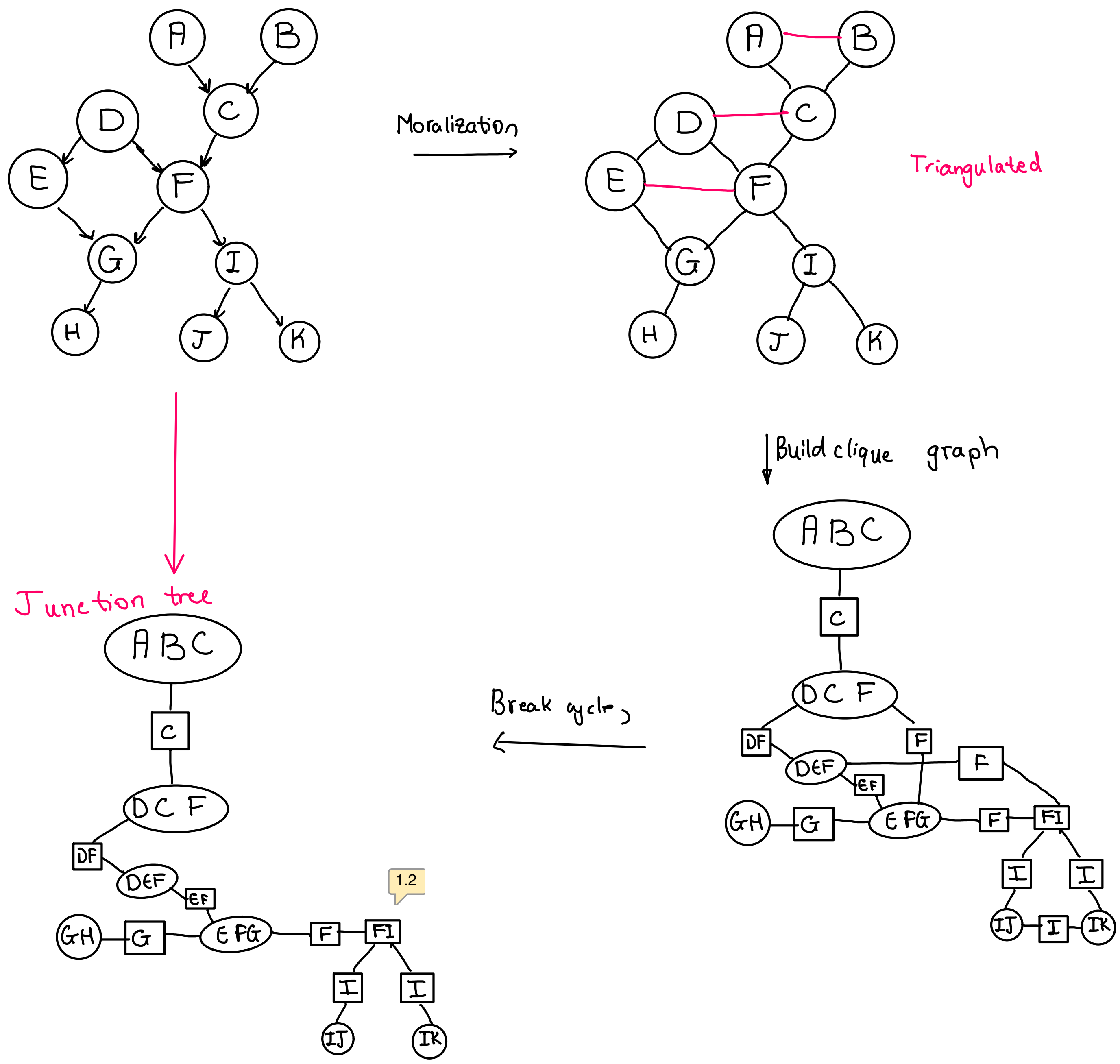
Exercise 7

Tuesday, April 16, 2024 15:05

Problem 17

(a) $U = \{A, \dots, K\}$



(b) $P(U) \rightarrow$ product of clique potentials divided by product of the separator potentials

$$P(U) = \frac{\psi(A,B,C)\cdot\psi(D,C,F)\cdot\psi(D,E,F)\cdot\psi(E,F,G)\cdot\psi(G,H)\cdot\psi(F,I)\cdot\psi(I,J)\cdot\psi(I,K)}{\psi(C)\cdot\psi(D,F)\cdot\psi(E,F)\cdot\psi(G)\cdot\psi(F)\cdot\psi(I)^2}$$

PROBLEM 18

2.1

a) FORWARD

$$\mu_\alpha(x_n) = \sum_{x_{n-1}} \psi_{n-1,n}(x_{n-1}, x_n)\, \mu_\alpha(x_{n-1})$$

BACKWARD

$$\mu_\beta(x_n) = \sum_{x_{n+1}} \psi_{n,n+1}(x_n, x_{n+1})\, \mu_\beta(x_{n+1})$$

b) LET'S COMPUTE ∀ FORWARD MESSAGES NEEDED FOR THE $P(X_4 = 1)$

$\mu_\alpha(X_2 = k) \dots \mu_\alpha(X_4 = k)$

FOR A FIXED k WE NEED K SUM. STEPS FOR $\mu_\alpha(X_i = k)$.
FOR ∀ POSSIBLE k WE NEED K^2 SUM. STEPS.

