Ingénierie de la fouille et de la visualisation de données massives (RCP216)

Classification automatique

Michel Crucianu
(prenom.nom@cnam.fr)
http ://cedric.cnam.fr/vertigo/Cours/RCP216/

Département Informatique
Conservatoire National des Arts & Métiers, Paris, France

19 octobre 2021

Plan du cours

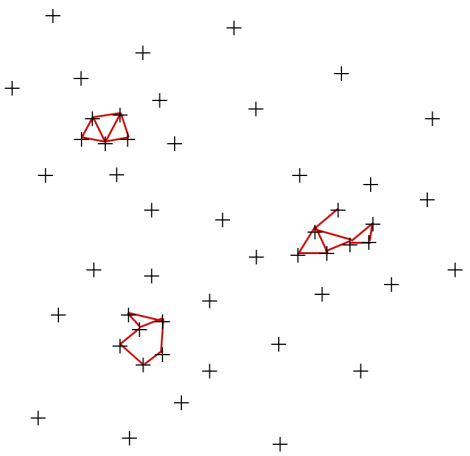
Typologie des méthodes de classification automatique

(*cluster analysis, clustering*)

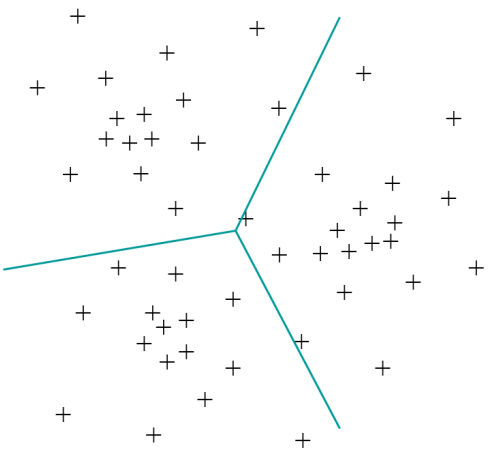
- Objectif :
  - Partitionnement des données
  - Hiérarchie de groupes ($\rightarrow$ plusieurs partitionnements disponibles)
- Nature des données : numériques, catégorielles, mixtes
- Représentation des données :
  - Riche : représentation vectorielle permettant de définir centre de gravité, densité, intervalles, différentes métriques
  - Simple : l'unique structure disponible est une métrique
- Nature des groupes recherchés : mutuellement exclusifs ou non, nets ou flous
- Définition des groupes (critère de regroupement) :
  - Ensembles compacts éloignés entre eux
  - Ensembles denses séparés par des régions moins denses

Classification automatique *vs* auto-jointure par similarité

- Auto-jointure par similarité
  - Données à distance $< \theta$
  - Extraction ultérieure de cliques, graphes connexes
  - Autres données : ignorées



- Classification automatique
  - Regroupement des données par similarité
  - Chaque donnée appartient à une partition



Centres mobiles : la méthode

- Ensemble $\mathcal{E}$ de N données décrites par p variables à valeurs dans $\mathbb{R}$
- Objectif : répartir les N données en k groupes disjoints $\mathcal{E}_1, \ldots, \mathcal{E}_k$ (inconnus *a priori*) en optimisant la somme des inerties intra-classe

$$\phi_{\mathcal{E}}(\mathcal{C}) = \sum_{j=1}^{k} \sum_{\mathbf{x}_i \in \mathcal{E}_j} d^2(\mathbf{x}_i, \mathbf{m}_j) \tag{1}$$

avec $\mathcal{C} = \{\mathbf{m}_j, 1 \leq j \leq k\}$ l'ensemble des centres des k groupes, d la distance dans $\mathbb{R}^p$ qui définit la nature des dissimilarités

