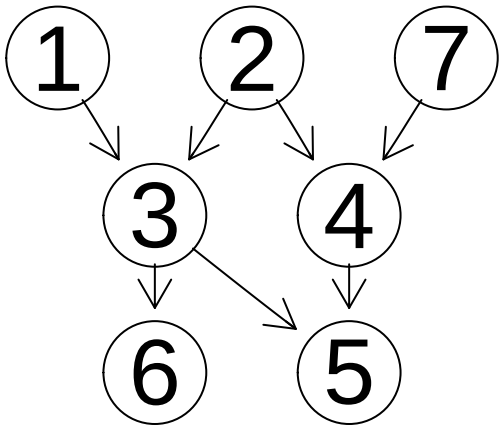
Ejercicios - Redes Bayesianas

Autor: Deyviss Jesús Oroya Villalta

```
library(bnlearn);
library(gRbase);
library(RBGL);
rm(list=ls())
graphics.off()
```

Trabajaremos sobre el sigueinte DAG

```
dag.study<-dag("1","2","7",c("3","1","2"),c("4","2","7"),c("5","3","4"),c("6","3"))
plot(dag.study)
```



Problema 3.1

Empleando las funciones que se han presentado en el tutorial, implementar una funcion que implemente el metodo de la d-separación para leer independencias condicionadas a partir del grafo.

Implementación de d-separation

```
d.separation = function (dag,x1,x2,c){
  # obtenemos la DAG ancestral
  dag.ancestral<- ancestralGraph(set=c(c,x1,x2),object=dag.study)
  # lo moralozamos
  moral.graph<-moralize(dag.ancestral)
  # vemos si el conjunto c, separa las variables x1 y x2
  boolean <- separates(x1,x2,c,moral.graph)
  return(boolean)}
```

Problema 3.2

1) Dado el DAG de la Figura 2: Comprobar si se verifican las independencias (1;4|5),(1;5|3, 4), (6;7|3), (1;7|5), (1;4|6) y (1;4|2, 3)

```r
# (1; 4|5)
d.separation(dag.study,"1","4","5")
```

```
## [1] FALSE
```

```r
# (1; 5|3, 4)
d.separation(dag.study,"1","5",c("3","4"))
```

```
## [1] TRUE
```

```r
#(6; 7|3)
d.separation(dag.study,"6","7",c("3"))
```

```
## [1] TRUE
```

```r
# (1; 7|5)
d.separation(dag.study,"1","7",c("5"))
```

```
## [1] FALSE
```

```r
# (1; 4|6)
d.separation(dag.study,"1","4",c("6"))
```

```
## [1] FALSE
```

```r
# (1; 4|2, 3)
d.separation(dag.study,"1","4",c("2","3"))
```

```
## [1] TRUE
```

2) Que conjunto(s) de variables C verifican (1; 7|C) y (1; 5|C) Definimos la función **set.search** que se encarga de buscar los conjunto dentro de un DAG(dag.study) que hace que dos variables del DAG sean independientes. La función devuelve una lista en la que cada índice índica el número de elementos que tiene los conjuntos que se encuentra en él.

```r
set.search<-function(dag.study,vars){
  # extraemos los nodos de dag.study
  list <- graph::nodes(dag.study)
  # quitamos los elementos que se encuentran en vars
  for (c in vars){list <- list[list != c]}
  result <- list()
  index = 0
  for (iter in 1:length(list)){
    # obtenemos todas las combinaciones de tamaño 'iter'
    conjunto <- combn(list,iter)
    # a cada conjunto obtenido le aplicamos la d.separacion para
    # verificar que el conjunto hace independiente condicionalmente
    # las variables dentro vars
    boolean.list <- apply(conjunto,2,function (x) d.separation(dag.study,vars[1],vars[2],x))
    index=index+1
    # recuperamos los conjuntos que tengan asociado un verdadero en 'boolean.list'
    result[[index]]=conjunto[,boolean.list==TRUE]
  }
  return(result)
}
```

los conjuntos que verifican (1;7|C)

```
# (1;7|C )
vars <- c("1","7")
set.1.7 <-set.search(dag.study,vars)
```

4 conjuntos de 1 elemento

```
set.1.7[[1]]
```

```
## [1] "2" "3" "4" "6"
```

4 conjuntos de 2 elementos

```
set.1.7[[2]]
```

```
##      [,1] [,2] [,3] [,4]
## [1,] "2"  "2"  "2"  "3"
## [2,] "3"  "4"  "6"  "6"
```

5 conjuntos de 3 elementos

```
set.1.7[[3]]
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,] "2"  "2"  "2"  "2"  "2"
## [2,] "3"  "3"  "3"  "4"  "4"
## [3,] "4"  "5"  "6"  "5"  "6"
```

4 conjuntos de 4 elementos

```
set.1.7[[4]]
```

```
##      [,1] [,2] [,3] [,4]
## [1,] "2"  "2"  "2"  "2"
## [2,] "3"  "3"  "3"  "4"
## [3,] "4"  "4"  "5"  "5"
## [4,] "5"  "6"  "6"  "6"
```

1 conjunto de 5 elementos

```
set.1.7[[5]]
```

```
## [1] "2" "3" "4" "5" "6"
```

los conjuntos que verifican (1;5|C)

```
# (1;5|C )
vars <- c("1","5")
set.1.5 <-set.search(dag.study,vars)
set.1.5
```

```
## [[1]]
## character(0)
##
## [[2]]
##      [,1] [,2]
## [1,] "2"  "3"
## [2,] "3"  "4"
##
## [[3]]
##      [,1] [,2] [,3] [,4] [,5]
## [1,] "2"  "2"  "2"  "7"  "3"
```

```
## [2,] "7"  "3"  "3"  "3"  "4"
## [3,] "3"  "4"  "6"  "4"  "6"
##
## [[4]]
##      [,1] [,2] [,3] [,4]
## [1,] "2"  "2"  "2"  "7"
## [2,] "7"  "7"  "3"  "3"
## [3,] "3"  "3"  "4"  "4"
## [4,] "4"  "6"  "6"  "6"
##
## [[5]]
## [1] "2" "7" "3" "4" "6"
```