# Script e análise dos resultados

install.packages("bnlearn")

library("bnlearn")

# criando um grafo vazio

dag <- empty.graph(nodes = c("A", "S", "E", "O", "R", "T"))

# visualizando o objeto com a estrutura de grafo vazia e seus atributos

dag

Random/Generated Bayesian network

model:

[A][S][E][O][R][T]

nodes: 6

arcs: 0

undirected arcs: 0

directed arcs: 0

average markov blanket size: 0.00

average neighbourhood size: 0.00

average branching factor: 0.00

generation algorithm: Empty

# criando uma relação entre idade e educação

dag <- set.arc(dag, from = "A", to = "E")

# criando uma relação entre sexo e educação

dag <- set.arc(dag, from = "S", to = "E")

# criando uma relação entre educaçao e cargo

dag <- set.arc(dag, from = "E", to = "O")

# criando uma relação entre educação e residência

dag <- set.arc(dag, from = "E", to = "R")

# criando uma relação entre cargo e transporte

dag <- set.arc(dag, from = "O", to = "T")

# criando uma relação entre residência e transporte

dag <- set.arc(dag, from = "R", to = "T")

# visualizando novamente a estrutura da rede

dag

Random/Generated Bayesian network

model:

[A][S][E|A:S][O|E][R|E][T|O:R]

nodes: 6

arcs: 6

undirected arcs: 0

directed arcs: 6

average markov blanket size: 2.67

average neighbourhood size: 2.00

average branching factor: 1.00

generation algorithm: Empty

# nós do grafo

nodes(dag)

[1] "A" "S" "E" "O" "R" "T"

# relações do grafo

arcs(dag)

from to

[1,] "A" "E"

[2,] "S" "E"

[3,] "E" "O"

[4,] "E" "R"

[5,] "O" "T"

[6,] "R" "T"

# outra forma de criar o grafo a partir de uma matriz

dag2 <- empty.graph(nodes = c("A", "S", "E", "O", "R", "T"))

arc.set <- matrix (c("A","E", "S","E","E","O","E","R","O","T","R","T"), byrow = TRUE, ncol = 2, dimnames = list(NULL, c("from","to")))

arcs(dag2) <- arc.set

# validando se os dois objetos de grafo são iguais

all.equal(dag, dag2)

[1] TRUE

# definindo os valores para as variáveis categóricas do grafo

A.lv <- c("young", "adult", "old")

S.lv <- c("M", "F")

E.lv <- c("high", "uni")

O.lv <- c("emp", "self")

R.lv <- c("small", "big")

T.lv <- c("car", "train", "other")

# definindo as probabilidades das observações serem jovem, adulto ou idoso

A.prob <- array(c(0.30, 0.50, 0.20), dim = 3, dimnames = list(A = A.lv))

A.prob

# definindo as probabilidades das observações serem homem ou mulher

S.prob <- array(c(0.60, 0.40), dim = 2, dimnames = list(S = S.lv))

S.prob

# cruzamento das probabilidades do cargo com o nível de educação

O.prob <- array(c(0.96, 0.04, 0.92, 0.08), dim = c(2, 2), dimnames = list(O = O.lv, E = E.lv))

O.prob

# cruzamento das probabilidades do residência com o nível de educação

R.prob <- array(c(0.25, 0.75, 0.20, 0.80), dim = c(2, 2), dimnames = list(R = R.lv, E = E.lv))

R.prob

# cruzamento das probabilidades entre educação, idade e sexo

E.prob <- array(c(0.75, 0.25, 0.72, 0.28, 0.88, 0.12, 0.64,0.36, 0.70, 0.30, 0.90, 0.10), dim = c(2, 3, 2),

dimnames = list(E = E.lv, A = A.lv, S = S.lv))

# cruzamento das probabilidades entre viagem, ocupação e residência

T.prob <- array(c(0.48, 0.42, 0.10, 0.56, 0.36, 0.08, 0.58,

0.24, 0.18, 0.70, 0.21, 0.09), dim = c(3, 2, 2),

dimnames = list(T = T.lv, O = O.lv, R = R.lv))