Centres mobiles : l'algorithme

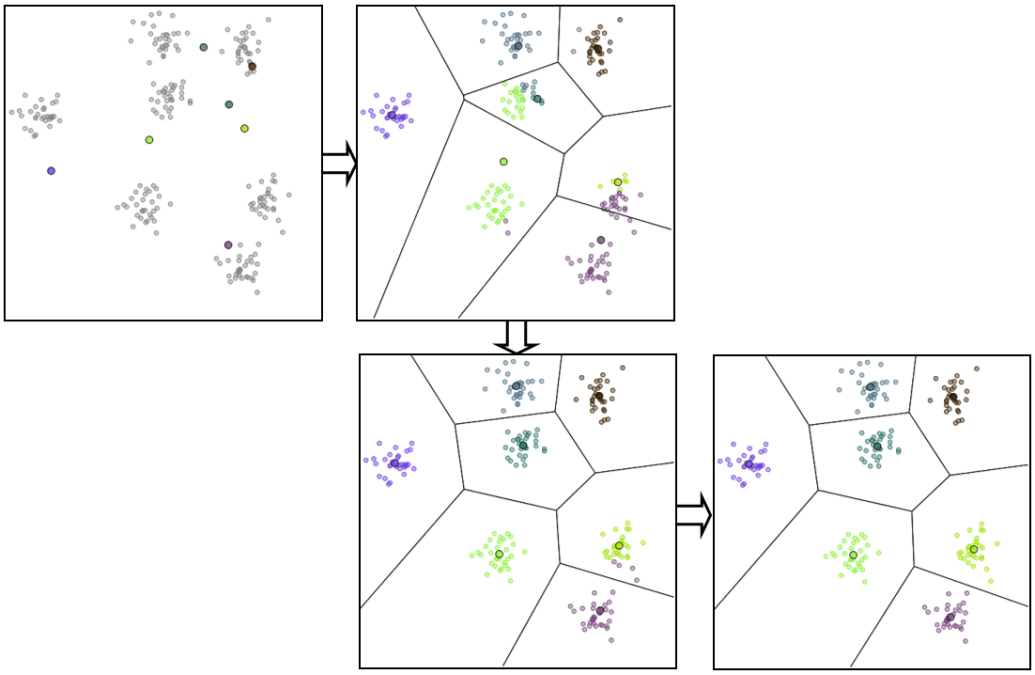
Data : Ensemble $\mathcal{E}$ de N données de $\mathbb{R}^p$
Result : k groupes (*clusters*) disjoints $\mathcal{E}_1, \ldots, \mathcal{E}_k$ et ensemble $\mathcal{C}$ de leurs centres

1 Initialization aléatoire des centres $\mathbf{m}_j$, $1 \leq j \leq k$;
2 **while** *centres non stabilisés* **do**
3 | Affectation de chaque donnée au groupe du centre le plus proche ;
4 | Remplacement des anciens centres par les centres de gravité des groupes ;
5 **end**

- $\phi_{\mathcal{E}}(\mathcal{C})$ diminue lors de chacune des deux étapes du processus itératif ; comme $\phi_{\mathcal{E}}(\mathcal{C}) \geq 0$, le processus itératif doit converger
- ... mais la solution obtenue sera un minimum *local*, dépendant de l'initialisation, souvent beaucoup moins bon que le minimum global

