

Licence d'Informatique 2

Analyse de Données Utilisateur (C5-160412)

TD 4 – Apprentissage Non Supervisé (*Clustering*)

Carl FRÉLICOT – Dpt Info / Lab MIA

Un tableau de données consiste en n individus pour lesquels on a observé p variables ; si elles sont quantitatives, il correspond à un nuage de n points dans un espace de dimension p . À la notion de *proximité* entre deux individus du tableau correspond naturellement la notion de *distance* entre deux points dans l'espace. Alors, on peut facilement obtenir un tableau de distances entre individus, symétrique, de taille $n \times n$, dont les valeurs sur la diagonale sont nulles. Parfois, les données ne sont pas directement accessibles et seul un tableau de distances (ou de *similarités*) est à disposition de l'analyste.

L'objectif des méthodes de classification automatique (*clustering*), appelée aussi *classification non supervisée*, est de déterminer si les données possèdent une structure de *groupes* (ou *clusters*) de sorte qu'on puisse leur associer une variable indicatrice¹. On cherche donc une partition des données telle que deux points d'un même groupe sont plus *proches* que deux points de groupes différents. Certaines méthodes nécessitent une mesure de distance D entre groupes en plus d'une distance d entre individus.

1. Distances entre individus

- Considérons deux points en dimension deux $x = (1, 4)$ et $y = (3, 5)$. Que vaut la distance $d(x, y)$ que vous connaissez ?
- À votre avis, pourquoi utilise-t-on plutôt $d^2(x, y)$?
- L'extension à plus de deux dimensions est immédiate. Calculez la distance entre $x = (0, 1, 4, 5)$ et $y = (1, 3, 5, 0)$. Cette distance est appelée *distance euclidienne usuelle* ; il en existe beaucoup d'autres...
- Dans le cas de données binaires, plusieurs distances usuelles existent :
 - distance de *Jaccard* $d_J = 1 - \frac{b_{11}}{b - b_{00}}$ où b est le nombre de bits, b_{11} (b_{00}) est le nombre de 1 (0) en commun
 - distance de *Hamming* $d_H = \frac{b_{10} + b_{01}}{b}$, c'est à dire le % de 0 et de 1 qui diffèrent
 Calculez ces distances entre $x = 0111000010$ et $y = 0101100011$.
- L'extension aux données qualitatives est naturelle. Soient par ex. deux séquences d'ADN dont on a extrait des morceaux $x = CTTAGGATAG$ et $y = GTATGGATTG$, où chaque lettre symbolise une *base* (A,C,G,T). Calculez la distance de Hamming entre x et y .

2. Méthodes de Partitionnement

La recherche de la meilleure (au sens d'un critère J) partition en m groupes d'un ensemble de n objets ne peut pas se faire de manière exhaustive car le nombre de possibilités est hautement combinatoire. Ci-contre sont donnés des exemples du nombre S^a de partitions différentes en m groupes que l'on peut former à partir d'un ensemble de n objets.

m	n	$S(m, n)$
2	4	7
2	10	511
5	10	42 535
2	30	536 870 911
3	150	$> 6 \times 10^{70}$

- Calculez ce nombre pour $m = 2$ et $n = 4$.

^a $S(m, n) = \frac{1}{m!} \sum_{i=1}^m (-1)^{m-i} \binom{m}{i} i^n$; si m et n sont grands, on peut l'approximer par $S(m, n) \simeq \frac{m^n}{m!}$