# combinando o grafo com as distribuições de probabilidade

cpt <- list(A = A.prob, S = S.prob, E = E.prob, O = O.prob, R = R.prob, T = T.prob)

bn <- custom.fit(dag, cpt)

setwd("/Users/raphaelalbino/Dropbox/ead-5917/exercicio\_5/")

# lendo o arquivo da pesquisa

survey <- read.table("survey.txt", header = TRUE)

head(survey)

# estimando a máxima verossimilhança

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

#exemplo de probabilidade cruzada entre ocupação e educação

bn.bayes$O

Parameters of node O (multinomial distribution)

Conditional probability table:

E

O high uni

emp 0.97432432 0.91071429

self 0.02567568 0.08928571

(nlevels(survey[, "T"]) - 1) \* (nlevels(survey[, "E"]) - 1) \* + (nlevels(survey[, "O"]) \* nlevels(survey[, "R"]))

ci.test("T", "E", c("O", "R"), test = "mi", data = survey)

ci.test("T", "E", c("O", "R"), test = "x2", data = survey)

ci.test("T", "E", c("O", "R"), test = "x2", data = survey)

# Os testes retornaram p-values altos, o que diz que na rede desenhada, não há relação de dependência significante entre educação e transporte

# Exceto ocupação e transporte, todos os arcos demonstraram p-value menor do que 0,05

arc.strength(dag, data = survey, criterion = "x2")

from to strength

1 A E 0.0009777168

2 S E 0.0012537013

3 E O 0.0026379469

4 E R 0.0005599201

5 O T 0.4339127237

6 R T 0.0013584250

# validando se há uma ligação entre S e R

dsep(dag, x = "S", y = "R")

[1] FALSE

# validando se há uma ligação entre O e R

dsep(dag, x = "O", y = "R")

[1] FALSE

# validando se E bloqueia o caminho e se S e R são independentes

dsep(dag, x = "S", y = "R", z = "E")

[1] TRUE

# podemos dizer que E é influenciado por A e S, isto é, se soubermos o tipo de educação da pessoa, algumas combinações de idade e sexo serão mais prováveis do que outras

dsep(dag, x = "A", y = "S", z = "E")

[1] FALSE

# tipos de conexão

# serial connections, since both arcs have the same direction and follow one after the other

# divergent connections , because the two arcs have divergent directions from a central node

# convergent connections, because the two arcs converge to a central node

install.packages("gRain")

source("https://bioconductor.org/biocLite.R")

biocLite("RBGL")

biocLite("Rgraphviz")

library("gRain")

junction <- compile(as.grain(bn))

jsex <- setEvidence(junction, nodes = "S", states = "F")

querygrain(jsex, nodes = "T")$T

# A consulta sugere que mulheres tem a mesma preferência quanto ao uso de trem e carro que a amostra de entrevistado completa

T

car train other

0.5620577 0.2806144 0.1573280

jres <- setEvidence(junction, nodes = "R", states = "small")

querygrain(jres, nodes = "T")$T

# Quando as pessoas tem como origem cidades pequenas, a probabilidade de uso de trem aumenta de 0,28 para 0,41. Já a probabilidade combinada de carro e trem sobe de 0,84 para 0,90.

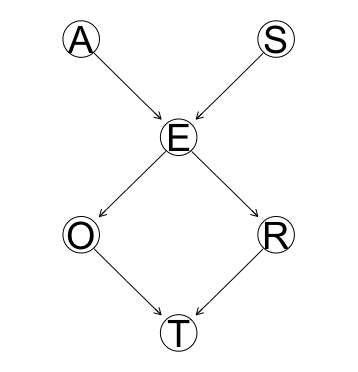
T

car train other

0.48388675 0.41708494 0.09902831

# plotando o grafo

graphviz.plot(dag)



# gráfico de barra para as probabilidades condicionais

bn.fit.barchart(bn.mle$T, main = "Travel", xlab = "Pr(T | R,O)", ylab = "")

