CFerrari_Assignment8.Rmd

Charley Ferrari

Wednesday, March 18, 2015

Problem Set 1

Your colleague either commutes by train or by the bus. 20 days of the month, she takes the train and the remaining 10 days she takes the bus. If she takes the train, she reaches work on time with a probability of 0.9. If she takes the bus, she frequently gets stuck in traffic and reaches work on time with a probability of 0.5. Given that she was on time today, what is the probability that she took the bus to work today?

Lets look at the tree of probabilities. First, lets look at whether she takes the train or bus:

```
p.bus <- 1/3 #Probability of taking the bus
p.train <- 2/3 # probability of taking the train
```

Now, lets look at the probabilities of being on time or late given the fact that your colleague takes the bus or train:

```
p.bus.ontime <- 0.5 #probability of being on time if she takes the bus
p.bus.late <- 1-p.bus.ontime #probability of being late if she takes the bus

p.train.ontime <- 0.9 #probability of being on time if she takes the train
p.train.late <- 1-p.train.ontime #probability of being late if she takes the train
```

So we have that our branches, lets calculate the probabilitiies of each node by multiplying the brances:

```
p.bus.ontime.node <- p.bus * p.bus.ontime
p.bus.late.node <- p.bus * p.bus.late
p.train.ontime.node <- p.train * p.train.ontime
p.train.late.node <- p.train * p.train.late
```

We know that our colleague is ontime, so the nodes that we're interested in are p.bus.ontime.node and p.train.ontime.node. Out of these two nodes, we want to find the probability she's at the p.bus.ontime.node.

```
p.bus.given.ontime <- p.bus.ontime.node / (p.bus.ontime.node + p.train.ontime.node)
p.bus.given.ontime
```

```
## [1] 0.2173913
```

Problem Set 2

In the Grade Network that we looked at in the notes, what happens to the probability of Difficulty of Course when you present the evidence that the received recommendation letter was good?

```
library(gRain)
```

```
## Loading required package: gRbase
```

```
yn <- c("yes", "no")
lh <- c("low", "high")

d <- cptable(~difficulty, values=c(7,3), levels=yn)
i <- cptable(~intelligence, values=c(8,2), levels=lh)
g <- cptable(~grade|difficulty:intelligence, values=c(8,2,6,4,1,9,1,99), levels=lh)
s <- cptable(~SAT|intelligence, values=c(9,1,2,8), levels=lh)
l <- cptable(~letter|grade, values=c(9,1,5,95), levels=lh)

net1 <- grain(compileCPT(list(d,i,g,s,l)))

net11 <- setEvidence(net1, nodes="letter", state="high")

querygrain(net11, nodes="difficulty", type="marginal")
```

```
## $difficulty
## difficulty
##       yes        no
## 0.6268561 0.3731439
```

In addition, now present the evidence that both SAT scores were good and the letter of recommendation was good, What is the probability of the Difficulty of Course now?

```
net12 <- setEvidence(net11, nodes="SAT", state="high")

querygrain(net12, nodes="difficulty", type="marginal")
```

```
## $difficulty
## difficulty
##       yes        no
## 0.6676522 0.3323478
```

You should use the gRain package in R to build your network and perform these calculations. You may need to install RBGL package from BioConductor in R to get gRain working. See http://www.bioconductor.org/packages/release/bioc/html/RBGL.html for instructions on RBGL. Please submit your assignment as an R markdown document.

```
library(gRain)

yn <- c("yes", "no")

a <- cptable(~asia, values=c(1,99), levels=yn)
t.a <- cptable(~tub|asia, values=c(5,95,1,99), levels=yn)
s <- cptable(~smoke, values=c(5,5), levels=yn)
l.s <- cptable(~lung|smoke, values=c(1,9,1,99), levels=yn)
b.s <- cptable(~bronc|smoke, values=c(6,4,3,7), levels=yn)
e.lt <- cptable(~either|lung:tub, values=c(1,0,1,0,1,0,0,1), levels=yn)
```

```
x.e <- cptable(~xray|either, values=c(98,2,5,95), levels=yn)
d.be <- cptable(~dysp|bronc:either, values=c(9,1,7,3,8,2,1,9), levels=yn)

plist <- compileCPT(list(a, t.a, s, l.s, b.s, e.lt, x.e, d.be))

net1 <- grain(plist)

net12 <- setEvidence(net1, nodes=c("asia", "dysp"), states=c("yes", "yes"))

net12
```

```
## Independence network: Compiled: TRUE Propagated: TRUE
##   Nodes: chr [1:8] "asia" "tub" "smoke" "lung" "bronc" "either" ...
##   Findings: chr [1:2] "asia" "dysp"
```

```
pEvidence(net12)
```

```
## [1] 0.004501375
```

```
querygrain(net1, nodes="asia", type="marginal")
```

```
## $asia
## asia
##  yes   no
## 0.01 0.99
```

```
querygrain(net1, nodes=c("asia", "tub"), type="joint")
```

```
##      tub
## asia     yes     no
##   yes 0.0005 0.0095
##   no  0.0099 0.9801
```

```
querygrain(net1, nodes=c("asia", "dysp", type="joint"))
```

```
## $asia
## asia
##  yes   no
## 0.01 0.99
##
## $dysp
## dysp
##       yes        no
## 0.4359706 0.5640294
```