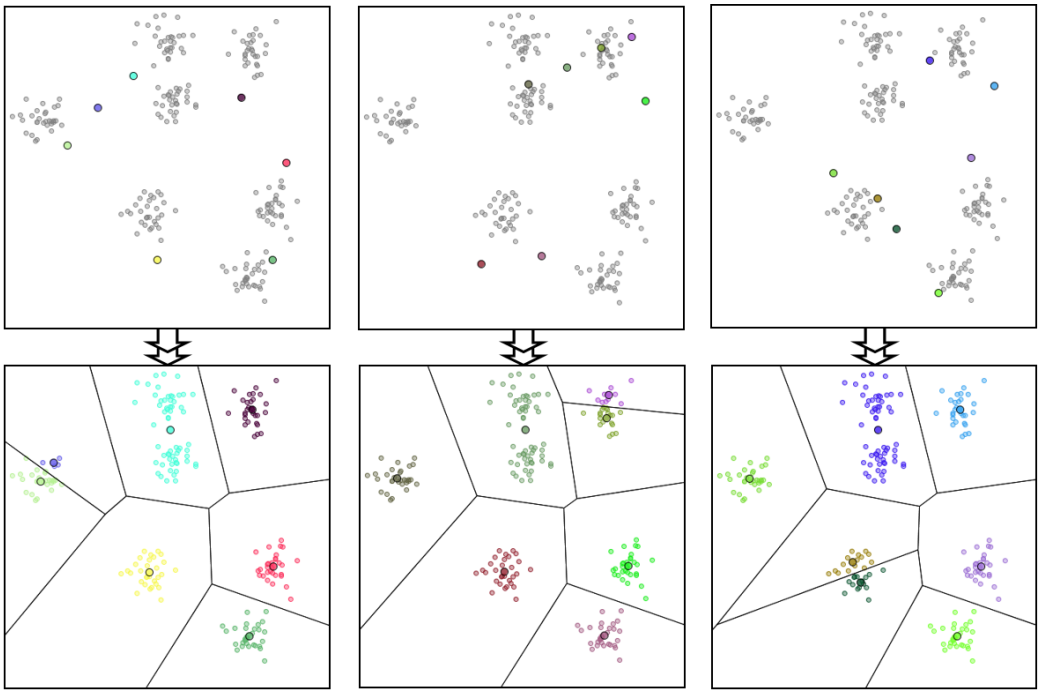
Centres mobiles : illustration

(données issues de 7 lois normales bidimensionnelles, classification avec 7 centres)



Centres mobiles : illustration (2)

(résultats avec 3 initialisations différentes)



K-means : l'algorithme

- Souvent, on appelle *K-means* une variante non *batch* de la méthode des centres mobiles ; parfois, *K-means* est utilisé comme synonyme des centres mobiles...

Data : Ensemble $\mathcal{E}$ de N données de $\mathbb{R}^p$
Result : k groupes (*clusters*) disjoints $\mathcal{E}_1, \ldots, \mathcal{E}_k$ et ensemble $\mathcal{C}$ de leurs centres

1 Initialization aléatoire des centres $\mathbf{m}_j$, $1 \leq j \leq k$;
2 **while** *centres non stabilisés* **do**
3 Choix aléatoire d'une des données ;
4 Affectation de la donnée au groupe du centre le plus proche ;
5 Recalcul des centres pour le groupe rejoint par la donnée et le groupe quitté par la donnée ;
6 **end**

- Faire tourner *K-means* plusieurs fois, à partir d'initialisations aléatoires différentes, ne donne pas la garantie d'arriver à une bonne solution !

K-means : une implémentation simple MapReduce

Data : Ensemble $\mathcal{E}$ de N données de $\mathbb{R}^p$
Result : k groupes (*clusters*) disjoints $\mathcal{E}_1, \ldots, \mathcal{E}_k$ et ensemble $\mathcal{C}$ de leurs centres

1 L'ensemble $\mathcal{E}$ de N données est découpé en fragments distribués aux nœuds de calcul ; un fragment doit tenir dans la mémoire d'un nœud ;
2 Un nœud initialise les k centres ;
3 **while** *centres non stabilisés* **do**
4 Transmettre l'ensemble $\mathcal{C}$ des centres à tous les nœuds de calcul ;
5 Chaque tâche *Map(t)*, pour chaque élément $\mathbf{x}_i$ de son fragment t, trouve le centre le plus proche j ; ensuite, pour chaque centre j ainsi trouvé, génère $(j, (n_{jt}, \tilde{\mathbf{m}}_{jt} = \sum_t \mathbf{x}_i))$, où n_{jt} est le nombre de données du fragment t qui ont comme centre le plus proche le centre j et la somme est faite sur les $\mathbf{x}_i$ du fragment t plus proches du centre j ;
6 Les paires $(j, (n_{jt}, \tilde{\mathbf{m}}_{jt}))$ sont groupées par j pour les tâches *Reduce* ;
7 Chaque tâche *Reduce* reçoit toutes les paires correspondant à une valeur de j, calcule $\mathbf{m}_j = \frac{\sum_t \tilde{\mathbf{m}}_{jt}}{\sum_t n_{jt}}$ et stocke le $\mathbf{m}_j$ résultant ;
8 **end**

