Aprendizaje de redes bayesianas con R

Daniel Ranchal Parrado

26 de febrero de 2022

```
library(bnlearn)
set.seed(100)
```

Ejercicio 1

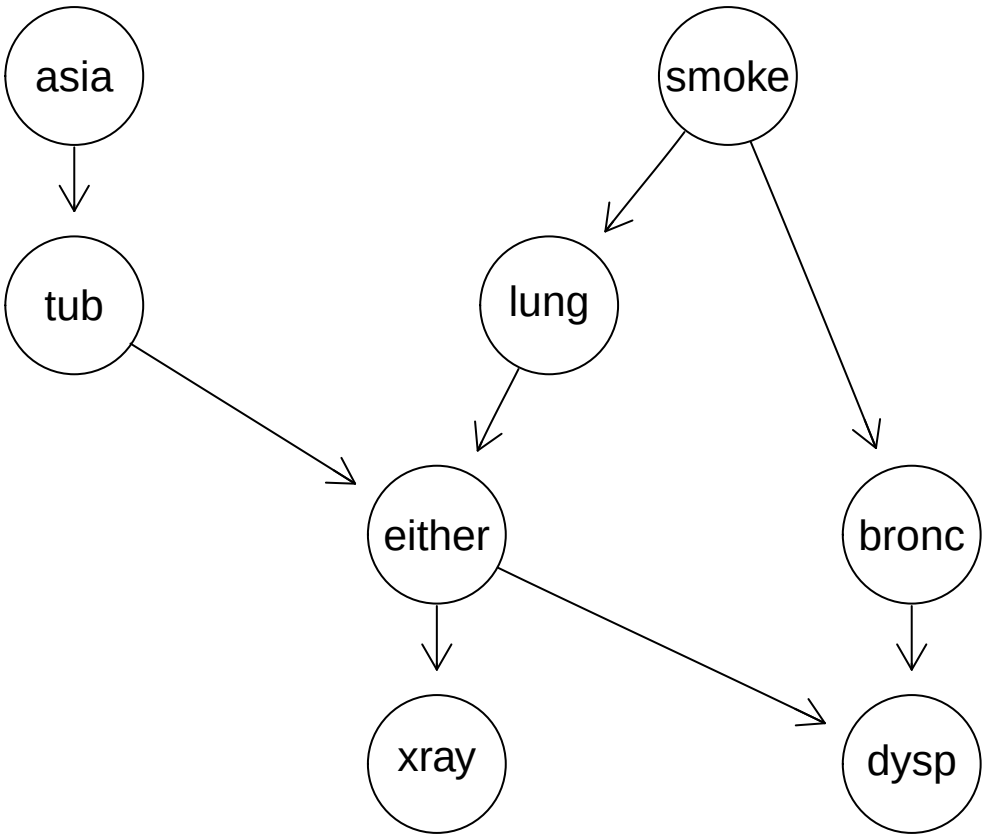
Como trabajo obligatorio de esta parte de la asignatura, se debe entrgar un script en R y un documento que haga lo siguiente

1. **Seleccione una red bayesiana con al menos 7 variables. No tiene que ser una red de nueva creación. Puede ser una usada en otras partes de la asignatura o del repositorio de bnlearn**.

Para este ejercicio se ha utilizado la red asia, que está disponible en el repositorio de bnlearn. Esta red está compuesta de 8 nodos. He descargado el fichero con el formato rds, que permite guardar objetos de R. En este caso el objeto que se ha guardado es de la clase **bn.fit**. En el ejercicio 4 este objecto dará problemas para hacer la comparación visual, por lo que se obtendrá el grafo acíclico no dirigido a partir de este.

```
asia_network <- readRDS("asia.rds")
graphviz.plot(asia_network)
```

```
## Loading required namespace: Rgraphviz
```



2. **Simular dos conjuntos de datos de distintos tamaños a partir de la red (por ejemplo uno con 200 casos y otro con 5000 casos).**

Para poder simular conjuntos de datos a partir de la red asia, se ha utilizado la función **rbn**. La función rbn implementa el muestreo lógico probabilístico y permite fijar el número de muestras que queremos generar. En este caso se han generado dos muestras, una con 200 instancias y otra con 5000 instancias.

```
small_dataset <- rbn(asia_network, n = 200)
big_dataset <- rbn(asia_network, n = 5000)
```

3. **Aprender la estructura con dos métodos distintos, uno basado en test de independencia y otro en scores, y con los dos conjuntos de datos**

En este apartado se ha aprendido la estructura utilizando el conjunto de datos pequeño y grande utilizando métodos basados en test de independencia y en score.