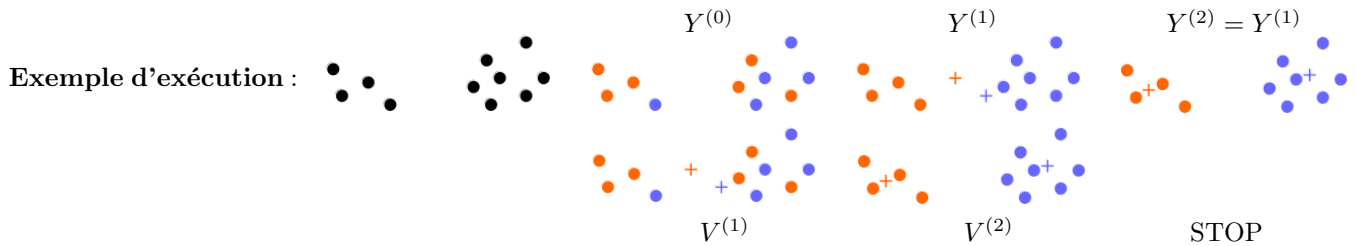
Notations :

- un tableau de données X de taille (n, p)
- une indicatrice de groupe Y de taille $(n, 1)$ telle que $Y_i = j$ si x_i est associé au groupe j , $j \in \{1, 2, \dots, m\}$
- un tableau de m représentants V , par exemple $V = [\bar{x}_1; \bar{x}_2; \dots; \bar{x}_m]$ de taille (m, p)

Le problème de *clustering* consiste donc, à partir de X , à déterminer (Y, V) qui optimisent un critère $J_m(Y, V)$. L'algorithme le plus basique est l'algorithme des *centres mobiles* (*k-means*) qui minimise l'*inertie intra-groupes* de la partition : $J_m(Y, V) = \frac{1}{n} \sum_{j=1}^m \sum_{x_i: Y_i=j} d^2(x_i, \bar{x}_j)$. Étant donné un tableau de données X , une distance d choisie et un nombre de groupe m fixé par l'utilisateur, il consiste à alterner deux phases :

- initialiser $V^{(0)} = \{\bar{x}_1, \bar{x}_2, \dots, \bar{x}_m\}$ ou bien la partition $Y^{(0)}$,
- à l'itération t et alors intervertir les deux phases
 - $Y^{(t)} = \argmin_Y J_m(Y, V^{(t-1)})$
 - $V^{(t)} = \argmin_V J_m(Y^{(t)}, V)$
 jusqu'à stabilité convergence vers un minimum local

¹ la variable indicatrice résultat pourra ensuite être utilisée comme une variable catégorielle dont les modalités sont l'appartenance aux groupes obtenus par apprentissage automatique, et pourra se substituer à celle donnée par un expert du domaine, à des fins de prédiction par *classification supervisée*



Exercice 1

Soit le tableau de données X ci-contre au sein duquel on souhaite découvrir $m = 2$ groupes. On vous propose plusieurs exécutions de l'algorithme *k-means* dont les résultats sont à compléter. Vous pourrez vider les tableaux ci-dessous et faire les calculs permettant de réaliser les exécutions complètes.

	Ind. / Var.	X1	X2
1	x	0.0	1.0
2	y	1.0	4.0
3	z	4.0	5.0
4	t	5.0	0.0

(a) $Y^{(0)} = [1, 2, 1, 2] \rightarrow V^{(1)} = [\bar{x}_1 = (2, 3), \bar{x}_2 = (\quad , \quad)] \rightarrow$

$d^2(\bar{x}_j, x_i)$	x	y	z	t
\bar{x}_1	8		8	18
\bar{x}_2	10		10	8
$Y^{(1)}$	1		1	2

\rightarrow

$V^{(2)} = [\bar{x}_1 = (\quad , \quad), \bar{x}_2 = (5, 0)] \rightarrow$

$d^2(\bar{x}_j, x_i)$	x	y	z	t
\bar{x}_1	8.22	0.89	8.22	22.22
\bar{x}_2	26	32	26	
$Y^{(2)}$	1	1	1	

\rightarrow STOP

(a') $Y^{(0)} = [1, 2, 1, 1] \rightarrow V^{(1)} = [\bar{x}_1 = (3, 2), \bar{x}_2 = (\quad , \quad)] \rightarrow$