Initialisation *K-means* : *K-means++*

- Une bonne initialisation de l'algorithme *K-means*
  - permet d'obtenir une solution de meilleure qualité et
  - une convergence plus rapide (avec moins d'itérations) vers cette solution
- Parmi les nombreux algorithmes d'initialisation nous considérerons *K-means++* [1]
- Idée : choisir les centres successivement, suivant une loi non uniforme qui privilégie les candidats éloignés des centres déjà sélectionnés

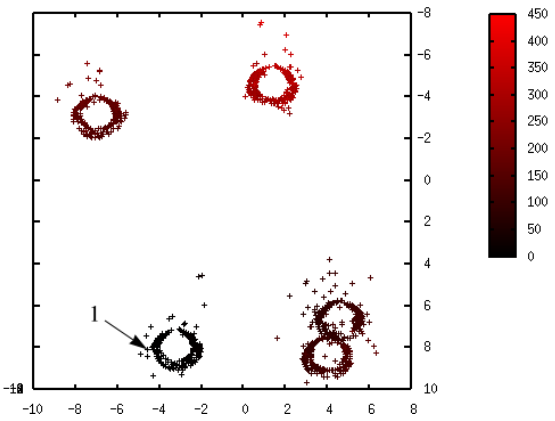
Data : Ensemble $\mathcal{E}$ de N données de $\mathbb{R}^p$; nombre souhaité de centres k
Result : $\mathcal{C} = \{\mathbf{c}_j\,, 1 \leq j \leq k\}$

1 $\mathcal{C} \leftarrow$ un $\mathbf{x}$ de $\mathcal{E}$ choisi au hasard ;
2 **while** $\|\mathcal{C}\| \leq k$ **do**
3 | Sélectionner $\mathbf{x} \in \mathcal{E}$ avec la probabilité $\frac{d^2(\mathbf{x},\mathcal{C})}{\phi_{\mathcal{E}}(\mathcal{C})}$;
4 | $\mathcal{C} \leftarrow \mathcal{C} \cup \{\mathbf{x}\}$;
5 **end**

- Notations : $d^2(\mathbf{x}\,,\mathcal{C}) = \min_{j=1,\ldots,t} d^2(\mathbf{x}\,,\mathbf{c}_j)$, $\phi_{\mathcal{E}}(\mathcal{C}) = \sum_{\mathbf{x}\in\mathcal{E}} d^2(\mathbf{x}\,,\mathcal{C})$
- Problème : *K-means++* n'est pas directement parallélisable

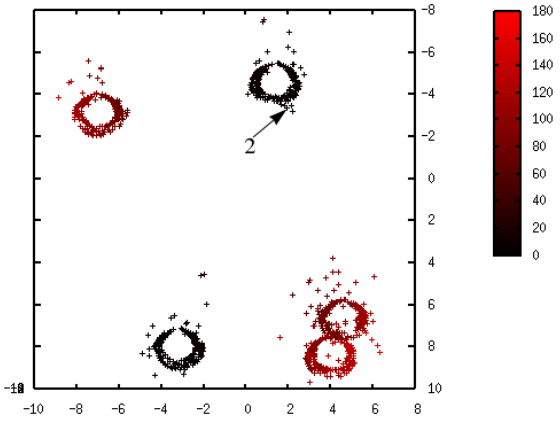
K-means++ : évolution des probabilités

(probabilité de sélection proportionnelle à $d^2(\mathbf{x}\,,\mathcal{C})$, représentée par la couleur rouge)



