes principales)

Scikitlearn (1)

- **Scikitlearn**
  - Librairie d'algorithmes d'apprentissage surpervisé et non suppervisé
  - Interface en Python
  - Basé sur (NumPy, SciPy, Matplotlib, Ipython, Sympy, Pandas)
  - Pour le chargement, la manipulation et le résumé de données : NumPy, Pandas

- **Scikit-learn homepage**
  - http://scikit-learn.org
- **Presentations et Tutorials**
  - http://scikit-learn.org/stable/presentations.html

Scikitlearn (2)



Scikitlearn (3)

- **Clustering**
  - 9 méthodes proposées et comparées
    - http://scikit-learn.org/stable/modules/clustering.html

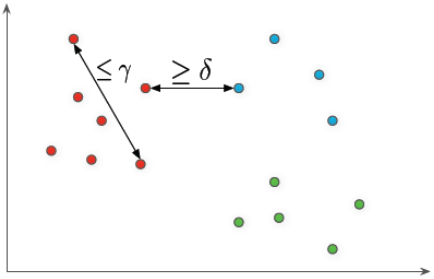
Conclusion

- **Clustering**
  - Problème complexe
    - mal défini
    - Taille de l'espace des solutions
  - Très nombreuses méthodes
    - Focus sur quelques méthodes approchées
      - Définition de similarité / distance
    - Méthodes dépendant du contexte applicatif
    - Assez peu de méthodes exactes
      - Graphes, Programmation Linéaire en nombre entiers, Programmation par Contraintes
      - Algorithmes de Branch and Bound, Méthodes de filtrage, ….

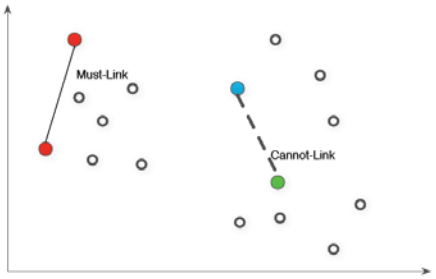
Conclusion

- **Variantes**
  - Clustering sous contraintes
    - Clusters

Contrainte de capacité:
$\alpha \leq |C_i| \leq \beta$
Contrainte du diamètre maximal
Contrainte de marge minimale
Contrainte de densité
...



    - Elements



T-B-H Dao, K-C Duong, C. Vrain - Constrained clustering by constraint programming. Artificial Intelligence Journal (244, pp. 70-94) - 2017