$d^2(\bar{x}_j, x_i)$	x	y	z	t
\bar{x}_1	10	8	10	
\bar{x}_2	10	0	10	
$Y^{(1)}$ [tie]	2	2	1	

\rightarrow

$V^{(2)} = [\bar{x}_1 = (4.5, 2.5), \bar{x}_2 = (0.5, 2.5)] \rightarrow$

$d^2(\bar{x}_j, x_i)$	x	y	z	t
\bar{x}_1	22.5	14.5	6.5	
\bar{x}_2	2.5	2.5	18.5	26.5
$Y^{(1)}$	2	2	1	

\rightarrow STOP

1-1) Les partitions finales obtenues à partir de (a) et (a') sont différentes. Qu'en pensez-vous ?

Vous pourrez dessiner le nuage de points...

1-2) Le tableau ci-contre décrit toutes les partitions possibles en termes de leur inertie :

- intra-groupes $I_W = \frac{1}{n} \sum_{j=1}^m \sum_{x_i: Y_i=j} d^2(x_i, \bar{x}_j)$
- inter-groupes $I_B = \frac{1}{n} \sum_{j=1}^m n_j d^2(\bar{x}, \bar{x}_j)$
- totale $I_T = I_W + I_B = \frac{1}{n} \sum_{i=1}^n d^2(x_i, \bar{x})$

Complétez-le, par déduction.

1-3) Calculez le plus simplement possible les inerties intra-groupes et inter-groupes des partitions finales obtenues en (a) et (a'), à savoir Y_4 et Y_5 .

1-4) Calculez la distance *cosinus* entre les deux variables.

HW) Chez vous, vous calculerez leur distance *corrélation*.

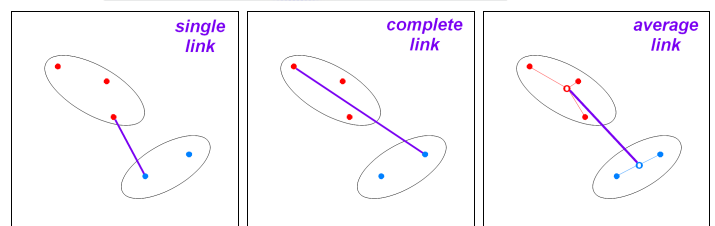
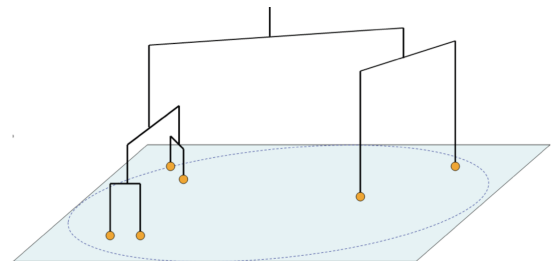
Partition	<i>intra</i>	<i>inter</i>	<i>totale</i>
$Y_1 = [1, 2, 2, 2]$	5.67	2.83	8.5
$Y_2 = [2, 1, 2, 2]$	7		
$Y_3 = [2, 2, 1, 2]$	5.67	2.83	
$Y_4 = [2, 2, 2, 1]$	4.33	4.17	
$Y_5 = [1, 1, 2, 2]$		4	
$Y_6 = [1, 2, 1, 2]$		0.5	
$Y_7 = [1, 2, 2, 1]$	4.5		

3. Méthodes Hiérarchiques

Il s'agit de faire émerger une structure hiérarchique de groupes de points, généralement de manière ascendante (*Hierarchical Agglomerative Clustering*) : au niveau 0, les n groupes-singletons, et au niveau $n - 1$ un seul groupe. À chaque niveau, on regroupe les deux groupes les plus proches, au sens d'une distance \mathcal{D} entre groupes choisie par l'utilisateur qui indexe la hiérarchie. Par ex., si on note C_i et C_j deux groupes :

- $\mathcal{D}_{\min}(C_i, C_j) = \min_{x \in C_i, y \in C_j} d(x, y)$ (single)
- $\mathcal{D}_{\max}(C_i, C_j) = \max_{x \in C_i, y \in C_j} d(x, y)$ (complete)
- $\mathcal{D}_{\text{moy}}(C_i, C_j) = \frac{\sum_{x \in C_i} \sum_{y \in C_j} d(x, y)}{n_i \times n_j}$ (average)

où d est une distance entre individus, elle aussi choisie.



