Projecy 7 Exact inference in graphical models

Team C - Minghang Li, Xiaocheng Yang, Xinyi Chen

April 18, 2024

Problem 17: Junction tree algorithm

(a) Build the *junction tree* of the network

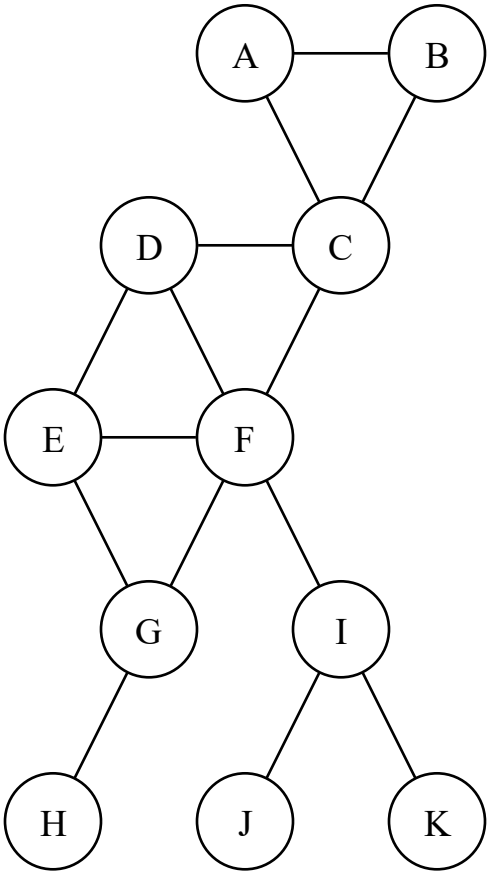

Figure 1: Moral graph

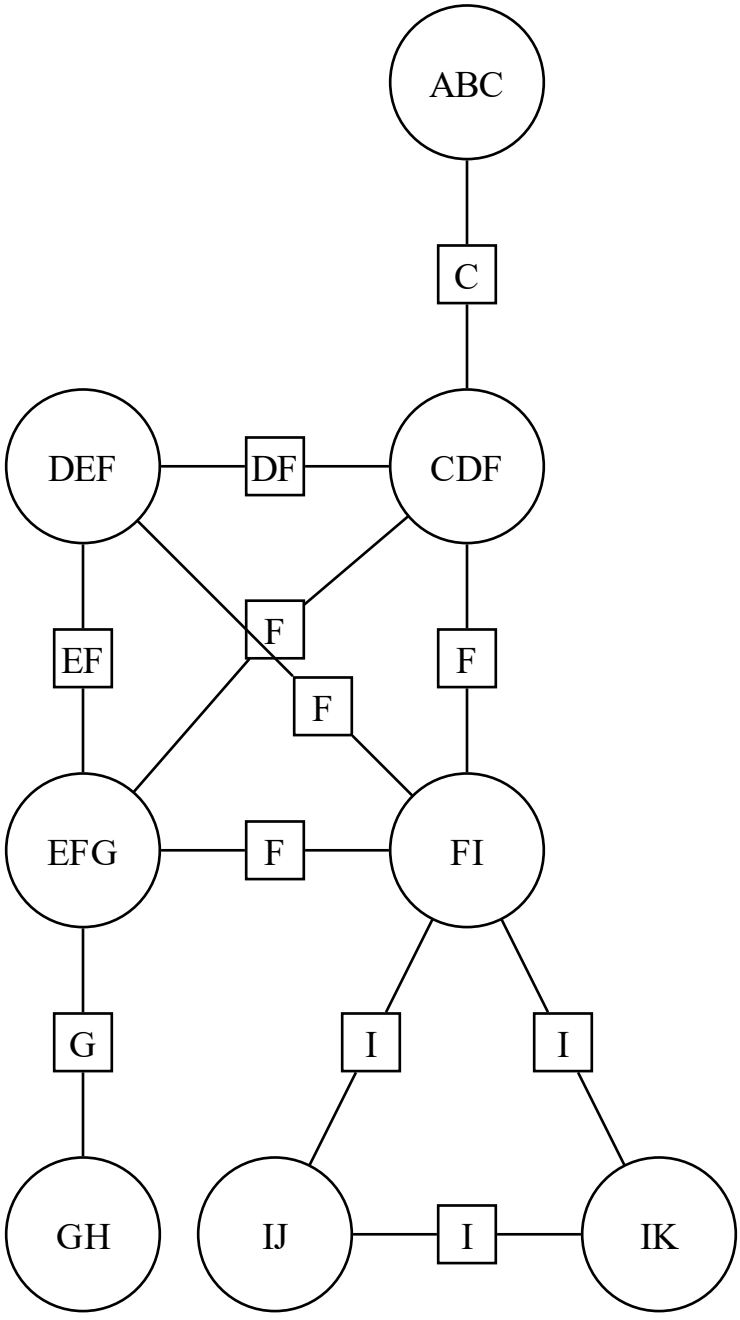

Figure 2: Junction graph

