Descubrimiento del Conocimiento usando herramientas de Big Data Módulo 2

Marco Andrés Vázquez Hernández

Práctica Patrones Comúnes.

Septiembre de 2018

Instituto Politécnico Nacional

Descripción

Utilizar los datos que se proporcionan para encontrar:

1.- Representación de las transacciones de cada mes (Transacción, [Productos])

2.- Representación binaria de los datos de transacciones

3.- Encontrar los patrones frecuentes utilizando un soporte de 0.001

Planteamiento

Se plantea la pregunta a contestar: ¿Que secciones de la tienda deberían de estar juntas (físicamente) para pomover las ventas?

Se tomó la variable product_subcategory como indicadora de las secciones de la tienda. como dicha variable está contenida en los datos de las ventas mensuales, no se usaron los datos del archivo de productos

Carga de archivos

```
library("arules")
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'arules'
```

```
## The following objects are masked from 'package:base':
##
##     abbreviate, write
```

```
library("ggplot2")
#install.packages("magrittr")
library("magrittr")
setwd("C:/Users/marco/IPN_BigData/Modulo2/Practica_patrones_frecuentes")

ventas01<-read.table("sales_01_Jan",nrows = -1,sep="|",quote="\"",header=T)
ventas02<-read.table("sales_02_Feb",nrows = -1,sep="|",quote="\"",header=T)
ventas03<-read.table("sales_03_Mar",nrows = -1,sep="|",quote="\"",header=T)
ventas04<-read.table("sales_04_Apr",nrows = -1,sep="|",quote="\"",header=T)
ventas05<-read.table("sales_05_May",nrows = -1,sep="|",quote="\"",header=T)
ventas06<-read.table("sales_06_Jun",nrows = -1,sep="|",quote="\"",header=T)
ventas07<-read.table("sales_07_Jul",nrows = -1,sep="|",quote="\"",header=T)
ventas08<-read.table("sales_08_Aug",nrows = -1,sep="|",quote="\"",header=T)
ventas09<-read.table("sales_09_Sep",nrows = -1,sep="|",quote="\"",header=T)
ventas10<-read.table("sales_10_Oct",nrows = -1,sep="|",quote="\"",header=T)
ventas11<-read.table("sales_11_Nov",nrows = -1,sep="|",quote="\"",header=T)
ventas12<-read.table("sales_12_Dec",nrows = -1,sep="|",quote="\"",header=T)
ventas<-rbind(ventas01,ventas02,ventas03,ventas04,ventas05,ventas06,ventas07,
              ventas08,ventas09,ventas10,ventas11,
              ventas12)
```

Transformación de datos

Para usar la librería de R llamada "arules" que hace uso del algoritmo apriori para detección de patrones frecuentes se deben de transformar los datos a una estructura llave - valor y después guardarlos en un csv y leerlos con la función read.transactions

```
aux<-ventas[,c("time_id","customer_id","product_subcategory")]
aux$llave<-paste0(aux$time_id,"-",aux$customer_id)
aux$time_id<-NULL
aux$customer_id<-NULL
aux<-aux[!duplicated(aux),]
write.table(aux[,c(2,1)],"transacciones.csv",quote=F,row.names=F,sep=",")

transacciones<-read.transactions("transacciones.csv",format="single",sep=",",cols =
                                   c("llave","product_subcategory"))
```

Análisis y resultados

Se usaron dichos datos para evaluar el algoritmo apriori y obtener los patrones más frecuentes. Se utilizó un soporte de 0.02 ya que se tienen suficientes datos de transacciones y "pocas" categorías.

```
itemsets <- apriori(data = transacciones,
                    parameter = list(support = 0.02,
                                     minlen = 2,
                                     maxlen = 20,
                                     target = "frequent itemset"))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##          NA    0.1    1 none FALSE            TRUE       5    0.02      2
##  maxlen            target   ext
##      20 frequent itemsets FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 410
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[102 item(s), 20522 transaction(s)] done [0.01s].
## sorting and recoding items ... [67 item(s)] done [0.00s].
## creating transaction tree ... done [0.01s].
## checking subsets of size 1 2 3 done [0.00s].
## writing ... [10 set(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
order_itemsets <- sort(itemsets, by = "support", decreasing = TRUE)
inspect(order_itemsets)
```