- (a) Qu'obtient-on en coupant la hiérarchie à un niveau particulier ?
- (b) Si on souhaite obtenir une partition, à quel niveau semble-t-il judicieux de couper la hiérarchie ?
- (c) Comment évolue l'inertie intra-groupes lorsqu'on passe d'un niveau à un niveau plus élevé ?
- (d) À quoi se réduisent \mathcal{D}_{min} , \mathcal{D}_{max} et \mathcal{D}_{moy} pour des groupes-singletons ?

Exercice 2

Les figures ci-dessous montrent différentes hiérarchies obtenues sur les données de l'exercice précédent avec la distance d euclidienne usuelle et \mathcal{D}_{min} , \mathcal{D}_{max} et \mathcal{D}_{moy} (de gauche à droite) :



- 2-1) Quel(s) commentaire(s) ces hiérarchies vous inspirent ?
- 2-2) À quel niveau couper ces hiérarchies pour obtenir la meilleure partition ?
- 2-3) Si on souhaite obtenir une partition en deux groupes, retrouve-t-on les bonnes ?

La méthode de *Ward* est très populaire car on peut lui attacher un critère familier : à chaque niveau, les deux groupes les plus proches au de \mathcal{D}_W sont ceux qui minimisent l'accroissement d'inertie intra-groupes si d est la euclidienne usuelle. On obtient alors : $\mathcal{D}_W(C_i, C_j) = \frac{p_i \times p_j}{p_i + p_j} d^2(\bar{x}_i, \bar{x}_j)$, où p_i et p_j représentent les poids des groupes.

- (a) À quoi se réduit \mathcal{D}_W pour des groupes-singletons ?
- (b) Calculez $\mathcal{D}_W(\{x\}, \{y\})$
- (c) **Alerte :**

La hiérarchie de Ward donnée par OrangeTM est visualisée ci-contre. La valeur calculée pour le regroupement des singletons $\{x\}$ et $\{y\}$ est : 3.1623
Quelle erreur a été commise ?

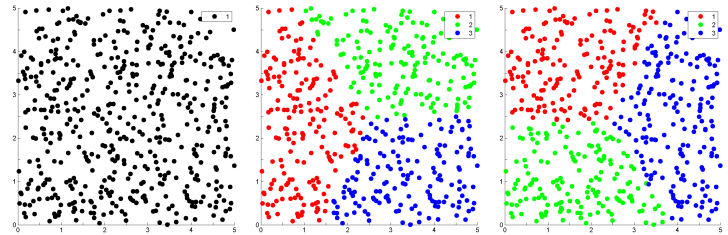


- (d) La hiérarchie est-elle cependant intéressante ?

Outre le choix des méthodes ou leur paramétrage, celui de la distance, d'autres problématiques existent.

4. Problèmes Connexes

- (a) Si on cherche une structure, un algorithme en trouve une.. mais les données ont-elles a priori une structure ?
- (b) Pour les méthodes de partitionnement, comment choisir le nombre k de groupes ?
- (c) Comment valider a posteriori un résultat ?
- (d) Comment comparer plusieurs résultats ?
- (e) Est-il possible de chercher non pas des groupes de points-individu mais des groupes de points-variable ? Dans quel but ?
- (f) Ne serait-il pas plus judicieux d'utiliser, à chaque itération, une assignation non stricte aux groupes ?



...