Como método basado en test de independencias se ha utilizado el método growth-shrink. Respecto al método basado en scores que se ha utilizado, se ha usado una búsqueda tabú con el score bde (Bayesian Dirichlet equivalent).

Se puede observar que en cada caso, después de aprender la estructura con cada método y con cada dataset, se ha aplicado la función **cextend**. Lo que hace esta función es convertir la estructura aprendida en una estructura consistente, lo que supone tener un grafo dirigido no acíclico.

```
gs.small_dataset.dag <- gs(small_dataset)
gs.small_dataset.dag <- cextend(gs.small_dataset.dag)

gs.big_dataset.dag <- gs(big_dataset)
gs.big_dataset.dag <- cextend(gs.big_dataset.dag)
```

```
tabu_search.bde.small_dataset.dag <- tabu(small_dataset, score = "bde")
tabu_search.bde.small_dataset.dag <- cextend(tabu_search.bde.small_dataset.dag)

tabu_search.bde.big_dataset.dag <- tabu(big_dataset, score = "bde")
tabu_search.bde.big_dataset.dag <- cextend(tabu_search.bde.big_dataset.dag)
```

4. **Comparar la estructura de las redes obtenidas con las originales. Comentar las diferencias**

En este apartado se ha comparado la red original con la redes que se han aprendido en la sección anterior. Para ello, se han comparado visualmente con la función **graphviz.compare** y con la métrica de distancia **hamming**. La función **graphviz.compare** recibe como primer argumento la red real y como segundo argumento la red aprendida. La notación visual que utiliza es la siguiente:

- Los arcos que son verdaderos positivos son negros
- Los arcos que son falsos positivos (que no existen o tienen otra dirección en la red original) son rojos
- Los arcos que son falsos negativos son azules.

En el gráfico generado se observan las diferencias entre las distintas redes aprendidas y la red original. Es destacable la diferencia que hay entre los métodos basados en test de independencia y aquellos basados en score. Aquellos basados en score suelen obtener un mayor número de arcos y ser más similares a la red original basándonos en la distancia de hamming. Por otra parte, aquellos basados en test de independencia obtienen una distancia mayor en hamming y por lo tanto, una diferencia muy considerable respecto al anterior método.

