Apprentissage non supervisé

19 janvier

Dans l'apprentissage non supervisé, on dispose d'un jeu de données non étiquetées et on cherche à dégager les similitudes pouvant exister entre certaines de ces données en les regroupant en différentes catégories : on cherche à les partitionner. On appelle cela le *clustering*, parfois (mal ?) traduit en classification (mais à ne pas confondre avec la classification dont on a parlé dans le chapitre précédent).

L'objectif d'un clustering est de créer des classes d'équivalence où on maximise l'homogénéité au sein d'une classe, et on maximise l'hétérogénéité entre différentes classes. Pour quantifier cette idée, on utilisera une fonction de distance qui pourra varier selon les données utilisées.

1 Clustering hiérarchique ascendant

Le clustering hiérarchique ascendant (ou CHA) consiste à regrouper les données petit à petit en fonction de leur proximité avec d'autres données. L'approche est relativement simple et peut se résumer en :

Entrée : ensemble de données $x_1, \dots, x_N$.
Début algorithme
  Créer N classes $\{x_1\}, \dots, \{x_n\}$.
  Tant que nécessaire **Faire**
    Fusionner les deux classes les plus proches.

Dans l'algorithme précédent, certaines choses sont implicites :

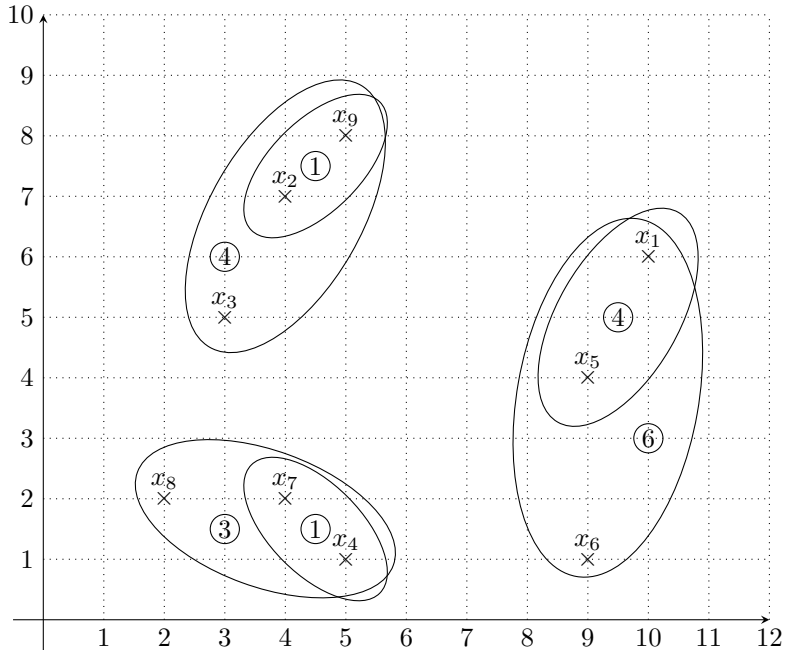
— La condition « nécessaire » de la boucle `Tant que` peut avoir différentes interprétations, on peut choisir d'arrêter les regroupements lorsque :
  - un nombre de classes déterminé à l'avance a été atteint ;
  - les deux classes les plus proches sont suffisamment éloignées l'une de l'autre ;
  - le diamètre d'une des classes devient trop important.

— Par ailleurs, la notion de « plus proches » est volontairement vague. On peut obtenir des résultats différents selon la distance $\Delta(A, B)$ choisie entre deux classes A et B :
  - liaison simple : $\min\{\delta(x, y) \mid x \in A, y \in B\}$;
  - liaison complète : $\max\{\delta(x, y) \mid x \in A, y \in B\}$;
  - liaison moyenne : $\dfrac{1}{|A||B|} \displaystyle\sum_{x \in A, y \in B} \delta(x, y)$;
  - liaison barycentrique : $\delta(b_A, b_B)$ où b_A et b_B sont les barycentres de A et B ;
  - liaison de Ward : $\sqrt{\dfrac{|A||B|}{|A| + |B|}}\delta(b_A, b_B)$, correspondant à une liaison barycentrique en donnant moins de poids aux classes composées de données isolées ;
  - ...