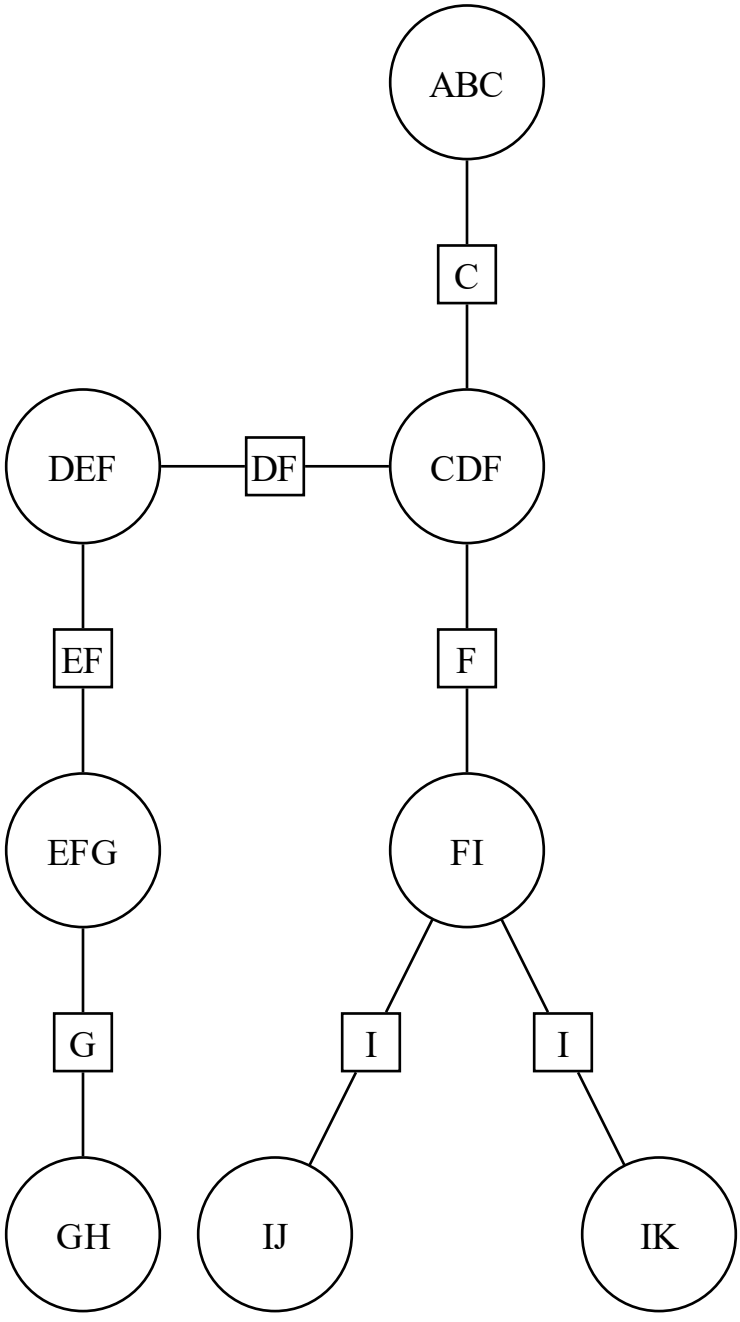

Figure 3: Junction tree

(b) Write the joint probability $P(U)$ in terms of the cluster and separator potentials

$$P(U) = \frac{\psi(\text{clique})}{\psi(\text{separator})}$$
$$= \frac{\psi(ABC)\psi(CDF)\psi(DEF)\psi(EFG)\psi(FI)\psi(GH)\psi(IJ)\psi(IK)}{\psi(C)\psi(DF)\psi(EF)\psi(F)\psi(G)\psi(I)^2}$$