Après la sélection d'un point

$$\mathcal{C} = \left\{ \begin{pmatrix} -4,6 \\ 8,0 \end{pmatrix} \right\}$$

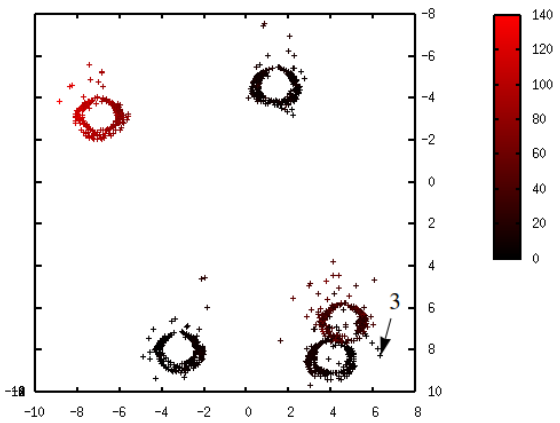


Après la sélection de 2 points

$$\mathcal{C} = \left\{ \begin{pmatrix} -4,6 & 2,15 \\ 8,0 & -3,45 \end{pmatrix} \right\}$$

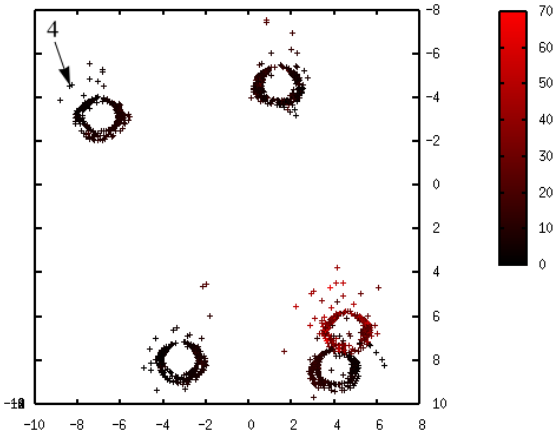
K-means++ : évolution des probabilités (2)

(probabilité de sélection proportionnelle à $d^2(\mathbf{x}, \mathcal{C})$, représentée par la couleur rouge)



Après la sélection de 3 points

$$\mathcal{C} = \left\{ \begin{pmatrix} -4,6 & 2,15 & 6,32 \\ 8,0 & -3,45 & 8,22 \end{pmatrix} \right\}$$



Après la sélection de 4 points

$$\mathcal{C} = \left\{ \begin{pmatrix} -4,6 & 2,15 & 6,32 & -8,37 \\ 8,0 & -3,45 & 8,22 & -4,54 \end{pmatrix} \right\}$$

Initialisation *K-means* parallélisable : *K-means||*

- *K-means||* [2] proposé comme variante parallélisable de *K-means++*
- Idée : choisir plus qu'un centre à chaque itération, mais suivant une loi non uniforme

Data : Ensemble $\mathcal{E}$ de N données de $\mathbb{R}^p$; nombre souhaité de centres k ; degré de sur-échantillonnage $l \sim \Omega(k)$ (l augmente au moins aussi vite que k)

Result : $\mathcal{C} = \{\mathbf{c}_j, 1 \leq j \leq k\}$

$\mathcal{C} \leftarrow$ un $\mathbf{x}$ de $\mathcal{E}$ choisi au hasard ;

$\psi \leftarrow \phi_{\mathcal{E}}(\mathcal{C})$;

for $O(\log \psi)$ *fois* **do**

  $\mathcal{C}' \leftarrow$ sélectionner chaque $\mathbf{x} \in \mathcal{E}$ indépendamment avec la probabilité $\frac{l \cdot d^2(\mathbf{x}, \mathcal{C})}{\psi}$;

  $\mathcal{C} \leftarrow \mathcal{C} \cup \mathcal{C}'$; $\psi \leftarrow \phi_{\mathcal{E}}(\mathcal{C})$;

end

for $\mathbf{m}_j \in \mathcal{C}$ **do**

  $w_m \leftarrow$ nombre de points de $\mathcal{E}$ plus proches de $\mathbf{m}_j$ que de tout autre point de $\mathcal{C}$;

end

Classification en k groupes des données $\mathcal{C}$ *pondérées par leurs poids* w_m ;