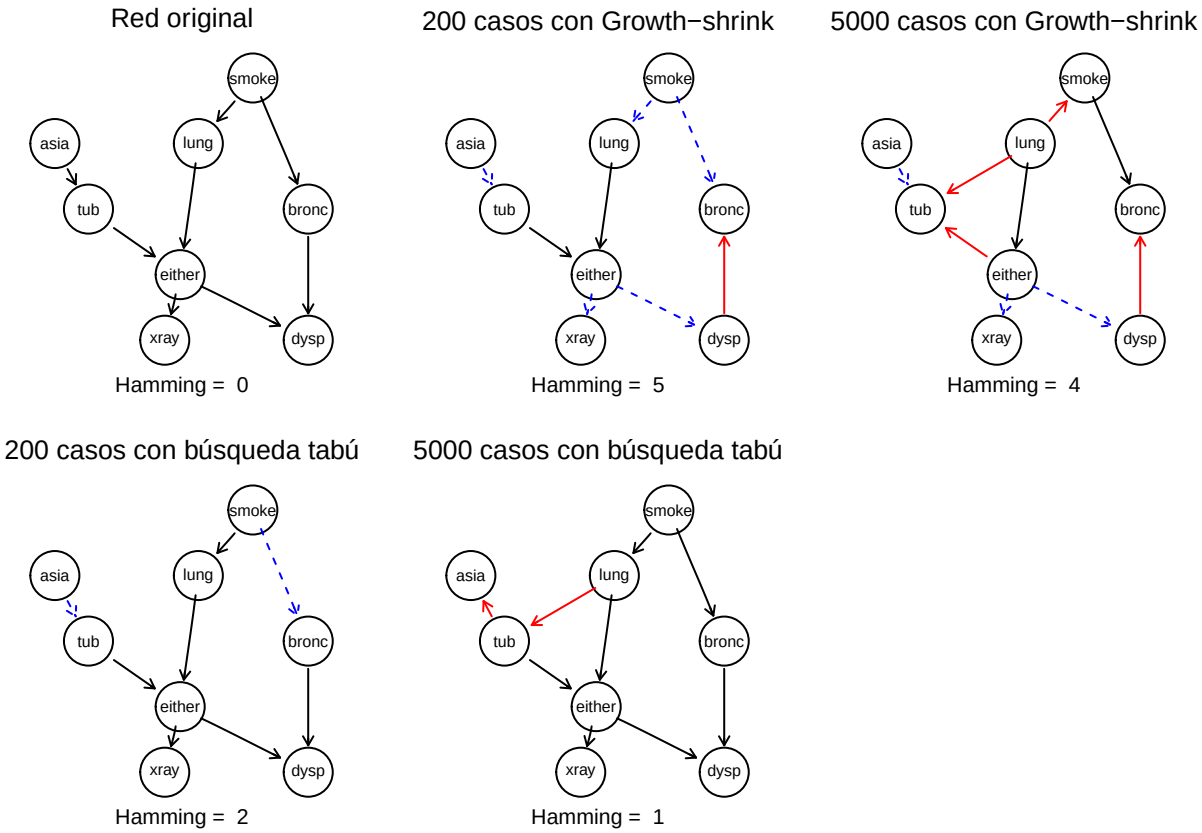
Si nos fijamos en las diferencias visuales, cuando se aprende la estructura con un dataset pequeño suele haber un mayor número de falsos negativos, es decir, arcos que faltan respecto a la red original. Por otro lado, con los conjuntos de datos grandes, se suelen cometer más fallos respecto a añadir arcos de más o en una dirección equivocada.

Si se comparan de manera general los dos métodos, aquellas redes aprendidas con métodos basados en score son mucho más densas que aquellas aprendidas con métodos basados en test de independencia.

```
# Just to convert bn.fit object to bn so we can use graphviz.compare function
asia_network_bn <- bn.net(asia_network)

par(mfrow = c(2, 3))
graphviz.compare(
    asia_network_bn,
    gs.small_dataset.dag,
    gs.big_dataset.dag,
    tabu_search.bde.small_dataset.dag,
    tabu_search.bde.big_dataset.dag,
    main = c(
        "Red original",
        "200 casos con Growth-shrink",
        "5000 casos con Growth-shrink",
        "200 casos con búsqueda tabú",
        "5000 casos con búsqueda tabú"
    ),
    sub = paste(
        "Hamming = ",
        c(
            "0",
            hamming(gs.small_dataset.dag, asia_network_bn),
            hamming(gs.big_dataset.dag, asia_network_bn),
            hamming(tabu_search.bde.small_dataset.dag, asia_network_bn),
            hamming(tabu_search.bde.big_dataset.dag, asia_network_bn)
```

```
        )
    )
)
```



