Extracción de reglas

Laura del Pino Díaz

23/1/2017

Índice

Introducción

La extracción de reglas de asociación es una técnica de la minería de datos que nos permite extraer conocimiento de las bases de datos. En este trabajo trata de extraer conocimiento de la base de datos *lymphography*.

La base de datos *lymphography* contiene los atributos más descriptivos de las linfografías médicas, a saber:

- Lymphatics, nominal cuyos valores son {normal,arched,deformed,displaced}
- Block_of_affere de tipo nominal que toma los valores {no,yes}
- Bl_of_lymph_c de tipo nominal que toma los valores {no,yes}
- Bl_of_lymph_s de tipo nominal que toma los valores {no,yes}
- By_pas de tipo nominal que toma los valores {no,yes}
- Extravasates de tipo nominal que toma los valores {no,yes}
- Regeneration_of de tipo nominal que toma los valores {no,yes}
- Early_uptake_in de tipo nominal que toma los valores {no,yes}
- Lym_nodes_dimi de tipo numérico discreto que toma los valores [0,3]
- Lym_nodes_enlar de tipo numérico que toma los valores [1,4]
- Change_in_lym de tipo nominal que toma los valores {bean,oval,round}
- Defect_in_node de tipo nominal que toma los valores {no,lacunar,lac_margin,lac_central}
- Changes_in_node de tipo nominal que toma los valores {no,lacunar,lac_margin,lac_central}
- Changes_in_stru de tipo nominal que toma los valores {no,grainy,drop_lik,coarse,diluted,reticular,stripped,faint}
- Special_form de tipo nominal que toma los siguientes valores {no,chalices,vesicles}
- Dislocation_of de tipo nominal que toma los valores {no,yes}
- Exclusion_of_no de tipo nominal que toma los valores {no,yes}
- No_of_nodes_in de tipo numérico discreto que toma los valores [1,8]
- Class de tipo nominal que toma los valores {normal,metastases,malig_lymph,fibrosis}

La base de datos la he tomado de la página web de KEEL por lo que habrá que tener en cuenta el apartado descriptivo que aparece en la parte superior del fichero.

```
datos <- read.csv("lymphography.dat", comment.char = "@")
colnames(datos) <- c("Lymphatics", "Block_of_affere", "Bl_of_lymph_c","Bl_of_lymph_s","By_pas", "Extrava
```

Preparación de los datos

Para realizar la extracción de reglas tenemos que preparar los datos para que estén en un formato que nos facilite el trabajo. Este formato es un formato binario donde los valores sean 0 y 1 que nos permita extraer el los casos en los que aparece el atributo y en los que no, para metricas como la *confianza confirmada* es necesario conocer los casos en donde se da un valor y su contrario.

Esta transformación la conseguimos con el siguiente conjunto de funciones:

```
transformDataToBinary <- function(lvl,data){
  result = sapply(data,function(x){ifelse(x==lvl,1,0)})
  return(result)
}

selectData <-function(column){
   if(is.factor(column) && length(levels(column))>2){
     lvls = levels(column)
     m = matrix(nrow = 147, ncol = 0)
     m = data.frame(m)
     for(lvl in lvls){
      d = transformDataToBinary(lvl,column)
      d = as.factor(d)
      m = cbind(m,d)
     }
     colnames(m) <- lvls
     return(m)
   }
  else if(!is.factor(column)){
    return(selectData(as.factor(column)))
  }
  else{
    return(column)
  }
}

expandDataFrame <- function(dataFrame){
  extendedData = selectData(datos[,1])
  for(i in 2:(dim(dataFrame)[2])){
    extendedData = cbind(extendedData,selectData(datos[,i]))
  }
  return(extendedData)
}

expandedDataFrame = expandDataFrame(datos)
```

En estos momentos en *expandedDataFrame* tenemos 54 variables creadas a partir de los valores que toman las variables, esto es, si la variable es categorica y tenía tres posibles valores tenemos tres variables a partir de esta de forma que cada nueva variable toma valores de 0 o 1 donde 0 es si no tiene ese valor y 1 en caso de que si toma el valor.

Por suerte aquellas variables que no son categóricas son numéricas discretas con un conjunto reducido de valores de forma que se puede crear una variable por cada uno de los valores del dominio de la variable.

Por desgracia la salida de la función *expandDataFrame* no da nombres significativos a todas las variables por lo que es conveniente cambiarles el valor a aquellas que no tienen nombre representativo o que el nombre se repite produciendo confusión.

Por restricciones sobre el número de páginas la sección de código que modifica los nombres de las columnas no se muestra. Se recomienda abrir el fichero .Rmd para visualizarlo.

Ahora que ya tenemos el dataframe en un estado que nos favorece nos queda que cambiar la interpretación que le damos, para que sea del tipo transacción. Para ello cargamos el paquete *arules* que nos permitirá hacer dicha transformación y además cargaremos el paquete de visualización de reglas de asociación *arulesViz* que será útil en pasos posteriores.

```
require(arules)
```

```
## Loading required package: arules
## Loading required package: Matrix
##
## Attaching package: 'arules'
## The following objects are masked from 'package:base':
##
##     abbreviate, write
```

```
require(arulesViz)
```

```
## Loading required package: arulesViz
## Loading required package: grid
## Warning: failed to assign NativeSymbolInfo for lhs since lhs is already
## defined in the 'lazyeval' namespace
## Warning: failed to assign NativeSymbolInfo for rhs since rhs is already
## defined in the 'lazyeval' namespace
```

```
transactionData <- as(expandedDataFrame, "transactions")
summary(transactionData)
```

```
## transactions as itemMatrix in sparse format with
##  147 rows (elements/itemsets/transactions) and
##  108 columns (items) and a density of 0.5
##
## most frequent items:
##                     normal=0        Changes_in_stru=no=0
##                          145                         145
## Changes_in_stru=reticular=0          No_of_nodes_in=8=0
##                          145                         145
##              Class=normal=0                     (Other)
##                          145                        7213
##
## element (itemset/transaction) length distribution:
## sizes
##  54
## 147
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      54      54      54      54      54      54
##
## includes extended item information - examples:
##       labels variables levels
## 1   arched=0    arched      0
## 2   arched=1    arched      1
```

```
## 3 deformed=0  deformed      0
##
## includes extended transaction information - examples:
##   transactionID
## 1             1
## 2             2
## 3             3
```

Del resumen anterior obtenermos que los elementos más repetidos son *normal=0*, *Changes in stru=no=0*, *Changes in stru=reticular=0*, *Changes in no nodes=8=0*, *Class=normal=0* lo que quiere decir que:

- En la mayoría de las transacciones hay un tipo de linfático distinto del normal.
- En la mayoría de las transacciones hay algún tipo de cambio en la estructura que no es de tipo reticular, convirtiendo este tipo en el más raro con solo dos casos.
- El cambio en el número de nodos es en la mayoría de los casos distinto de 8.
- Es raro el caso en el que el linfoma sea normal.

Además tenemos la información sobre la longitud máxima de la transacción que en este caso coincide con el número de variables.

Visualización de las transacciones

Podemos visualizar las transacciones para ver su comportamiento:

```
image(transactionData)
```