K-means|| : caractéristiques

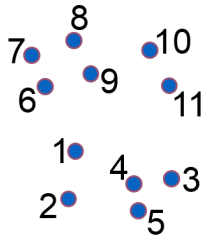
- A l'étape (4), $\sum_{\mathbf{x}\in\mathcal{E}} \frac{l \cdot d^2(\mathbf{x},\mathcal{C})}{\psi} = l$, donc à chaque itération env. l nouveaux points sont choisis
- L'étape finale traite un nombre réduit de données, $O(l \cdot \log \psi)$, elle peut donc se dérouler sur un seul nœud de calcul et employer *K-means++*
- Comme *K-means++*, *K-means||* donne des garanties de qualité de la solution obtenue (voir [2]) : $E[\phi_{\mathcal{E}}(\mathcal{C})] \leq O(\log k) \cdot \phi_{\mathcal{E}}(\mathcal{C}^*)$, E étant l'espérance et $\mathcal{C}^*$ la solution optimale
- D'après [2], après 5 itérations on atteint déjà une très bonne solution, il n'est pas nécessaire de faire $O(\log \psi)$ itérations

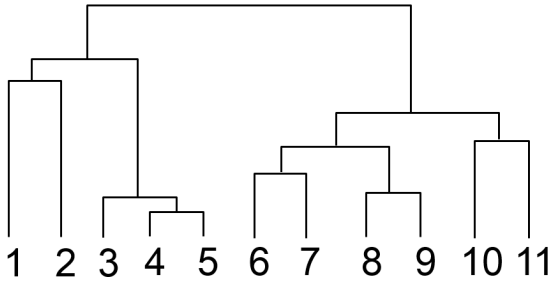
K-means|| : implémentation MapReduce

- Comme pour l'implémentation de *K-means*, après sa mise à jour dans l'étape (5), l'ensemble $\mathcal{C}$ des centres est transmis à tous les nœuds de calcul
- Le calcul de ψ dans l'étape (5) est réalisé comme pour l'implémentation de *K-means*, le résultat est transmis à tous les nœuds de calcul pour l'étape (4)
- Le calcul des $d^2(\mathbf{x}, \mathcal{C})$ dans l'étape (4) peut être fait par chaque nœud de calcul pour les $\mathbf{x}$ de son fragment
- Les tirages de l'étape (4) sont réalisés de façon indépendante par les nœuds de calcul
- Le calcul des poids w_m dans l'étape (8) est fait comme le calcul des n_j dans l'implémentation de *K-means*
- La classification des données de $\mathcal{C}$ dans l'étape (10) peut être faite sur un seul nœud de calcul avec *K-means++*

Classification ascendante hiérarchique (CAH)

- Objectif : obtenir une hiérarchie de groupes, structure plus riche qu'un simple partitionnement
- Permet d'examiner l'ordre des agrégations de groupes, les rapports des similarités entre groupes, etc.
- Classification ascendante : procède par agrégation des données et des groupes

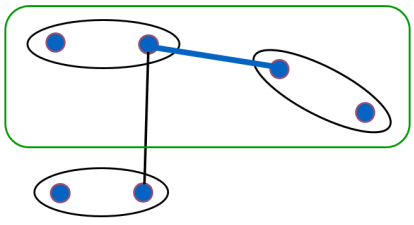




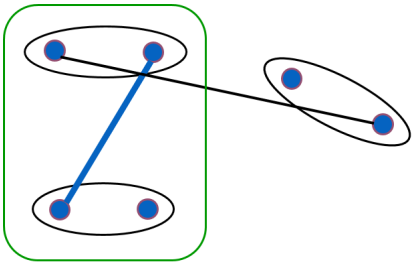
CAH : indices d'agrégation

- Sur la base de la distance entre données, $d_{\mathcal{X}} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$, différents indices d'agrégation peuvent être utilisés pour mesurer la dissimilarité entre groupes :

$$\delta_s(h_p, h_q) = \min_{x_i \in h_p, x_j \in h_q} d_{\mathcal{X}}(x_i, x_j) \qquad \delta_s(h_p, h_q) = \max_{x_i \in h_p, x_j \in h_q} d_{\mathcal{X}}(x_i, x_j)$$