5. **Aprender los parámetros de las redes**

Finalmente, en esta sección se aprenden los parámetros para cada estructura de red aprendida con su conjunto de datos correspondiente. Para todos los casos, se ha utilizado un procedimiento bayesiano para estimar los parámetros y se ha especificado un tamaño muestral equivalente del 10, que está dentro de la recomendación de fijarlo entre 1 y 15. Mientras más grande sea este tamaño muestral equivalente, la distribuciones a posteriori estimadas tenderán hacia una distribución uniforme.

```
gs.small_dataset.bayes <- bn.fit(
    gs.small_dataset.dag,
    data = small_dataset,
    method = "bayes",
    iss = 10
)
gs.small_dataset.bayes
```

```
##
##   Bayesian network parameters
##
##   Parameters of node asia (multinomial distribution)
##
## Conditional probability table:
##         yes         no
```

```
## 0.03333333 0.96666667
##
##   Parameters of node tub (multinomial distribution)
##
## Conditional probability table:
##         yes         no
## 0.04761905 0.95238095
##
##   Parameters of node smoke (multinomial distribution)
##
## Conditional probability table:
##        yes        no
## 0.5095238 0.4904762
##
##   Parameters of node lung (multinomial distribution)
##
## Conditional probability table:
##  yes  no
## 0.1 0.9
##
##   Parameters of node bronc (multinomial distribution)
##
## Conditional probability table:
##
##      dysp
## bronc       yes        no
##   yes 0.7783019 0.2163462
##   no  0.2216981 0.7836538
##
##   Parameters of node either (multinomial distribution)
##
## Conditional probability table:
##
## , , lung = yes
##
##       tub
## either         yes          no
##    yes 0.500000000 0.932432432
##    no  0.500000000 0.067567568
##
## , , lung = no
##
##       tub
## either         yes          no
##    yes 0.833333333 0.006887052
##    no  0.166666667 0.993112948
##
##
##   Parameters of node xray (multinomial distribution)
##
## Conditional probability table:
##       yes       no
## 0.152381 0.847619
##
```

```
##   Parameters of node dysp (multinomial distribution)
##
## Conditional probability table:
##        yes        no
## 0.5047619 0.4952381
```

```
gs.big_dataset.bayes <- bn.fit(
    gs.big_dataset.dag,
    data = big_dataset,
    method = "bayes",
    iss = 10
)
gs.big_dataset.bayes
```

```
##
##   Bayesian network parameters
##
##   Parameters of node asia (multinomial distribution)
##
## Conditional probability table:
##         yes         no
## 0.01277445 0.98722555
##
##   Parameters of node tub (multinomial distribution)
##
## Conditional probability table:
##
## , , either = yes
##
##      lung
## tub            yes           no
##   yes 0.0333333333 0.9731182796
##   no  0.9666666667 0.0268817204
##
## , , either = no
##
##      lung
## tub            yes           no
##   yes 0.5000000000 0.0002668944
##   no  0.5000000000 0.9997331056
##
##
##   Parameters of node smoke (multinomial distribution)
##
## Conditional probability table:
##
##      lung
## smoke       yes        no
##   yes 0.8875000 0.4774841
##   no  0.1125000 0.5225159
##
##   Parameters of node lung (multinomial distribution)
##
## Conditional probability table:
##         yes         no
```

```
## 0.05588822 0.94411178
##
##   Parameters of node bronc (multinomial distribution)
##
## Conditional probability table:
##
## , , dysp = yes
##
##      smoke
## bronc        yes         no
##   yes 0.89246356 0.76235992
##   no  0.10753644 0.23764008
##
## , , dysp = no
##
##      smoke
## bronc        yes         no
##   yes 0.27373921 0.08727429
##   no  0.72626079 0.91272571
##
##
##   Parameters of node either (multinomial distribution)
##
## Conditional probability table:
##
##       lung
## either         yes          no
##    yes 0.991071429 0.009830867
##    no  0.008928571 0.990169133
##
##   Parameters of node xray (multinomial distribution)
##
## Conditional probability table:
##        yes        no
## 0.1125749 0.8874251
##
##   Parameters of node dysp (multinomial distribution)
##
## Conditional probability table:
##        yes        no
## 0.4321357 0.5678643
```

```
