Assignment 4 SSY316

Matthew Newson, Erik Norlin

November 2023

Exercise 1

To prove that there are $\frac{2M(M-1)}{2}$ distinct undirected graphs over a set of M distinct random variables, let's consider the process of forming an undirected graph.

There are $\binom{M}{2}$ ways to choose 2 vertices from M distinct vertices, and for each pair, there are 2 possibilities. So, the total number of distinct undirected graphs is given by:

$$\text{Total number of graphs} = 2^{\binom{M}{2}}$$

Now, let's simplify this expression:

$$\binom{M}{2} = \frac{M!}{(M-2)!2!} = \frac{M \cdot (M-1)}{2}$$

Substituting this back into the expression for the total number of graphs:

$$\text{Total number of graphs} = 2^{\frac{M \cdot (M-1)}{2}}$$

The 8 possibilities for the case of M=3 are the following :

- 0 edges all disconnected (a b c) (1 graph) - 1 edge (a-b c, b-c a, c-a b) (3 graphs) - 2 edges (a-b-c, a-c-b, b-a-c) (3 graphs) - 3 edges all connected (a-b-c-a) (1 graph)

Exercise 2

The node b is observed as true, so the first message passed is $\mu_{b \to f}(b) = \delta(b)$

$$\begin{aligned}
\mu_{f \to a}(a) &= \sum_b f(a, b)\mu_{b \to f}(b) \\
&= \sum_b \left(0.9\delta(a)\delta(b) + 0.2\delta(\neg a)\delta(b) + 0.1\delta(a)\delta(\neg b) + 0.8\delta(\neg a)\delta(\neg b)\right)\delta(b) \\
&= 0.9\delta(a) + 0.2\delta(\neg a).
\end{aligned}$$

The Bern factor sends a message to the node a it is given by the following:

$$\mu_{\text{Bern}(a;0.3) \to a}(a) = 0.3\delta(a) + 0.7\delta(\bar{a})$$

$$\begin{aligned}
p(a) \propto \mu_a(a) &= \mu_{\text{Bern}(a;0.3) \to a}(a) \cdot \mu_{f \to a}(a) \\
&= (0.3\delta(a) + 0.7\delta(\bar{a}))(0.9\delta(a) + 0.2\delta(\bar{a})) \\
&= (0.27\delta(a) + 0.14\delta(\bar{a}))
\end{aligned}$$

Normalizing the local marginal we get

$$Z_a = \sum_a \mu_a(a) = \mu_a(\text{true}) + \mu_a(\text{false}) = 0.27 + 0.14 = 0.41$$

and thus we get the following:

$$p(a) = \frac{\mu_a(a)}{Z_a} \approx 0.659\delta(a) + 0.341\delta(\neg a).$$

Exercise 3

a)

First we shall compute the marginal distribution of a.
The leaf node b transmits the message $\mu_{b\to f_2}(b) = 1$ to factor f_2. Factor f_2 then calculates and forwards the message

$$\mu_{f_2\to a}(a) = \int f_2(a,b)\mu_{b\to f_2}(b)\,db = \int N(b;\alpha a, \sigma_2^2)\,db = 1.$$

On the other hand, leaf node f_1 conveys the message $\mu_{f_1\to a}(a) = N(a;\mu_1,\sigma_1^2)$ to node a.
The marginal distribution of a is then determined by

$$p(a) \propto \mu_a(a) = \mu_{f_1\to a}(a)\cdot\mu_{f_2\to a}(a) = N(a;\mu_1,\sigma_1^2)\cdot 1 = N(a;\mu_1,\sigma_1^2).$$

Hence, the graphical model yields $p(a) = N(a;\mu_1,\sigma_1^2)$.

b)

Now we shall look at the marginal distribution of b.
The leaf factor f_1 sends the message to the variable node a.

$$\mu_{f_1\to a}(a) = N(a;\mu_1,\sigma_1^2)$$

Node a forwards this message to f_2 :

$$\mu_{a\to f_2}(a) = \mu_{f_1\to a}(a) = N(a;\mu_1,\sigma_1^2)$$

and f_2 forwards the message as

$$\mu_{f_2\to b}(b) = \int f_2(a,b)\mu_{a\to f_2}(a)\,da = \int N(b;\alpha a,\sigma_2^2)N(a;\mu_1,\sigma_1^2)\,da = N(b;\alpha\mu_1,\alpha^2\sigma_1^2+\sigma_2^2)$$

where we used results from conditional Gaussians to get the marginal $p(b) = \mu_{f_2\to b}(b)$. Hence, the graphical model yields $p(b) = N(b;\alpha\mu_1,\alpha^2\sigma_1^2+\sigma_2^2)$.

