Correction - Examen final

Clustering, Classification et Règles d'Association

le 12 décembre 2024

1pt d'office

1. **Clustering** 10pts au total

   (a) **Code** 4pts

   Soit `X` un dataframe en R. Que fait le code suivant :

```
n <- dim(X)[1]
indice <- sample(1:n, n)
X2 <- X[indice,]

kcl <- kmeans(X, k = 3)
kcl2 <- kmeans(X2,k = 3)
table(kcl$cluster[indice], kcl2$cluster)

d <- dist(X)
d2 <- dist(X2)
hcl <- cutree(hclust(d=d, method='single'), k=3)
hcl2 <- cutree(hclust(d=d2, method='single'), k=3)
table(hcl[indice], hcl2)

dcl <- dbscan(data = X, eps = 0.2, MinPts = 3)
dcl2 <- dbscan(data = X2, eps = 0.2, MinPts = 3)
table(dcl$cluster[indice], dcl2$cluster)
```

   A quoi s'attendre comme résultats pour chacune des fonctions `table` ?

   Dans `X2` on obtient une permutation d'observations (lignes) de `X`. La permutation est gardée dans le tableau `indice`.

   Pour `X` et `X2` on applique les mêmes algorithmes de clustering et on compare les résultats. Pour comparer correctement on fait la permutation via `indice` du résultats pour `X`.

   Pour k-means il est très peu probable d'obtenir les mêmes résultats. Si il y a 3 clusters bien séparés on peut avoir la même solution calculée, modulo une autre numérotation des clusters, sinon, le résultat de `table` est très loin d'une matrice diagonale.

   Pour le clustering hiérarchique idem, on peut obtenir des résultats différents.

   Pour DBSCAN on obtient strictement les mêmes points isolés et les mêmes clusters ; pour les clusters de même taille leur numéro d'ordre peut changer.

(b) **Calculs** 6pts : 2 pts pour kmeans (1pt le pas 1 correct, 1 pt la bonne suite du calcul), CHA : 4 pts (2 pts pour les clusters corrects, 2pt pour le dendogramme correct

Pour les données représentées dans la figure dans l'espace $\mathbb{R}^2$ avec la distance euclidienne, mettez en pratique deux algorithmes de clustering :

— k-moyennes avec les centres initiaux c et k

— hiérarchique ascendant avec la stratégie de la distance maximale.

Travaillez directement sur la dernière feuille que vous détacherez et que vous allez glisser dans la double copie

Pour k-means :

— au pas 0 : les centres son $(6, 5)$ et $(7, 5)$ ce qui partage les noeuds en deux clusters $\{a, b, c, d, e, f, i\}$ et $\{g, h, k, j\}$

— au pas 1 : les centres sont : $(4.285714, 5.00)$ et $(7.75, 3.75)$ et la séparation en clusers ne change pas

— on peut arêter le calcul

Voir la figure 1

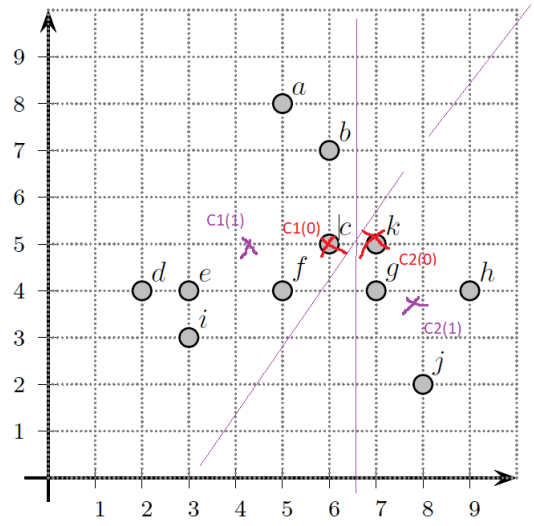

FIGURE 1 – Résultat k-means

Pour le CHA avec la stratégie max, voir la figure 2.

2. **Itemsets fréquents et règles d'association** 3pts : une valeur correcte et justifuée des RA (1pt), bonne solution pour confiance = 1 (1pt), généralisation (1pt)

Soit X_1, X_2, X_3 un 3-itemset fréquent. On essaie de générer toutes les règles d'association à partir de cet itemset.

(a) combien il y en a (sans regarder la confiance) ? Si les RA ont 3 items, on a au total $2 \times C_3^2 = 6$. Si les RA ont ont 2 items, il y en $2 \times C_3^1 = 6$. un calcul pour les RA à 3 items suffit.

(b) Si on ne cherche que les règles d'association "très fortes", de confiance = 1, avons nous besoin de les générer toutes les règles possibles et calculer expressément la confiance ? Généraliser le raisonnement pour un k-itemset avec un k quelconque.

