BLin_Assign8

Bin Lin

2017-3-25

1. 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?

P(T) = 20/30

P(B) = 10/30

P(On Time|T) = 0.9

P(On Time|B) = 0.5

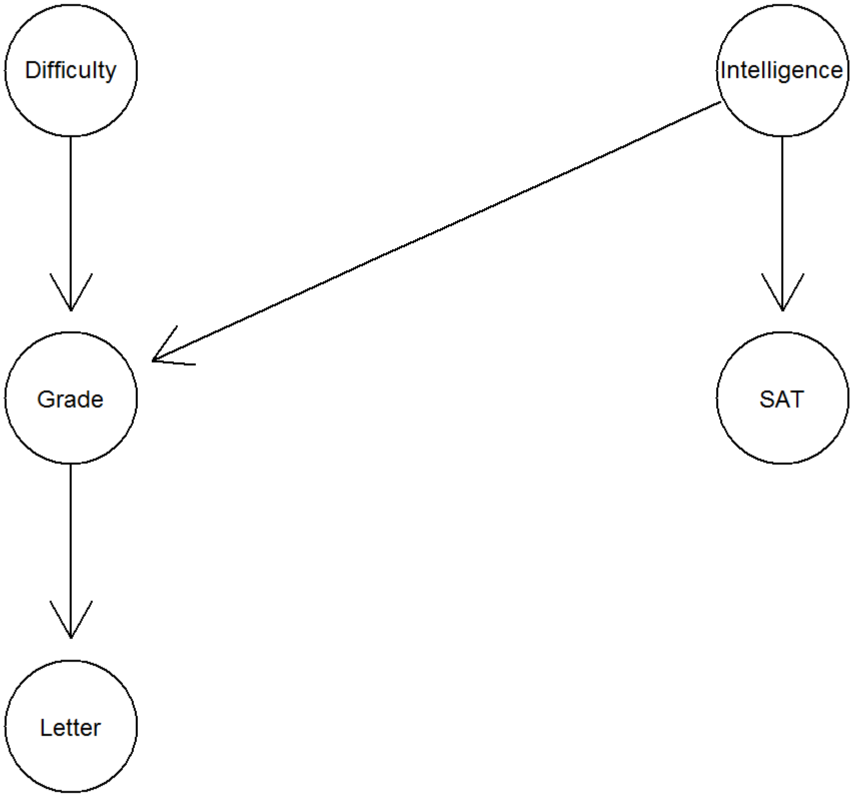
$$P(B|OnTime) = \frac{P(OnTime|B) \times P(B)}{P(OnTime|B) \times P(B) + P(OnTime|T) \times P(T)} = \frac{0.5 \times (10/30)}{0.5 \times (10/30) + 0.9 \times (20/30)} = 0.2173913$$

2. 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? 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? 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 (http://www.bioconductor.org/packages/release/bioc/html/RBGL.html) for instructions on RBGL.

```
#source("http://bioconductor.org/biocLite.R")
#biocLite("RBGL")
#biocLite("Rgraphviz")
#biocLite("graph")
#browseVignettes("RBGL")

suppressMessages(suppressWarnings(library(gRain)))
suppressMessages(suppressWarnings(library(Rgraphviz)))


#1) Creates conditional probability tables
no_yes <- c("NO", "YES")
low_high <- c("LOW", "HIGH")
D <- cptable(~Difficulty, values = c(0.3, 0.7), levels = no_yes)
I <- cptable(~Intelligence, values = c(0.8, 0.2), levels = low_high)
S <- cptable(~SAT | Intelligence, values = c(0.9, 0.1, 0.2, 0.8), levels = low_high)
G <- cptable(~Grade | Intelligence:Difficulty, values = c(0.6, 0.4, 0.01, 0.99, 0.8, 0.2, 0.1,
0.9), levels = low_high)
L <- cptable(~Letter | Grade, values = c(0.9, 0.1, 0.05, 0.95), levels = low_high)


#2) Create Baysian network from data and plot the network graph.
cptlist <- compileCPT(list(D, I, S, G, L))
cptlist
```

```
## CPTspec with probabilities:
##  P( Difficulty )
##  P( Intelligence )
##  P( SAT | Intelligence )
##  P( Grade | Intelligence Difficulty )
##  P( Letter | Grade )
```

```
bn <- grain(cptlist)
plot(bn)
```



```
#3) Compiles a Bayesian network and display all the nodes.
bnc <- compile(bn, propagate=TRUE)
querygrain(bnc, nodes = nodeNames(bnc), type = "marginal", evidence = NULL, exclude = TRUE, normalize = TRUE, result = "array", details = 0)
```

```
## $Difficulty
## Difficulty
##  NO YES
## 0.3 0.7
##
## $Intelligence
## Intelligence
##  LOW HIGH
##  0.8  0.2
##
## $Grade
## Grade
##    LOW   HIGH
## 0.6066 0.3934
##
## $SAT
## SAT
##  LOW HIGH
## 0.76 0.24
##
## $Letter
## Letter
##     LOW    HIGH
## 0.56561 0.43439
```

```
#4) Query an independence network for course difficulty, given that letter was good.
evidence1 <- setEvidence(bnc, evidence=list(Letter = "HIGH"))

querygrain(evidence1, nodes = "Difficulty")
```

```
## $Difficulty
## Difficulty
##        NO       YES
## 0.3731439 0.6268561
```

```
#5) Query an independence network for course difficulty, given that both SAT and letter were goo
d.
evidence1 <- setEvidence(bnc, evidence=list(Letter = "HIGH", SAT = "HIGH"))
querygrain(evidence1, nodes = "Difficulty")
```

```
## $Difficulty
## Difficulty
##        NO       YES
## 0.3323478 0.6676522
```