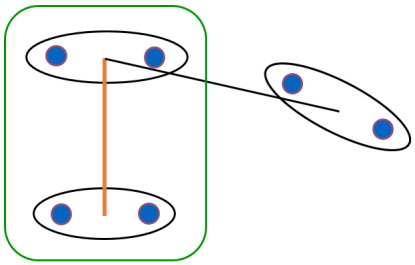
lien minimum (*single linkage*)



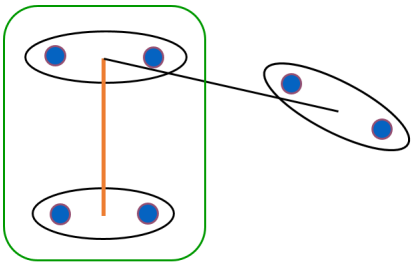
lien maximum (*complete linkage*)

CAH : indices d'agrégation (2)

$$\delta_s(h_p, h_q) = \frac{1}{\|h_p\| \cdot \|h_q\|} \sum_{x_i \in h_p, x_j \in h_q} d_{\mathcal{X}}(x_i, x_j) \qquad \delta_s(h_p, h_q) = \frac{\|h_p\| \cdot \|h_q\|}{\|h_p\| + \|h_q\|} d^2_{\mathcal{X}}(\mathbf{m}_p, \mathbf{m}_q)$$



lien moyen (*average linkage*)



indice de Ward (données vectorielles !)

CAH : algorithme, mise en œuvre

Data : Ensemble $\mathcal{E}$ de N données de $\mathcal{X}$ muni de la distance $d_{\mathcal{X}}$
Result : Hiérarchie de groupes (dendrogramme)

```
1 Chaque donnée définit un groupe ;
2 while nombre de groupes > 1 do
3 |   Calcul indices d'agrégation entre tous les groupes issus de l'itération précédente ;
4 |   Regroupement des 2 groupes ayant la plus petite valeur de l'indice d'agrégation ;
5 end
```

- Complexité algorithmique $O(N^2 \log N)$!
- N élevé : application de *K-means* avec nombre de groupes (k) élevé (mais $k \ll N$), ensuite application de la CAH sur les groupes obtenus par *K-means*
- Avec *single linkage*, la CAH est équivalente à la recherche de l'arbre couvrant de poids minimal $\rightarrow$ parallélisation par découpage de $\mathcal{E}$ + calcul dans chaque partition + fusion des résultats

La classification automatique dans Spark

- *K-means* : initialisation par *K-means*|| [2]
- Estimation de mélanges gaussiens (méthode d'estimation de densité) :
  - Ajout de variables auxiliaires et estimation par espérance-maximisation (*Expectation-Maximization*, EM)
  - EM : algorithme itératif, à chaque itération étape de calcul de l'espérance de la vraisemblance suivie d'étape de calcul des paramètres pour maximiser la vraisemblance
  - On obtient des équations de mise à jour facilement parallélisables
- *Bisecting k-means* : méthode de classification *descendante* hiérarchique (partitionnement récursif de chaque *cluster*)
- *Power Iteration Clustering* (PIC [4]) : simplification de la classification spectrale, travaille sur la matrice des similarités entre données
- *Latent Dirichlet Allocation* (LDA [3]) : identification de « thèmes » (*topics*) dans un ensemble de documents textuels, « explication » de chaque document par un ou plusieurs thèmes

Références I


D. Arthur and S. Vassilvitskii. K-means++ : The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, pages 1027–1035, Philadelphia, PA, USA, 2007. Society for Industrial and Applied Mathematics.


B. Bahmani, B. Moseley, A. Vattani, R. Kumar, and S. Vassilvitskii. Scalable k-means++. *Proc. VLDB Endow.*, 5(7) :622–633, 2012.


D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3 :993–1022, Mar. 2003.


F. Lin and W. W. Cohen. Power iteration clustering. In J. Fürnkranz and T. Joachims, editors, *ICML*, pages 655–662. Omnipress, 2010.