

Travaux Pratiques n° 1 : Association

Objectifs : Manipuler les règles d'association

1 Règles d'association pas à pas

Le but ici est d'illustrer un exemple pour l'inférence de règles d'association en utilisant le paquet `arules` qu'il vous faudra télécharger. Nous allons suivre l'exemple donné dans www.rdatamining.com.

Executer et commenter les différentes étapes du code suivant.

```
#load and prepare the data set Titanic

str(Titanic)
df <- as.data.frame(Titanic)
head(df)
titanic.raw <- NULL
for(i in 1:4) {
  titanic.raw <- cbind(titanic.raw, rep(as.character(df[,i]), df$Freq))
}
titanic.raw <- as.data.frame(titanic.raw)
names(titanic.raw) <- names(df)[1:4]
dim(titanic.raw)
str(titanic.raw)
head(titanic.raw)
summary(titanic.raw)

# have a look at the 1st 5 lines
readLines("./data/Dataset.data", n=5)
# read it into R
titanic <- read.table("./data/Dataset.data", header=F)
names(titanic) <- c("Class", "Sex", "Age", "Survived")

#load the library rules
library(arules)

# find association rules with default settings
```

```
rules.all <- apriori(titanic.raw)

# inspection
rules.all
inspect(rules.all)

# rules with rhs containing "Survived" only
rules <- apriori(titanic.raw, control = list(verbose=F),
                parameter = list(minlen=2, supp=0.005, conf=0.8),
                appearance = list(rhs=c("Survived=No", "Survived=Yes"),
                                   default="lhs"))
quality(rules) <- round(quality(rules), digits=3)
rules.sorted <- sort(rules, by="lift")

inspect(rules.sorted)

# find redundant rules
subset.matrix <- is.subset(rules.sorted, rules.sorted)
subset.matrix[lower.tri(subset.matrix, diag=T)] <- NA
redundant <- colSums(subset.matrix, na.rm=T) >= 1
which(redundant)

# remove redundant rules
rules.pruned <- rules.sorted[!redundant]
inspect(rules.pruned)

#find rules with another setting
rules <- apriori(titanic.raw,
                parameter = list(minlen=3, supp=0.002, conf=0.2),
                appearance = list(rhs=c("Survived=Yes"),
                                   lhs=c("Class=1st", "Class=2nd", "Class=3rd",
                                           "Age=Child", "Age=Adult"),
                                   default="none"),
                control = list(verbose=F))
rules.sorted <- sort(rules, by="confidence")
inspect(rules.sorted)

#Visualization

library(arulesViz)

plot(rules.all)

plot(rules.all, method="grouped")

plot(rules.all, method="graph")
```

```
plot(rules.all, method="graph", control=list(type="items"))
```

```
plot(rules.all, method="paracoord", control=list(reorder=TRUE))
```

2 Marketing et promotions

On considère une épicerie qui a comme bilan d'achat le tableau ci-dessous.

| Achats | Produit 1 | Produit 2 | Produit 3 | Produit 4 | Produit 5 |
|----------|-----------|-----------|-----------|-----------|-----------|
| Client 1 | x | | | x | x |
| Client 2 | x | x | | | x |
| Client 3 | | | | | x |
| Client 4 | | | x | x | x |
| Client 5 | x | x | x | x | x |
| Client 6 | x | | | | x |
| Client 7 | x | | | x | x |
| Client 8 | | x | x | | |

1. Générez un fichier arff contenant les données de ce bilan d'achat.
2. Extraire les règles d'associations avec un support de 0.5 puis de 0.1
3. Que pouvez-vous conseiller comme promotions au commerçant ?