Association Model for Market Basket Analysis

> # Load the library

> library(arules)

Loading required package: Matrix

Attaching package: ‘arules’

The following objects are masked from ‘package:base’:

abbreviate, write

> # Load your data

> mydata = read.csv("D:\\ishita\\college\_py\\MBA.csv", header = T)

> head(mydata)

ID Products

1 1 Product A

2 1 Product B

3 1 Product C

4 1 Product I

5 2 Product E

6 2 Product F

> # Split the data into transactions

> df <- split(mydata$Products, mydata$ID)

> transactions = as(df, "transactions")

> # Print summary of transactions

> summary(transactions)

transactions as itemMatrix in sparse format with

15 rows (elements/itemsets/transactions) and

15 columns (items) and a density of 0.3644444

most frequent items:

Product I Product A Product C Product F Product H (Other)

11 7 7 7 7 43

element (itemset/transaction) length distribution:

sizes

3 4 5 6 7 9

1 2 6 3 2 1

Min. 1st Qu. Median Mean 3rd Qu. Max.

3.000 5.000 5.000 5.467 6.000 9.000

includes extended item information - examples:

labels

1 Product A

2 Product B

3 Product C

includes extended transaction information - examples:

transactionID

1 1

2 2

3 3

> # Calculate the relative frequency of each item in the transactions

> itemFrequency(transactions, type = "relative")

Product A Product B Product C Product D Product E Product F Product G Product H

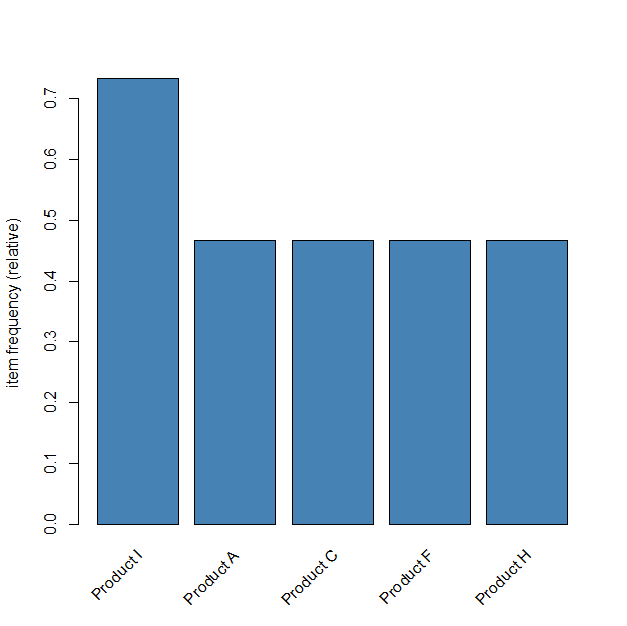
0.46666667 0.40000000 0.46666667 0.20000000 0.40000000 0.46666667 0.33333333 0.46666667

Product I Product J Product K Product L Product M Product N Product O

0.73333333 0.40000000 0.20000000 0.33333333 0.26666667 0.06666667 0.26666667

> # Plot item frequency

> itemFrequencyPlot(transactions, topN = 5, type = "relative", col = "steelblue")



> # Apply the Apriori algorithm - association rules

> rules <- apriori(transactions, parameter = list(supp = 0.005, conf = 0.8))

Apriori

Parameter specification:

confidence minval smax arem aval originalSupport maxtime support minlen maxlen target ext

0.8 0.1 1 none FALSE TRUE 5 0.005 1 10 rules TRUE

Algorithmic control:

filter tree heap memopt load sort verbose

0.1 TRUE TRUE FALSE TRUE 2 TRUE

Absolute minimum support count: 0

set item appearances ...[0 item(s)] done [0.00s].

set transactions ...[15 item(s), 15 transaction(s)] done [0.00s].

sorting and recoding items ... [15 item(s)] done [0.00s].

creating transaction tree ... done [0.00s].

checking subsets of size 1 2 3 4 5 6 7 8 9 done [0.00s].

writing ... [2328 rule(s)] done [0.00s].

creating S4 object ... done [0.00s].

> # View top 5 rules

> inspect(head(sort(rules, by = "lift"), 5))

lhs rhs support confidence coverage lift count

[1] {Product N} => {Product D} 0.06666667 1 0.06666667 5 1

[2] {Product G, Product N} => {Product D} 0.06666667 1 0.06666667 5 1

[3] {Product E, Product N} => {Product D} 0.06666667 1 0.06666667 5 1

[4] {Product H, Product N} => {Product D} 0.06666667 1 0.06666667 5 1

[5] {Product F, Product N} => {Product D} 0.06666667 1 0.06666667 5 1