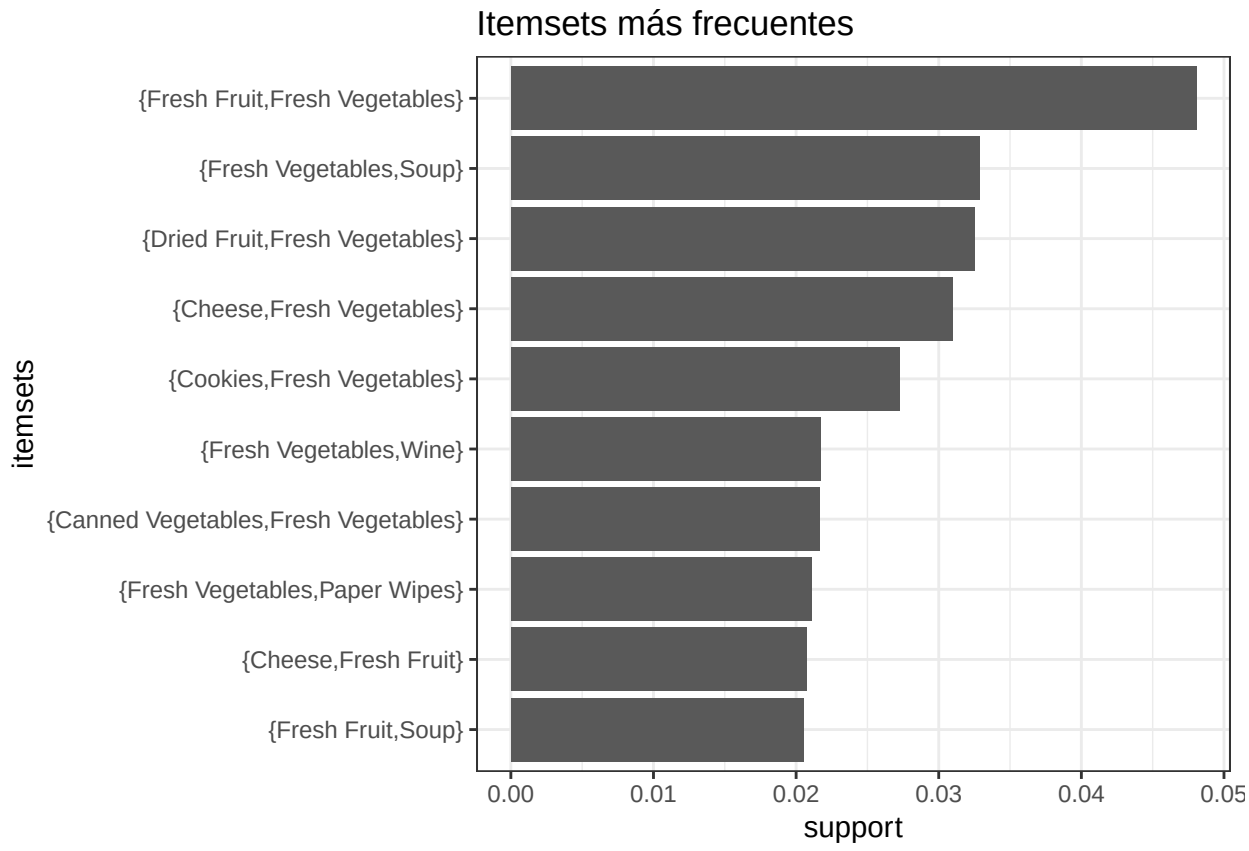
```
##      items                                 support    count
## [1]  {Fresh Fruit,Fresh Vegetables}        0.04804600 986
```

```
## [2]  {Fresh Vegetables,Soup}                0.03284280 674
## [3]  {Dried Fruit,Fresh Vegetables}         0.03250171 667
## [4]  {Cheese,Fresh Vegetables}              0.03094240 635
## [5]  {Cookies,Fresh Vegetables}             0.02723906 559
## [6]  {Fresh Vegetables,Wine}                0.02173277 446
## [7]  {Canned Vegetables,Fresh Vegetables}   0.02163532 444
## [8]  {Fresh Vegetables,Paper Wipes}         0.02109931 433
## [9]  {Cheese,Fresh Fruit}                   0.02070948 425
## [10] {Fresh Fruit,Soup}                     0.02051457 421
```

Visualización

Se presenta la visualización de dichos patrones por medio de una gráfica de barras:

```
as(order_itemsets, Class = "data.frame") %>%
  ggplot(aes(x = reorder(items, support), y = support)) +
  geom_col() +
  coord_flip() +
  labs(title = "Itemsets más frecuentes", x = "itemsets") +
  theme_bw()
```



Reglas

Para las reglas los resultados quedan:

```
rules <- apriori(data = transacciones,
                    parameter = list(support = 0.02,
                                     minlen = 2,
                                     maxlen = 20,
                                     confidence = .2,
                                     target = "rules"))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.2    0.1    1 none FALSE            TRUE       5    0.02      2
##  maxlen target   ext
##      20  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 410
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[102 item(s), 20522 transaction(s)] done [0.01s].
## sorting and recoding items ... [67 item(s)] done [0.00s].
## creating transaction tree ... done [0.01s].
## checking subsets of size 1 2 3 done [0.00s].
## writing ... [8 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
order_rules <- sort(rules, by = "support", decreasing = TRUE)
inspect(order_rules)
```

```
##     lhs                    rhs                support    confidence
## [1] {Fresh Fruit}       => {Fresh Vegetables} 0.04804600 0.2831706
## [2] {Soup}              => {Fresh Vegetables} 0.03284280 0.2756646
## [3] {Dried Fruit}       => {Fresh Vegetables} 0.03250171 0.2908853
## [4] {Cheese}            => {Fresh Vegetables} 0.03094240 0.2639235
## [5] {Cookies}           => {Fresh Vegetables} 0.02723906 0.2610929
## [6] {Wine}              => {Fresh Vegetables} 0.02173277 0.2756489
## [7] {Canned Vegetables} => {Fresh Vegetables} 0.02163532 0.2759478
## [8] {Paper Wipes}       => {Fresh Vegetables} 0.02109931 0.2684439
##     lift      count
## [1] 1.0145298 986
## [2] 0.9876378 674
## [3] 1.0421697 667
## [4] 0.9455724 635
## [5] 0.9354311 559
## [6] 0.9875817 446
## [7] 0.9886523 444
## [8] 0.9617677 433
```

Cabe destacar que la confianza es muy baja (20%) y que en el presente caso sólo podríamos establecer qué

secciones deberían de estar cerca de la sección de vegetales frescos.

Visualización

Se presenta la visualización de dichas reglas por medio de una gráfica de barras:

```
as(order_rules, Class = "data.frame") %>%
  ggplot(aes(x = reorder(rules, confidence), y = confidence)) +
  geom_col() +
  coord_flip() +
  labs(title = "Reglas derivadas", x = "reglas") +
  theme_bw()
```