Si $U \rightarrow V$ est une RA de confiance 1, alors $support(U) = support(UV)$. Donc U n'est pas un itemset fermé et on cherche un itemset UV de même support.

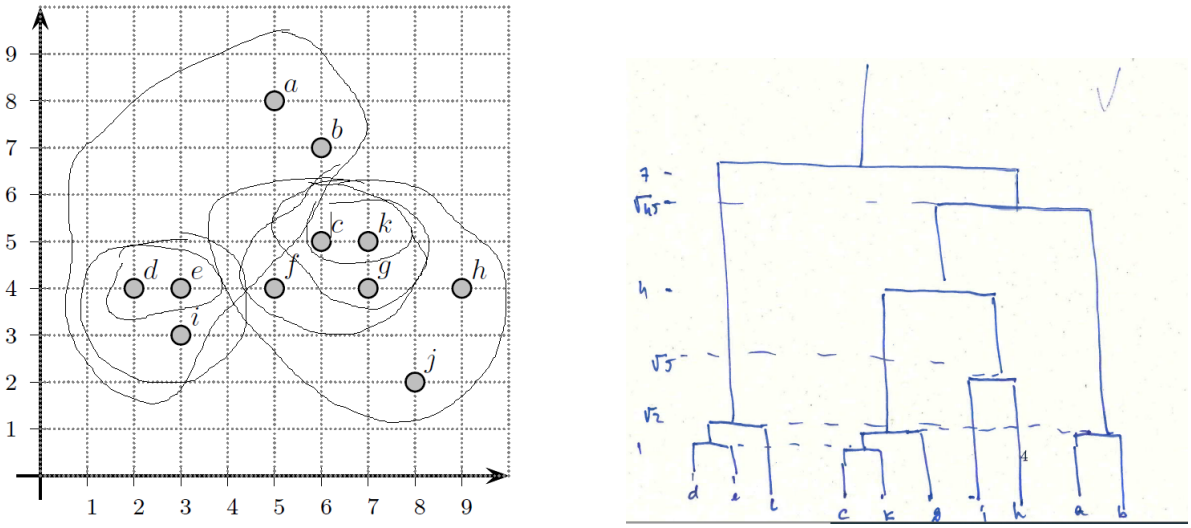

FIGURE 2 – Résultat CHA distance maximum

3. **Calcul d'itemsets fréquent et de règles d'association**

— 10pts pour le calcul fait avec $support \geq 3$
— 6pts pour $support > 3$

Soit la base de transactions suivante :

TID	items
T1	C, H, A, T
T2	C, A, T
T3	C, H, T
T4	A, H
T5	C, A, T, H
T6	T, H, E
T7	H, T, E, A
T8	H, A, E

(a) Détaillez le déroulement d'un algorithme de calcul des itemsets fréquents pour

$$min\_support\_count = 3$$

Listez explicitement les itemsets fréquents obtenus.

Le $min\_support\_count$ est pris avec la limite 3 (sinon, très peu de calculs). Toutefois, si on a interprété la limite comme stricte ($> \text{"} >$), on attend une résolution parfaite. Dans la correction, je cherche la mise en oeuvre des élagages ou autre procédés qui limitent les calculs.

Pour $support \geq 3/8$ on obtient :

```
items      support count
{E}        0.375   3
{C}        0.500   4
*{A}        0.750   6
*{T}        0.750   6
*{H}        0.875   7
*{E, H}     0.375   3
{A, C}     0.375   3
```

```
*{C, T}    0.500   4
{C, H}    0.375   3
*{A, T}    0.500   4
*{A, H}    0.625   5
*{H, T}    0.625   5
*{A, C, T} 0.375   3
*{C, H, T} 0.375   3
*{A, H, T} 0.375   3
```

Pour $support > 3/8$ on obtient :

```
items     support count
{C}    0.500   4
*{A}    0.750   6
*{T}    0.750   6
*{H}    0.875   7
*{C, T} 0.500   4
*{A, T} 0.500   4
*{A, H} 0.625   5
*{H, T} 0.625   5
```

(b) Mettez en évidence les itemsets fermés.

ils sont marqués d'un `*`.

(c) Générez les règles d'association très fortes avec $min\_confiance = 1$, calculez aussi le $LIFT$.

Ce n'était pas la peine de tout générer. Il suffisait de prendre à gauche un itemset pas fermé et à droite l'itemset obtenu par la différence d'un itemset plus grand qui le contient et qui a le même support. Donc ceux sans `*` dans la liste ci-dessus.

Pour $support \geq 3$:

```
E -> H 1.14
C -> T 1.33
A C -> T 1.33
C H -> T 1.33
```

Pour $support > 3$:

```
C -> T 1.33
```