Problem 18: Benefit of storing messages

Formula for recursively computing forward and backward messages

Forward message

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

Initialization:

$$\mu_1(x_1) = 1 \quad (\text{so that } \mu_\alpha(x_2) = \sum_{x_1} \psi_{1,2}(x_1, x_2)\mu_1(x_1)...)$$

Backward message

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

Initialization:

$$\mu_5(x_5) = 1 \quad (\text{so that } \mu_\beta(x_4) = \sum_{x_5} \psi_{4,5}(x_4, x_5)\mu_5(x_5)...)$$

(b) Complexity for computing marginal probabiliy $P(X_4 = 1)$

It's $O(NK^2)$.

Each node contains K values. Forward $(n-1)K^2$, backward $(N-n)K^2$). Total = $(N-1)K^2 \to O(NK^2)$.

(c) Complexity for computing all marginal probability distributions if messages are stored

If values are stored, after doing one message passing in forward directoin and one message passing in backward direction, we can directly "read off" all probabilities. The total calculation is then $2(N-1)K^2$, or $O(NK^2)$.

In the general case, it will be $N(N-1)K^2$, or $O(N^2K^2)$.

Problem 19: Message passing on a chain

(a) Store clique potentials in an R object

```
# use the hint!
clique_potentials <- array(dim = c(2, 2, 4),
                           dimnames = list(c("0", "1"), c("0", "1"),
                                           c("Psi12", "Psi23", "Psi34", "Psi45")))
```

```
X1 <- c(2/3, 1/3)
X2 <- c(4/5, 2/3)
X3 <- c(5/7, 1/3)
X4 <- c(3/5, 2/5)
X5 <- c(1/2, 7/9)
```

```
clique_potentials[, 1, "Psi12"] <- X1 * (1 - X2)
clique_potentials[, 2, "Psi12"] <- X1 * X2
clique_potentials[,  , "Psi23"] <- c(1 - X3, X3)
clique_potentials[,  , "Psi34"] <- c(1 - X4, X4)
clique_potentials[,  , "Psi45"] <- c(1 - X5, X5)
clique_potentials
```

```
## , , 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

```
fwd_msg <- array(dim = c(5, 2),
                 dimname = list(c("X1", "X2", "X3", "X4", "X5"),
                                c("0", "1")))
fwd_msg["X1", ] <- c(1, 1)

foreach(i = 2:5) %do% {
  fwd_msg[i, ] <- fwd_msg[i - 1, ] %*% clique_potentials[, , i - 1]
}
```

```
fwd_msg
```

```
##            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 messages

Initialize in such shape for easy computation :)

```
bkwd_msg <- array(dim = c(2, 5),
                  dimname = list(c("0", "1"),
                                 c("X1", "X2", "X3", "X4", "X5")))
bkwd_msg[, "X5"] <- c(1, 1)
foreach (i = 4:1) %do% {
  bkwd_msg[, i] <- clique_potentials[, , i] %*% bkwd_msg[, i + 1]
}
```

```
bkwd_msg
```

```
##          X1 X2 X3 X4 X5
## 0 0.6666667  1  1  1  1
## 1 0.3333333  1  1  1  1
```

(d) Compute the marginal probability distribution for each node

```
marginal <- array(0, dim = c(5, 2),
                  dimname = list(c(
                    "P(X1)", "P(X2)", "P(X3)", "P(X4)", "P(X5)"
                  ),
                  c("0", "1")))
foreach (i = 1:5) %do% {
  marginal[i,] <- fwd_msg[i,] * bkwd_msg[, i]
}
Z <- rowSums(marginal)
marginal <- marginal / Z
```

```
marginal
```

```
##                0         1
## P(X1) 0.6666667 0.3333333
## P(X2) 0.2444444 0.7555556
## P(X3) 0.5735450 0.4264550
## P(X4) 0.4852910 0.5147090
## P(X5) 0.3570253 0.6429747
```

The normalising constant Z is 1.