

- 7) Los pasos 1 al 6 se repiten tantas veces como consecuentes se quieran definir dentro del conjunto de ítems etiquetados y obtenidos de cada serie temporal.

A continuación se anexa la codificación de este algoritmo para un caso particular de un conjunto de series temporales financieras. En este ejemplo, nuestro consecuente esta etiquetado como el ítem TEF+ que implica tendencias alcistas de la acción Telefónica. El ancho de ventana para la extracción de reglas temporales es de cada 15 cotizaciones (unas tres semanas hábiles normales).

#Llamado de librerías necesarias para el algoritmo en una sesión de trabajo de R

```
library(utils)
library(Matrix)
library(stats)
library(lattice)
library(arules)
```

Variables comienzo

Fijamos el consecuente y el ancho de ventana (PASO 1 del Algoritmo)

```
CONSEC <- "TEF+"
WinW <- 15
```

Ordenamos los tramos de todas las series temporales por posición en el tiempo de manera ascendente (PASO 1 del Algoritmo)

```
MATITEMS <- SUBPATMAT[order(SUBPATMAT[,1]),]
```

Ubicamos la posición del consecuente "TEF+" dentro de la base de datos que contiene los ítems obtenidos desde CONOTOOL y que han sido previamente ordenados según el tiempo – punto de ocurrencia (PASO 1 del Algoritmo)

```
POS <- MATITEMS[MATITEMS$namePatt==CONSEC,]$PosP
```

Buscamos los ítems que caen antes del ítem consecuente según el ancho de la ventana o retraso temporal creando la base de datos transaccional-temporal T (PASO 1 del Algoritmo)

```
PATT <- NULL
CONT <- 0
for (h in 1:length(POS))
{
  INI <- POS[h]-WinW
  FIN <- POS[h]
  if (INI>0)
  {
    CAEN <- (MATITEMS$PosP >=INI & MATITEMS$PosP<FIN)
    if (sum(as.numeric(CAEN)>0))
    {
      print(as.character(t(MATITEMS[CAEN,]$namePatt)))
    }
  }
}
```

```

PATT <- c(PATT,list(as.character(t(MATITEMS[CAEN,]$namePatt))))
CONT <- CONT+1
    }
  }
}

PATTAUX <- PATT

```

Eliminamos el consecuente de la Base de Datos Transaccional-temporal (PASO 2 del Algoritmo)

```

PATTAUX2<-PATTAUX
for (i in 1:length(PATTAUX2))
{
  DONDE <- (PATTAUX2[[i]]==CONSEC)
  PATTAUX2[[i]] <- PATTAUX2[[i]][!DONDE];
}

```

Creamos la base de datos transaccional para ECLAT (PASOS 3 y 4 del Algoritmo)

```

BASE <- as(PATTAUX2,"transactions")
ITEMS_FREC <- eclat(BASE,parameter=list(support=0.15),control=list(verbose=FALSE))
inspect(SORT(ITEMS_FREC),by="support")
ITEMSFINAL <- labels(SORT(ITEMS_FREC,by="support"))

```

Para cada itemset frecuente obtenido desde ECLAT se procede a contabilizar el número de veces que aparece cada itemset frecuente en el tiempo y dentro de una ventana deslizante de ancho predefinido recorriendo la base de datos original que contiene todos los tramos extraídos de las series temporales. De esta forma, se puede determinar las métricas de las reglas del tipo X implica Y según el retraso y el rango de tiempo determinado por la ventana deslizante (PASO 5 del Algoritmo)

#Sacamos la base de datos transaccional con un paso por ítem frecuente y ancho de ventana igual a WinW

```

FINALPATTERNS <- SUBPATMAT[,c(1,2,5)]

```

Extraemos las transacciones con ventana de ancho WINDOWSWIDTH y paso de ítem en ítem

```

MAT <- FINALPATTERNS[order(FINALPATTERNS[, 1]), ]
Ini=1
End=max(MAT[, 1])
MATTRANS <- NULL
NROW <- nrow(MAT)
POS3 <- MAT[, 1]
for (h in 1:(nrow(MAT)-1))
{
  CAEN3 <- ((POS3 >= MAT[h, 1]) & (POS3 < (MAT[h,1] + WinW)))
  MATTRANS <- c(MATTRANS, list(c(as.character(MAT[h, 1]),
as.character(t(MAT[CAEN3,3])))))
}

```

Eliminamos los repetidos y los vacíos

```

MATTRANS2 <- NULL
for (i in 1:length(MATTRANS))
{

```

```

      MATTRANS[[i]] <- unique(MATTRANS[[i]])
      if (length(MATTRANS[[i]])>1) MATTRANS2 <- c(MATTRANS2,
list(c(as.character(MATTRANS[[i]]))))
    }
# Buscamos los ítems frecuentes en la base de datos transaccional #
NUMITEMS <- NULL
for (i in 1:length(ITEMSFINAL))
{
  # Extraemos separadamente los ítems frecuentes
  AUX <- sub("[{}","",ITEMSFINAL[i])
  AUX2 <- sub("[{}","",AUX)
  AUX3 <- strsplit(AUX2,"")
  # Para cada ítem frecuente buscamos donde está en la base de datos transaccional
  PosAux <- NULL
  PosenMat <- NULL
  for (j in 1:length(MATTRANS2))
  {
    if (sum(!(AUX3[[1]] %in% MATTRANS2[[j]]))==0)
    {
      # Posición de la ventana donde hay coincidencia de ítems
      PosAux <- c(PosAux,as.numeric(MATTRANS2[[j]][1]))
      # Fila (num de transacción) en MATTRANS2 donde hay coincidencia
      PosenMat <- c(PosenMat,j)
    }
  }
  PosAux2 <- NULL
  if (length(PosAux)>0)
  {
    PosAux2 <- PosAux[1]
    Indice <- 1
    for (j in 2:length(PosAux))
    {
      if (PosAux[j]>=(PosAux2[Indice]+WinW))
      {
        PosAux2 <- c(PosAux2,PosAux[j]);
        Indice <- Indice+1
      }
    }
    NUMITEMS <- c(NUMITEMS, length(PosAux2))
  }
ITEMSFRECTOT <-
data.frame(as(quality(SORT(ITEMS_FREC,by="support"))*length(PATTAUX2),"data.frame"),ITEM
SFINAL, TOTALSUPPORT=NUMITEMS)
ITEMSFRECTOT2 <- data.frame(ITEMSFINAL=ITEMSFRECTOT[,2],
  SUPPCONSEC=rep(length(PATTAUX2),length(ITEMSFINAL)),
  SUPPORTRULE=ITEMSFRECTOT[,1],
  SUPRULEPERC=as(quality(SORT(ITEMS_FREC,by="support")),"data.frame"),
  TOTALSUPPORT=ITEMSFRECTOT[,3],
  CONFIDENCE=ITEMSFRECTOT[,1]/ITEMSFRECTOT[,3])

# Exportación y presentación de reglas temporales (PASOS 6 y 7 del Algoritmo)

write.csv2(ITEMSFRECTOT2,paste(CONSEC,".CSV",sep=""))

```