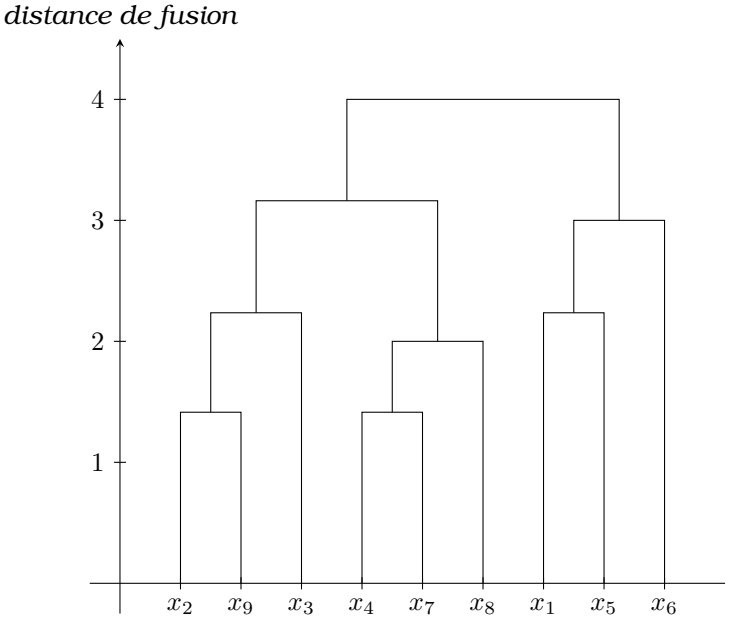
Exemple 1. *On considère l'ensemble de points suivant :*

$$x_1(10, 6), x_2(4, 7), x_3(3, 5), x_4(5, 1), x_5(9, 4), x_6(9, 1), x_7(4, 2), x_8(2, 2), x_9(5, 8)$$

On souhaite appliquer l'algorithme CHA par liaison simple, avec une distance maximale avant fusion de deux clusters de 3*. On obtient les fusions successives suivantes (les cas d'égalité sont simultanés) :*



On peut représenter l'ordre des fusions, ainsi que les distances au moment de la fusion par un ***dendrogramme*** *(qu'on a continué ici jusqu'à la fusion de l'ensemble des classes) :*



Remarque 1. *Bien que simple à appréhender, le CHA possède deux inconvénients :*

- *il n'y a pas de « remise en question » : deux classes fusionnées ne seront jamais séparées ;*
- *la complexité peut être élevée : la recherche des deux classes les plus proches peut prendre un temps $O(N^2)$, ce qui donne une complexité totale en $O(N^3)$ (et encore, seulement si les structures de données sont optimisées).*

Exemple 2. *Reprendre l'exemple précédent et déterminer le dendrogramme correspondant à un CHA avec liaison complète.*

2 k-MOYENNES

L'algorithme des k-moyennes est un algorithme qui cherche à construire un nombre de clusters défini à l'avance appelé k dans la littérature. L'objectif est de résoudre le problème d'optimisation `CLUSTERING` :

- ⋆ **Instance** : un ensemble de données $E = \{x_1, \ldots, x_N\}$ et un entier k.
- ⋆ **Solution** : une partition de E en k classes $C_1, \ldots, C_k$.
- ⋆ **Optimisation** : minimiser $\sum_{j=1}^{k} \sum_{x \in C_j} \delta(x, b_j)^2$, où b_j est le barycentre de la classe C_j.

Cependant, ce problème d'optimisation est NP-difficile donc l'algorithme présenté ici est une heuristique considérée comme satisfaisante mais qui peut renvoyer un résultat non optimal. L'idée de l'algorithme des k-moyennes est la suivante :

— créer k points $b_1, \ldots, b_k$, soit au hasard dans l'espace considéré, soit choisis au hasard parmi les x_i ;
— tant que les b_j changent :
  — affecter à chaque point x_i la classe j correspondant au barycentre b_j qui lui est le plus proche ;
  — modifier chaque b_j en le barycentre de la nouvelle classe j.

Proposition 1. *Chaque itération de la boucle* `Tant que` *de l'algorithme précédent a une complexité en* $O(k \times N)$.

Proposition 2. *L'algorithme des k-moyennes termine.*

Remarque 2. *L'algorithme des k-moyennes, bien que généralement plus rapide que l'algorithme de clustering hiérarchique ascendant présente certains défauts :*

— *Le choix initial des b_j est un point critique de l'algorithme. Un mauvais choix peut entraîner un grand déséquilibre par rapport à une solution optimale (si les points choisis sont éloignés des données réelles, ou si deux points sont initialement choisis très proches).*
— *Il peut converger vers un minimum local qui n'est pas un minimum global.*
— *En pratique, on fixe un nombre limite d'itérations plutôt que d'attendre la convergence.*
— *il n'est valable que pour des classes convexes ce qui n'est pas le cas de CHA.*

3 EXERCICE

On considère l'ensemble de points :

$$x_1(2,10), x_2(2,5), x_3(8,4), x_4(5,8), x_5(7,5), x_6(6,4), x_7(1,2), x_8(4,9)$$

1. Faire une figure précise.
2. En supposant qu'on souhaite obtenir 3 classes, appliquer l'algorithme des k-moyennes avec des barycentres initialisés à $\{x_1, x_4, x_7\}$.
3. En supposant qu'on arrête les fusions lorsque la distance entre deux clusters dépasse 4, appliquer l'algorithme CHA par liaison simple.

On mesurera une approximation des distances à la règle.