

Rédigé par : Equipe pédagogique

A l'intention de : Elèves ING1-GI

Durée : 3h00

## Exercice 1

Considérons les transactions d'une épicerie suivantes

Transactions	items
T1	HotDogs, Buns, Ketchup
T2	HotDogs, Buns
T3	HotDogs, Coca, Chips
T4	Chips, Coca
T5	Chips, Ketchup
T6	HotDogs, Coca, Chips

- 1) Combien y a-t-il de sous-ensembles candidats ?
- 2) On fixe une fréquence minimale à  $S_0=30\%$ . Calculer la fréquence des items. Doit-on tous les garder pour la suite de l'algorithme ?
- 3) Combien y a-t-il de sous-ensembles candidats de cardinal 2. Quels sont ceux qui ont une fréquence supérieure à 30% ?
- 4) Combien y a-t-il de sous-ensembles candidats de cardinal 3. Quels sont ceux qui ont une fréquence supérieure à 30% ?
- 5) Etablir les règles d'association de confiance supérieure à 60%.
- 6) Calculer leur *lift*.

## Exercice 2

Il s'agit maintenant d'extraire des règles d'un vrai jeu de données à l'aide du logiciel R. Pour cela, nous allons utiliser le package `arules`.

Installez et chargez le package `arules`

« The Mushroom data set includes descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota Family. It contains information about 8124 mushrooms (transactions). 4208 (51.8%) are edible and 3916 (48.2%) are poisonous. The data contains 22 nominal features plus the class attribute (edible or not). These features were translated into 114 items »

`data("Mushroom")`

Mushroom est déjà formaté dans la classe `large transaction` spécifique aux règles d'association.

Résumé du jeu de données

## ING1 - GENIE INFORMATIQUE : DATA EXPLORATION : REGLES D'ASSOCIATION

`summary(Mushroom)` : donne un résumé des données, notamment le nombre d'items, le nombre de transactions et le nombre d'items par transaction.

`inspect(Mushroom[1])` : affiche les transactions (ici la 1<sup>ère</sup>). Les transactions ici sont les champignons.

Les items sont les caractéristiques des champignons. Pour connaître les items, il y a les fonctions

`itemInfo(Mushroom)` : affiche les noms des items et les variables associées

`itemFrequency(Mushroom)` : calcule les fréquences apparition des items dans les transactions

`itemFrequencyPlot(Mushroom)` : affiche le diagramme bâtons des fréquences

Paramètres :

`support` = a numeric value. Only display items which have a support of at least `support`.

`topN` = a integer value. Only plot the `topN` items with the highest item frequency.

Question 1 : Combien y a-t-il d'items? De transactions ?

Afficher le graphique des items de support supérieur à 0.2

Afficher les 20 items les plus fréquents

### Calcul des règles

La fonction `apriori` calcule les règles d'association avec l'algorithme Apriori pour les règles d'élagage données en paramètres

```
rules=apriori(Mushroom,parameter=list(support=0.5,confidence=0.7))
```

construit les règles dont le support est supérieur à 50% et la confiance à 70%.

```
inspect(rules[1:100])
```

affiche les règles construites (ici les 100<sup>ères</sup>) avec le support, la confiance et le lift.

On pourra afficher l'aide de la fonction `apriori` pour avoir la liste des paramètres avec notamment `maxlen` le nombre maximum d'items par règles ou `target` pour avoir un item spécifique en conclusion.

Question 2 : Combien y-a-t-il de règles construites ? Combien y-a-t-il d'items par règles ?

Faire varier les seuils du support et la confiance et regarder l'évolution du nombre de règles et d'items par règles.

### Trier les règles

La fonction `sort` permet de trier les règles suivant le support, la confiance ou le lift.

```
inspect(sort(rules,by="confidence"))
```

La fonction `subset` permet de créer un sous-ensemble de règles suivants un critère ou des critères

```
rules2=subset(rules,subset=lhs %pin% "close" & lift>1)
inspect(rules2)
```

Ici, on affiche les règles donc l'antécédent contient l'item « close » et donc le lift est supérieur à 1.

Question 3 : Afficher les règles par lift décroissant puis par confiance décroissante.

Afficher les règles avec l'item « poisonous » en conclusion.

Un nouveau champignon est répertorié avec les caractéristiques suivantes.

RingNumber=one

veilColor=white

Déterminer les règles le concernant. Peut-on conclure qu'il est vénéneux ou comestible ?

### Remarque

Si le temps de calcul est trop long, on peut travailler sur un sous-échantillon. Pour le construire, on utilise la fonction sample.

```
sample(Mushroom, size=50, replace=F)
```

### Exercice 3

Il s'agit dans cet exercice de transformer un tableau de données de la classe data.frame pour le mettre dans la classe large transaction afin de pouvoir créer des règles d'association.

On utilise pour cela l'instruction

```
as(nom.du.data.frame, "transactions")
```

Attention, il faut bien entendu qu'il n'y ait que des variables qualitatives dans le data.frame

Question 4 : Mettre le jeu de données « Titanic » sous la forme transaction.

Combien y-a-t-il d'items ? de transactions ?

Construire toutes les règles de support supérieur à 10% et de confiance supérieure à 50%

Quelles sont les règles intéressantes ayant comme item de sortie « survivant » ? Pourquoi n'y-a-t-il aucune règle avec les enfants ?