

Project 7

Contents

Problem 19	1
Message passing on a chain	1
(a) Store clique potentials in an R object	1
(b) Computing forward messages	2
(c) Computing backward message	2
(d) Compute the marginal probability distribution for each no	3

Problem 19

Message passing on a chain

We will now use R to perform the message passing shown in Figure 1. Suppose that we have the following (conditional) probability distributions:

```
# Conditional probabilities
x1 <- 1/3
x2_1 <- c(4/5, 2/3)
x3_2 <- c(5/7, 1/3)
x4_3 <- c(3/5, 2/5)
x5_4 <- c(1/2, 7/9)
```

Note that these equations fully determine each (conditional) probability distribution, since $X_i \in \{0, 1\}$ for $i \in \{1, \dots, 5\}$.

(a) Store clique potentials in an R object

```
# Potential matrix
pot <- array(dim = c(2, 2, 4), dimnames = list(c("0", "1"), c("0", "1"),
c("Psi12", "Psi23", "Psi34", "Psi45")))

# Filling it up
pot[, , "Psi12"] <- cbind(1-x2_1, x2_1) * c(1-x1, x1)
pot[, , "Psi23"] <- cbind(1-x3_2, x3_2)
pot[, , "Psi34"] <- cbind(1-x4_3, x4_3)
pot[, , "Psi45"] <- cbind(1-x5_4, x5_4)

pot

## , , Psi12
##
##      0      1
## 0 0.1333333 0.5333333
## 1 0.1111111 0.2222222
##
## , , Psi23
```

```
##
##          0          1
## 0 0.2857143 0.7142857
## 1 0.6666667 0.3333333
##
## , , Psi34
##
##          0          1
## 0 0.4 0.6
## 1 0.6 0.4
##
## , , Psi45
##
##          0          1
## 0 0.5000000 0.5000000
## 1 0.2222222 0.7777778
```

(b) Computing forward messages

```
# Forward message
mu_a <- array(dim = c(5, 2), dimnames = list(c("X1", "X2", "X3", "X4", "X5"), c("0", "1")))

# Initialization
mu_a[1, ] <- 1

# Computation
for (i in 2:5){
  mu_a[i, ] <- mu_a[i-1, ] %*% pot[, , i-1]
}

mu_a
```

```
##          0          1
## X1 1.0000000 1.0000000
## X2 0.2444444 0.7555556
## X3 0.5735450 0.4264550
## X4 0.4852910 0.5147090
## X5 0.3570253 0.6429747
```

(c) Computing backward message

```
# Backward message
mu_b <- array(dim = c(5, 2), dimnames = list(c("X1", "X2", "X3", "X4", "X5"), c("0", "1")))

# Initialization
mu_b[5, ] <- 1

# Computation
for (i in 4:1){
  mu_b[i, ] <- mu_b[i+1, ] %*% pot[, , i]
}

mu_b
```

```
##           0           1
## X1 0.2429159 0.7341564
## X2 0.9312169 1.0687831
## X3 1.0555556 0.9444444
## X4 0.7222222 1.2777778
## X5 1.0000000 1.0000000
```

(d)) Compute the marginal probability distribution for each no

```
# Marginal prob.
marg <- array(dim = c(5, 2), dimnames = list(c("X1", "X2", "X3", "X4", "X5"), c("0", "1")))

# Calculate
for (i in 1:5){
  marg[i, ] <- mu_a[i, ] * t(mu_b)[, i]
}

marg
```

```
##           0           1
## X1 0.2429159 0.7341564
## X2 0.2276308 0.8075250
## X3 0.6054086 0.4027631
## X4 0.3504879 0.6576837
## X5 0.3570253 0.6429747
```

Normalizing constant Z

```
Z <- rowSums(marg)
Z
```

```
##           X1           X2           X3           X4           X5
## 0.9770723 1.0351558 1.0081717 1.0081717 1.0000000
```

Normalizing

```
marg <- marg/Z
print(rowSums(marg))
```

```
## X1 X2 X3 X4 X5
## 1 1 1 1 1
```

```
print(marg)
```

```
##           0           1
## X1 0.2486161 0.7513839
## X2 0.2199000 0.7801000
## X3 0.6005015 0.3994985
## X4 0.3476471 0.6523529
## X5 0.3570253 0.6429747
```

```
library(rmarkdown)
render("Gibbs.Rmd", pdf_document(TRUE), "Indilewitsch_Toidze_Houhamdi_Pudziuelyte_Project6.pdf")
```