tabu_search.bde.small_dataset.bayes <- bn.fit(
    tabu_search.bde.small_dataset.dag,
    data = small_dataset,
    method = "bayes",
    iss = 10
)
tabu_search.bde.small_dataset.bayes
```

```
##
##   Bayesian network parameters
##
##   Parameters of node asia (multinomial distribution)
##
```

```
## Conditional probability table:
##         yes         no
## 0.03333333 0.96666667
##
##   Parameters of node tub (multinomial distribution)
##
## Conditional probability table:
##         yes         no
## 0.04761905 0.95238095
##
##   Parameters of node smoke (multinomial distribution)
##
## Conditional probability table:
##        yes        no
## 0.5095238 0.4904762
##
##   Parameters of node lung (multinomial distribution)
##
## Conditional probability table:
##
##      smoke
## lung         yes         no
##   yes 0.15420561 0.04368932
##   no  0.84579439 0.95631068
##
##   Parameters of node bronc (multinomial distribution)
##
## Conditional probability table:
##  yes  no
## 0.5 0.5
##
##   Parameters of node either (multinomial distribution)
##
## Conditional probability table:
##
## , , lung = yes
##
##       tub
## either         yes          no
##    yes 0.500000000 0.932432432
##    no  0.500000000 0.067567568
##
## , , lung = no
##
##       tub
## either         yes          no
##    yes 0.833333333 0.006887052
##    no  0.166666667 0.993112948
##
##
##   Parameters of node xray (multinomial distribution)
##
## Conditional probability table:
##
```

```
##      either
## xray         yes         no
##   yes 0.90384615 0.04619565
##   no  0.09615385 0.95380435
##
##   Parameters of node dysp (multinomial distribution)
##
## Conditional probability table:
##
## , , either = yes
##
##      bronc
## dysp        yes        no
##   yes 0.7068966 0.8913043
##   no  0.2931034 0.1086957
##
## , , either = no
##
##      bronc
## dysp        yes        no
##   yes 0.7983425 0.1417112
##   no  0.2016575 0.8582888
```

```
tabu_search.bde.big_dataset.bayes <- bn.fit(
    tabu_search.bde.big_dataset.dag,
    data = big_dataset,
    method = "bayes",
    iss = 10
)

tabu_search.bde.big_dataset.bayes
```

```
##
##   Bayesian network parameters
##
##   Parameters of node asia (multinomial distribution)
##
## Conditional probability table:
##
##      tub
## asia         yes         no
##   yes 0.13157895 0.01140723
##   no  0.86842105 0.98859277
##
##   Parameters of node tub (multinomial distribution)
##
## Conditional probability table:
##
##      lung
## tub           yes          no
##   yes 0.037500000 0.009830867
##   no  0.962500000 0.990169133
##
##   Parameters of node smoke (multinomial distribution)
##
```

```
## Conditional probability table:
##        yes        no
## 0.5003992 0.4996008
##
##   Parameters of node lung (multinomial distribution)
##
## Conditional probability table:
##
##      smoke
## lung         yes         no
##   yes 0.09912246 0.01258490
##   no  0.90087754 0.98741510
##
##   Parameters of node bronc (multinomial distribution)
##
## Conditional probability table:
##
##      smoke
## bronc       yes        no
##   yes 0.6208616 0.2918498
##   no  0.3791384 0.7081502
##
##   Parameters of node either (multinomial distribution)
##
## Conditional probability table:
##
## , , lung = yes
##
##       tub
## either          yes           no
##    yes 0.8809523810 0.9953617811
##    no  0.1190476190 0.0046382189
##
## , , lung = no
##
##       tub
## either          yes           no
##    yes 0.9731182796 0.0002668944
##    no  0.0268817204 0.9997331056
##
##
##   Parameters of node xray (multinomial distribution)
##
## Conditional probability table:
##
##      either
## xray         yes         no
##   yes 0.97067901 0.05324370
##   no  0.02932099 0.94675630
##
##   Parameters of node dysp (multinomial distribution)
##
## Conditional probability table:
##
```

```
## , , either = yes
##
##      bronc
## dysp         yes         no
##   yes 0.93213296 0.66376307
##   no  0.06786704 0.33623693
##
## , , either = no
##
##      bronc
## dysp         yes         no
##   yes 0.79052931 0.09158752
##   no  0.20947069 0.90841248
```