Chapter 1—The discrete case: mulitnomial Bayesian networks

mz

1.1 Introductiory example: Train use survey

```
# preliminaries:
library(bnlearn)
```

```
## Loading required package: methods

##
## Attaching package: 'bnlearn'

## The following object is masked from 'package:stats':
##
##     sigma
```

```
library(vcd)
```

```
## Loading required package: grid
```

```
library(gRain)
```

```
## Loading required package: gRbase

##
## Attaching package: 'gRbase'

## The following objects are masked from 'package:bnlearn':
##
##     ancestors, children, parents
```

note that some caution must be exercised in interpreting both direct and indirect dependencies. The presence of arrows or arcs seems to imply, at an intuitive level, that for each arc one variable should be interpreted as a cause and the other as an effect. This interpretation, which is called causal, is difficult to justify in most situations; for this reason in general we speak about dependence relationships instead of causal effects. s dddd

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

```r
dag <- set.arc(dag, from = "A", to = "E")
dag <- set.arc(dag, from = "S", to = "E")
dag <- set.arc(dag, from = "E", to = "O")
dag <- set.arc(dag, from = "E", to = "R")
dag <- set.arc(dag, from = "O", to = "T")
dag <- set.arc(dag, from = "R", to = "T")
modelstring(dag)
```

```
## [1] "[A][S][E|A:S][O|E][R|E][T|O:R]"
```

```r
arcs(dag)
```

```
##      from to
## [1,] "A"  "E"
## [2,] "S"  "E"
## [3,] "E"  "O"
## [4,] "E"  "R"
## [5,] "O"  "T"
## [6,] "R"  "T"
```

```r
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")

A.prob <- array(c(0.3, 0.5, 0.2), dim = 3,
                dimnames = list(A = A.lv))

S.prob <- array(c(0.6, 0.4), dim = 2,
                dimnames = list(S = S.lv))

R.prob <- array(c(0.25, 0.75, 0.2, 0.8), dim = c(2,2),
                dimnames = list(R = R.lv, E = E.lv))

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

E.prob <- array(c(.75, .25, .72, .28, .88, .12, .64, .36, .70, .30, .90, .10),
                dim = c(2,3,2),
                dimnames = list(E = E.lv, A = A.lv, S = S.lv))

T.prob <- array(c(.48, .42, .10, .56, .36, .08, .58, .24, .18, .70, .21, .09),
                dim = c(3,2,2),
                dimnames = list(T = T.lv, O = O.lv,  R = R.lv))

dag3 <- model2network("[A][S][E|A:S][O|E][R|E][T|O:R]")
all.equal(dag, dag3)
```

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

```r
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)
```

```r
nparams(bn)
```

```
## [1] 21
```

```r
arcs(bn)
```

```
##      from to
## [1,] "A"  "E"
## [2,] "S"  "E"
## [3,] "E"  "O"
## [4,] "E"  "R"
## [5,] "O"  "T"
## [6,] "R"  "T"
```

```r
bn$R
```

```
##
##   Parameters of node R (multinomial distribution)
##
## Conditional probability table:
##
##        E
## R       high  uni
##   small 0.25 0.20
##   big   0.75 0.80
```

```r
coef(bn$R)
```

```
##        E
## R       high  uni
##   small 0.25 0.20
##   big   0.75 0.80
```