Exercise 4

a)

Let $x :=$ clips, $y :=$ pins, and $Q :=$ high quality steel, then the conditional probabilities in the model are

$$\begin{cases} p(x|Q) = P(x|\lambda=10)\delta(Q) + P(x|\lambda=7)\delta(\bar{Q}) \\ p(y|Q) = P(y|\lambda=10)\delta(Q) + P(y|\lambda=7)\delta(\bar{Q}) \end{cases}$$

Figure 1 shows the Bayesian Network of the model. The arrows point from the quality of steel to the production of clips and pins indicating that the production of clips x and pins y are conditioned on the steel's quality.

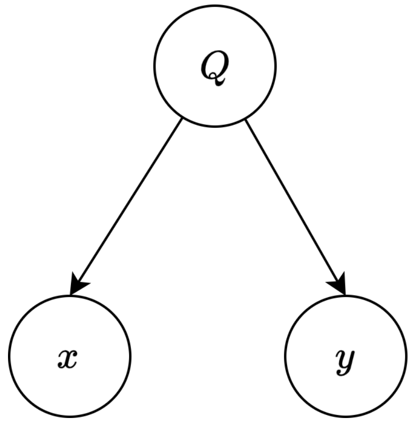

Figure 1: Bayesian Network of the model.

b)

Figure 2 shows a factor graph of the Bayesian Network in Figure 1. A factor f is added to every node. Since we have observed 10 produced clips and 8 produced pins we add two leaf factors defined as the dirac delta function.

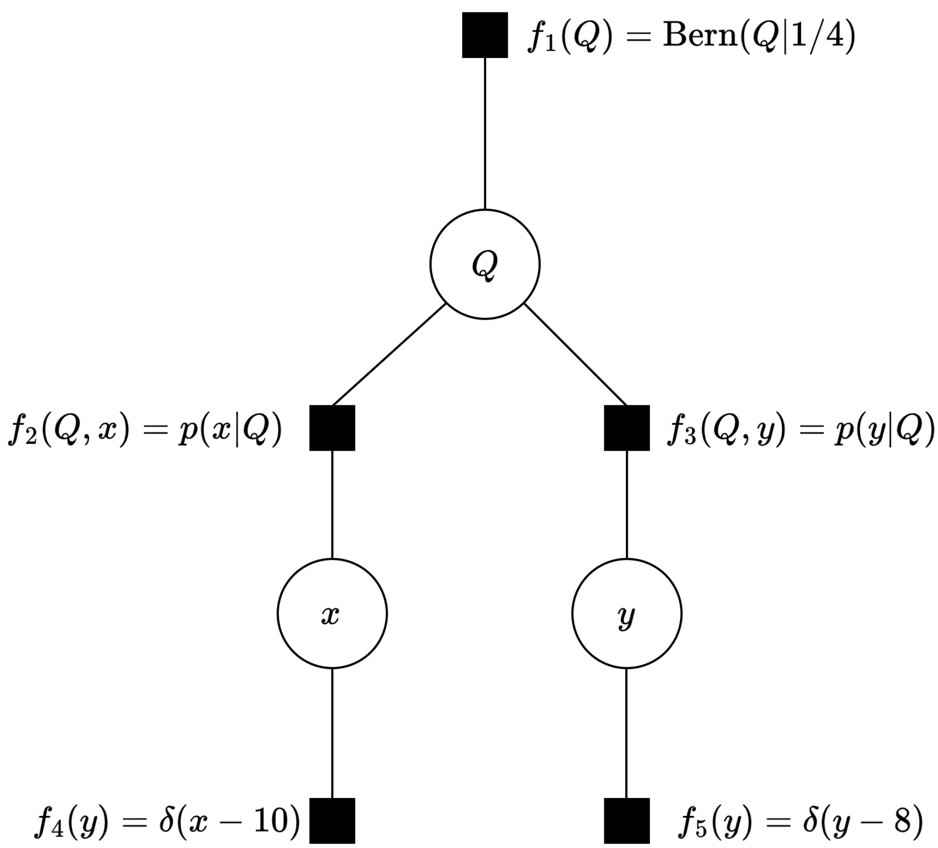

Figure 2: Factor graph of the Bayesian Network.

c)