THE FOLLOWING $\mu_\alpha(X_{i+1} = k)$ NEEDS K SUM. STEPS FOR A FIXED k IF ALL MESSAGES $\mu_\alpha(X_i = k)$ WERE STORED. FOR ∀ POSSIBLE k, WE NEED K^2 SUM. STEPS.

THEREFORE, SINCE WE NEED TO COMPUTE $\mu_\alpha(X_2 = k) \dots \mu_\alpha(X_4 = k)$ FORWARD MESSAGES FOR $P(X_4 = 1)$, WE WILL HAVE

3.1

$3 \cdot K^2$ SUMMATION STEPS

$\mu_\beta(X_4 = k)$ FOR ∀ k WE WILL NEED K^2 SUMMATION STEPS.

AND TO SUM PRODUCTS $\mu_\alpha(X_4 = k)\mu_\beta(X_4 = k)$ WE WILL NEED ADDITIONAL K STEPS.

$$P(X_4 = 1) = \frac{1}{Z} \mu_\alpha(X_4 = 1)\, \mu_\beta(X_4 = 1)$$

Z - NEEDS $(3K^2 + K^2 + K) = 4K^2 + K$ STEPS

$\mu_\alpha(X_4 = 1)$ & $\mu_\beta(X_4 = 1)$ ARE ASSUMED TO BE STORED & NOT COMPUTED AGAIN.

THEREFORE WE NEED $4K^2 + K$ STEPS TO COMPUTE $P(X_4 = 1)$

COMPLEXITY IN "BIG O" NOTATION: $O(NK^2)$

c) TO COMPUTE $\forall$ MARGINAL DISTRIBUTIONS, WE NEED TO COMPUTE $\forall$ POSSIBLE MESSAGES.

$$\begin{cases} \text{FOR FIXED } k: \ \mu_\alpha(x_2 = k) : \ \text{NEEDS } K \text{ STEPS} \\ \text{FOR } \forall k \text{ WE WILL NEED } K^2 \text{ STEPS.} \end{cases}$$

$$\begin{cases} \text{FOR FIXED } k \ \mu_\alpha(x_3 = k) \ \& \text{ KNOWING } \forall \mu_\alpha(x_2 = k) \text{ ALREADY COMPUTED} \\ \text{WE NEED } K \text{ STEPS. FOR } \forall k, \text{ WE WILL NEED } K^2 \text{ STEPS} \end{cases}$$

... IT IS APPLIED FOR $\mu_\alpha(x_2 = k) \dots \mu_\alpha(x_5 = k)$. THEREFORE WE NEED $(5-1)K^2 = 4K^2$ STEPS TO COMPUTE $\forall$ FORWARD MESSAGES.

FOR BACKWARD MESSAGES WE START FROM $\mu_\beta(x_4 = k)$ & CONTINUE UNTIL $\mu_\beta(x_1 = k)$ FOR $\forall k$ ARE COMPUTED, WHICH ALSO RESULTS IN $4K^2$ SUMMATION STEPS.

TO COMPUTE ONE MARGINAL PROBABILITY $P(x_i = k)$ (FOR FIXED k) WE NEED ADDITIONAL K SUMMATION STEPS (FOR Z COMPUTATION)

WE HAVE 5 NODES & K POSSIBLE NODE STATES, THUS TO COMPUTE $\forall$ MARGINAL DISTRIBUTIONS WE NEED

$$(4K^2 + 4K^2 + K) \cdot 5 \cdot (K-1) = (8K^2 + K)\,5\,(K-1) \text{ SUMMATION STEPS}$$

IN GENERAL, FOR CHAIN OF LENGTH N, WE NEED

$$(2(N-1)K^2 + K)\,N\,(K-1) \quad \text{SUMMATION STEPS}$$

COMPLEXITY IN "BIG O" NOTATION: $O(N^2K^3)$

4.1

Project 7

Contents

5.1

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 $i \in \{1, ..., 5\}$.

(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

```r
# 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

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

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

Normalizing

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

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

```r
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
```

```r
library(rmarkdown)
render("Gibbs.Rmd", pdf_document(TRUE), "Indilewitsch_Toidze_Houhamdi_Pudziuvelyte_Project6.pdf")
```

Index of comments

1.1 2/2

1.2 FI is probably not meant as a separator-node here.

2.1 2/3

3.1 some symbols here are rather difficult to discern in handwriting

4.1 -1: Complexity without storing messages is naively O(N * NK^2). Using stored messages it is O(NK^2).

5.1 4.5/5

6.1 -0.5: should be transposed here.
mu_b[i,] <- mu_b[i+1,] %*% t(pot[, , i])