

Examen final

UP3 - clustering, classification, règles d'association

le 5 janvier 2020, durée 1h

1pt d'office

1. (4pt : 3pt justification modification du support + 1pt summary) Un data scientist travaille sur un ensemble de transactions `DonneesPaniers` en R. L'ensemble est correctement construit, toutefois lors de l'exécution de l'algorithme **Apriori** le résultat ne contient aucune règle d'association :

```
rules.all <- apriori(DonneesPaniers)
inspect(rules.all)
```

(l'ensemble `rules.all` est vide.)

Que faut-il faire ?

Il faut adapter, bien entendu, le `min_support`, à la rigueur le seuil de confiance mais pas en dessus de 0,6 (60%).

Pour faire ça il convient d'examiner les fréquences de d'apparition d'item avec la commande `summary(DonneesPaniers)`.

2. (6pt) Soit le petit ensemble de transactions :

items	transactionID
[1] {Burger, Buns, Ketchup}	T1
[2] {Burger, Buns}	T2
[3] {Burger, Coke, Chips}	T3
[4] {Chips, Coke}	T4
[5] {Chips, Ketchup}	T5
[6] {Burger, Coke, Chips}	T6

- (a) (4,5 pt : 1,5pt déroulé de l'algo, 1,5 pt le résultat correct, 0,5 pt pour l'affichage in extenso des itemsets fréquents) à l'aide de

l'algorithme Apriori ou de l'algorithme ECLAT calculez tous les itemsets fréquents pour un $min_support = 0,334$ (supérieur donc à $1/3$).

Un des algos Apriori ou ECLAT devait être déroulé pas à pas.

Les itemsets fréquents au-delà de $0,334$ sont : *Burger*, *Chips*, *Coke* et $\{\text{Chips}, \text{Coke}\}$.

- (b) (1,5 pt : 0,75pt pour le bon résultat, 0,75 pt pour explorer uniquement l'itemset $\{\text{Chips}, \text{Coke}\}$ déterminez toutes les règles d'association ayant une confiance plus grande que $8/10$.

Si on considère les RA triviales de type *Coke* $\rightarrow \emptyset$, ce n'est pas pénalisé.

Le seul itemset qui engendre des RA non triviales est $\{\text{Chips}, \text{Coke}\}$. On peut générer deux RA :

$R1 : \text{Chips} \rightarrow \text{Coke}$, $support(R1) = 3/6 = 0,5$, $confiance(R1) = support(\text{Chips}, \text{Coke})/support(\text{Coke}) = 3/3 = 1$

$R2 : \text{Coke} \rightarrow \text{Chips}$, $support(R2) = 3/6 = 0,5$, $confiance(R2) = support(\text{Chips}, \text{Coke})/support(\text{Chips}) = 3/4 = 0,75 < 0,8$

L'unique RA retenue : $\text{Chips} \rightarrow \text{Coke}$

3. 9pt = 2pt + 2pt + 5pt Outlier.

Un outlier (donnée aberrante) est une observation qui se trouve à grande distance de toutes les autres données.

- (a) parmi les algorithmes de clustering lesquels vous semblent plus adaptés pour détecter les outliers ?

On peut utiliser DBSCAN avec un ϵ assez grand et un $NbPts$ quelconque. Toutefois les points isolés ne sont pas tous des outliers si ϵ est petit.

On peut également utiliser le clustering hiérarchiques, les outliers apparaîtront intégrés dans le cluster final à la fin avec des distances ou coefficients énormes.

- (b) quels algorithmes de clustering sont affectés par la présence des outliers ?

K-means est particulièrement sensible à la présence d'outlier, si un cluster se forme avec un unique point celui-ci clairement un outlier, si le cluster contient des "bons" points et un outlier le centre du cluster sera déplacé (attiré) par le outlier.

- (c) On vous fournit des données séparées en 3 classes pour faire de la classification et l'algorithme k-NN semble bien fonctionner. Toutefois parmi les données à traiter (à classifier) par la suite il est possible que des outliers se glissent et qu'ils n'appartiennent vraiment à aucune des 3 classes. Comment les identifier ?

J'attendais un algorithme clair capable d'identifier l'outlier et de faire aussi au passage du k-NN si pas outlier

Notons X les données séparées en trois classes et C_1, C_2, C_3 les classes (bien évidemment $C_1 \cup C_2 \cup C_3 = X$ et $C_i \cap C_j = \emptyset$ pour $i \neq j$).

Notons Y l'ensemble des données correctes sur lequel l'algorithme k-NN semble bien fonctionner. Imaginons que la valeur précise k est bien connue pour cet algorithme (le nombre de plus proche voisins pris en considération).

Pour chaque élément $\alpha \in Y$ correctement classifié dans une classe i on détermine :

$$d(\alpha) = \max\{dist(\alpha, y) | y \in C_i \text{ et } y \text{ est un } k \text{ proche voisin}\}$$

On calcule ensuite δ le seuil d'outlier :

$$\delta = \max_{\alpha \in Y} d(\alpha)$$

Pour tout nouvel point z à classifier, qui peut être aussi un outlier, on détermine les k plus proches voisins ; si ces voisins sont situés au-delà de δ par rapport à z , c'est un outlier, sinon, on rend le résultat de k-NN.