INFO411
Laboratory exercises on Association Rule Mining

Overview:
This lab is a non-assessed group exercise. Association rule mining can help uncover relationships between seemingly unrelated data in a transactional database. In data mining, association rules are useful in discovering consequences of commonly observed patterns within a set of transactions.

What you need:
1. R software package (already installed on the lab computers)
2. The file "Lab5_AssociationRules.zip" on Moodle.

Preparation:
1. Work in a group of size three to four (minimum size of a group is three. But no more than four students are to work together).
2. Boot computer into Windows mode.
3. Download laboratory_week6.zip then save to an arbitrary folder, say "C:\Users\yourname\Desktop"
4. Uncompress laboratory_week6.zip into this folder
5. Start "R"
6. Change the working directory by entering: `setwd("C:/Users/yourname/Desktop")`
   (Note that R expects forward slashes rather than backwars slashes as used by Windows.)

Your task:
Your are to submit a PDF document which contains your answers of the questions in this laboratory exercise. One document is to be submitted by each group. The header of the document must list the name and student number of all students in the group. Clearly indicate which question you have answered.
The following link provides a documentation of the association rule module in R (called arules). The link can help you develop a better understanding of the usage and parameters of the association rule package in R: http://cran.r-project.org/web/packages/arules/arules.pdf

Work through the following step and answer given questions:
Step1: Familiarize yourself with the arules package in R.
Start R and type:

```
 library(arules)
```

to load the package. We shall start from the analysis of a small file `sample1.csv` that contains some transactional data. To load data into R enter:

```
sample1.transactions <- read.transactions("sample1.csv", sep=",")
```

To get information about the total number of transactions in a file sample1.csv enter:

```
sample1.transactions
```

To get a summary of data set sample1.csv enter:

```
summary(sample1.transactions)
```

The data set is described as sparse matrix that consists of 10 rows and five columns. The density of

the matrix is 0.48.
Next, list of the most frequent items and the distribution of items per transactions. In our case two transactions consist of one item, five transactions consist of two items, two transactions consist of three items and one transaction consists of four items. To discover the association rules enter:

```
sample1.apriori <- apriori(sample1.transactions)
```

The results start from a summary of parameters used for the execution of the Apriori algorithm. Note that a default value for confidence and support is being used. The minimum (minlen) and maximum (maxlen) number of items in an items follows. The default target is rules. It could also be itemsets. The other targets can be set in the call to apriori with the parameter argument which takes a list of keyword arguments. To list the association rules found by Apriori algorithm enter:

```
inspect(sample1.apriori)
```

It is possible to change the values of the parameters support and confidence to get more association rules:

```
sample1.apriori <- apriori(sample1.transactions,parameter=list(supp=0.5,
conf=1.0))
```

R's implementation of Apriori algorithm is also capable of processing data stored in a file with transaction ID and a single item per line. For example the file sample2.csv contains the same data as the file sample1.csv. To load such data into R enter:

```
sample2.transactions <-
read.transactions("sample2.csv",sep=",",format="single",cols=c(1,2))
```

To discover and to list the association rules found by Apriori algorithm enter:

```
sample2.apriori <- apriori(sample2.transactions,parameter=list(supp=0.5,
conf=1.0))

inspect(sample2.apriori)
```

Use the data set sample4.csv. To load data into R enter:

```
sample4.transactions <- read.transactions("sample4.csv", sep=",")
```

R should reply with a message: cannot coerce list with transactions with duplicated items. The message means that at least one of the transactions in an input data set has duplicated items. Indeed, a transaction beer, milk, bread, sausage, beer has a duplicated item beer. To eliminate duplicated items load data into R in the following way:

```
sample4.transactions <- read.transactions("sample4.csv", sep=",",
rm.duplicates=TRUE)
```

Task1: Visualize the mined association rules from sample1.csv.
Compute the association rules of the data in file sample1.csv by setting the support threshold to 0.12 and confidence threshold to 1.0. Show the association rules.
Load the association rule visualization module in R:

```
library(arulesViz)
```

Then plot the association rules by using

```
plot(sample1.apriori,method="graph")
```

Explain what can be seen in this graph.

Task2:
Mine association rules from the data set `words.csv`. List information of the transactions included in the data set and list a summary of the data set. Discover the association rules for the default values

of the parameters support and condence. Find the largest values of the parameters support and confidence such that discovery of association rules provides a nonempty set of nontrivial rules (a rule is trivial if its left or right hand side is empty). List the association rule(s) found.

Task3:
Load the survival data of passangers on the Titanic:

```
load("titanic.raw.rdata")
```

Explain the output shown by the command
str(titanic.raw)
Mine association rules that only have "survived=No" or "Survived=Yes" on the RHS:

```
rules <-
apriori(titanic.raw,parameter=list(minlen=2,supp=0.005,conf=0.8),appearance=list
(rhs=c("Survived=No", "Survived=Yes"),default="lhs"))
```

Sort by "lift" then show results

```
rules.sorted<-sort(rules,by="lift")

inspect(rules.sorted)
```

Explain the output shown by the call of "inspect".

Create the following two plots then explain what is shown

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

Write up all your answers, then submit your answer as a PDF document (one submission - one member submits - per group) via the link provided on MOODLE. One submission per group!