We want to compute the probability that the company was using high-quality steel, i.e., $p(Q = \texttt{True})$, with message passing. We define this probability as

$$p(Q) \propto \mu_Q(Q) = \mu_{f_1 \to Q}(Q) \cdot \mu_{f_2 \to Q}(Q) \cdot \mu_{f_3 \to Q}(Q)$$

where

$$\begin{aligned}
\mu_{f_1 \to Q}(Q) &= f_1(Q) \\
&= \mathrm{Bern}(Q|1/4) \\
&= \frac{1}{4}\delta(Q) + \frac{3}{4}\delta(\bar{Q})
\end{aligned}$$

and

$$\begin{aligned}
\mu_{f_2 \to Q}(Q) &= \int f_2(Q, x) \cdot \mu_{x \to f_2}(x) dx \\
&= \int f_2(Q, x) \cdot \mu_{f_4 \to x}(x) dx \\
&= \int f_2(Q, x) \cdot \delta(x - 10) dx \\
&= f_2(Q, 10) \\
&= p(x = 10|Q) \\
&= P(x = 10|\lambda) \\
&= \frac{10^{10}\exp(-10)}{10!}\delta(Q) + \frac{7^{10}\exp(-7)}{10!}\delta(\bar{Q})
\end{aligned}$$

as well as

$$\begin{aligned}
\mu_{f_3 \to Q}(Q) &= \int f_3(Q, y) \cdot \mu_{y \to f_3}(y) dy \\
&= \int f_3(Q, y) \cdot \mu_{f_5 \to y}(y) dy \\
&= \int f_3(Q, y) \cdot \delta(y - 8) dy = f_3(Q, 8) \\
&= p(y = 8|Q) \\
&= P(y = 8|\lambda) \\
&= \frac{10^{8}\exp(-10)}{8!}\delta(Q) + \frac{7^{8}\exp(7)}{8!}\delta(\bar{Q})
\end{aligned}$$

Thus, we have that

$$\begin{aligned}
p(Q|x = 10, y = 8) \propto \mu_Q(Q) &= \frac{1}{4} \cdot \frac{10^{10}\exp(-10)}{10!} \cdot \frac{10^{8}\exp(-10)}{8!}\delta(Q) + \frac{3}{4} \cdot \frac{7^{10}\exp(-7)}{10!} \cdot \frac{7^{8}\exp(7)}{8!}\delta(\bar{Q}) \\
&= \frac{10^{18}\exp(-20)}{4 \cdot 10!8!}\delta(Q) + \frac{3 \cdot 7^{18}\exp(-14)}{4 \cdot 10!8!}\delta(\bar{Q})
\end{aligned}$$

We normalise $\mu_Q(Q)$ as

$$
\begin{aligned}
Z_Q &= \sum_Q \mu_Q(Q) \\
&= \mu_Q(\texttt{True}) + \mu_Q(\texttt{False}) \\
&= \frac{10^{18}\exp(-20) + 3 \cdot 7^{18}\exp(-14)}{4 \cdot 10!8!}
\end{aligned}
$$

Finally, we have that

$$
\begin{aligned}
p(Q = \texttt{True}|x = 10, y = 8) &= \frac{10^{18}\exp(-20)}{4 \cdot 10!8!} \Big/ Z_Q \\
&= \frac{10^{18}\exp(-20)}{10^{18}\exp(-20) + 3 \cdot 7^{18}\exp(-14)} \\
&\approx 0.3366
\end{aligned}
$$

d)

Listing 1 shows a script of a Monte Carlo simulation that simulates an estimated probability $\hat{p}(Q = \texttt{True}|x = 10, y = 8)$ of $p(Q = \texttt{True}|x = 10, y = 8)$. The output of this program was $\hat{p}(Q = \texttt{True}|x = 10, y = 8) = 0.3379$ which is close enough to verify the claim that $p(Q = \texttt{True}|x = 10, y = 8) \approx 0.3366$.

```python
import numpy as np
from icecream import ic

Z = 0
n = 0
n_samples = 10000

while Z < n_samples:

    r = np.random.rand() < 0.25
    lam = 10 if r else 7
    x = np.random.poisson(lam)
    y = np.random.poisson(lam)

    if np.isclose(x, 10, atol=0.01) and np.isclose(y, 8, atol=0.01):
        if r:
            n += 1
        Z += 1
        print('Number of samples:', Z, end='\r')

pQ = n/Z
ic(pQ)
```

Listing 1: Python script for estimating $p(Q = \texttt{True}|x = 10, y = 8)$ using Monte Carlo simulation.