EECS 391

P2_Write_up

Qiwen Luo qxl216

1. Noisy Fuel Guage

a. $P(B = 1) = 0.8 \Rightarrow P(B = 0) = 0.2$

$P(F = 1) = 0.9 \Rightarrow P(F = 0) = 0.1$

$P(G = 1 \mid B = 0, F = 0) = 0.1 \Rightarrow P(G = 0 \mid B = 0, F = 0) = 0.9$

$P(G = 1 \mid B = 1, F = 0) = 0.1 \Rightarrow P(G = 0 \mid B = 1, F = 0) = 0.9$

$P(G = 1 \mid B = 0, F = 1) = 0.2 \Rightarrow P(G = 0 \mid B = 0, F = 1) = 0.8$

$P(G = 1 \mid B = 1, F = 1) = 0.9 \Rightarrow P(G = 0 \mid B = 1, F = 1) = 0.1$

b. Since $P(X_1, \dots, X_n) = \prod_{i=1}^{n} P(X_i|(parents(X_i)))$, $P(B, F, G) = P(B)P(F)P(G|B,F)$

$P(F|G = 0) = \alpha P(F, G = 0)$

$= \alpha \sum_B P(F, B, G = 0)$

$= \alpha(P(F, B = 0, G = 0) + P(F, B = 0, G = 0))$

$= \alpha(< P(F = 0, B = 0, G = 0), P(F = 1, B = 0, G = 0) > + < P(F = 0, B = 1, G = 0), P(F = 1, B = 1, G = 0) >)$

$= \alpha(< P(F = 0)P(B = 0)P(G = 0|B = 0, F = 0), P(F = 1)P(B = 0)P(G = 0|B = 0, F = 1) > + < P(F = 0)P(B = 1)P(G = 0|B = 1, F = 0), P(F = 1)P(B = 1)P(G = 0|B = 1, F = 1) >)$

For simplicity, rewrite $P(F|G= 0) = \alpha(< A, B > + < C, D >)$

Therefore, $P(F = 0|G = 0) = \frac{A+C}{A+B+C+D} = 0.2941$ which is consistent with (c).

c.

```
infer = VariableElimination(model)

# Exercise 1 c)
print("E1 C: P(F = 0| G = 0) = ")
print(infer.query(['F'], evidence={'G': 'ReadEmpty'}))
```

```
+----------+----------+
| F        |   phi(F) |
+==========+==========+
| F(Empty) |   0.2941 |
+----------+----------+
| F(Full)  |   0.7059 |
+----------+----------+
```

P(F = 0|G = 0) = 0.2941

d.

```
# Exercise 1 d)1)
print("E1 D1: P(F) = ")
g_dist = infer.query(['F'])
print(g_dist)

# Exercise 1 d)2)
print("E1 D2: P(F| G = 0) = ")
print(infer.query(['F'], evidence={'G': 'ReadEmpty'}))

# Exercise 2 d)3)
print("E1 D3: P(F | B = 0, G = 0) = ")
print(infer.query(['F'], evidence={'G': 'ReadEmpty', 'B': 'Dead'}))
```

1) P(F) and it is given in the question

```
+----------+----------+
| F        |   phi(F) |
+==========+==========+
| F(Empty) |   0.1000 |
+----------+----------+
| F(Full)  |   0.9000 |
+----------+----------+
```

2) P(F | G = 0) and it is the same as (c)

```
+----------+----------+
| F        |   phi(F) |
+==========+==========+
| F(Empty) |   0.2941 |
+----------+----------+
| F(Full)  |   0.7059 |
+----------+----------+
```

3) P(F | B = 0, G= 0)

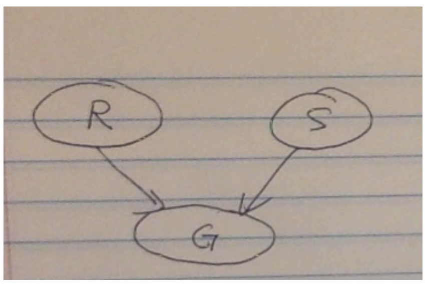
```
+----------+----------+
| F        |   phi(F) |
+==========+==========+
| F(Empty) |   0.1111 |
+----------+----------+
| F(Full)  |   0.8889 |
+----------+----------+
```

e. G is the common evidence(effect) of B and F. From d)2) to d)3), we have additional information about battery. Since the battery is dead, it is more likely that the guage does not work. Namely, the measurement from the guage is mostly incredulous (trustless). Therefore, our belief that the fuel tank is empty goes closer to when we have not made any observation. Namely, the probability moves from 0.2941 (d2) to 0.1111 that is closer to 0.1(d1).

f. The probability pf our belief if P(F|B=0). Since G has not been observed, B and F are independent. Therefore, P(F|B=0) = P(F). Namely, our belief that the fuel tank is empty is 0.1 and the fuel tank is full is 0.9.

2. Wet Grass

a.



b.

P(R = 0) NotRain	P(R = 1) Rain
0.8	0.2

P(S = 0) Off	P(S = 1) LeftOn
0.9	0.1

	P(G = 0) Dry	P(G =1) Wet
R = 0, S = 0	1	0
R = 0, S = 1	0.95	0.05
R = 1, S = 0	0	1
R = 1, S = 1	0	1

c.

```
# Defining the model structure.
model2 = BayesianModel([('R', 'G'), ('S', 'G')])

# Defining individual CPDs.
cpd_r = TabularCPD(variable='R', variable_card=2, values=[[0.8, 0.2]], state_names={'R': ['NotRain', 'Rain']})
cpd_s = TabularCPD(variable='S', variable_card=2, values=[[0.9, 0.1]], state_names={'S':['Off', 'LeftOn']})

cpd_g2 = TabularCPD(variable='G', variable_card=2,
                   values=[[1, 0.95, 0, 0],
                           [0, 0.05, 1, 1]],
                  evidence=['R', 'S'],
                  evidence_card=[2, 2],
                  state_names={'G': ['Dry', 'Wet'], 'R': ['NotRain', 'Rain'], 'S' : ['Off', 'LeftOn']})

# Associating the CPDs with the network
model2.add_cpds(cpd_r, cpd_s, cpd_g2)

# check_model checks for the network structure and CPDs and verifies that the CPDs are correctly
# defined and sum to 1.
model2.check_model()
```

d.

```
infer2 = VariableElimination(model2)

# Exercise 2 d)
# Answer to "Was it due to an overnight rain? Or that last night she forgot to turn off her sprinkler?"
print("E2 D:")
print(infer2.query(variables=['R'], evidence={'G': 'Wet'}))
print(infer2.query(variables=['S'], evidence={'G': 'Wet'}))
print(infer2.map_query(variables=['R','S'], evidence={'G': 'Wet'}))
```

```
E2 D:
Finding Elimination Order: : 100%|██████████| 1/1 [00:00<00:00,
2008.77it/s]
Eliminating: S: 100%|████████████████████| 1/1 [00:00<00:00,
1050.15it/s]
+------------+----------+
| R          |   phi(R) |
+============+==========+
| R(NotRain) |   0.0196 |
+------------+----------+
| R(Rain)    |   0.9804 |
+------------+----------+
Finding Elimination Order: : 100%|██████████| 1/1 [00:00<00:00,
2551.28it/s]
Eliminating: R: 100%|████████████████████| 1/1 [00:00<00:00,
1512.55it/s]
+-----------+----------+
| S         |   phi(S) |
+===========+==========+
| S(Off)    |   0.8824 |
+-----------+----------+
| S(LeftOn) |   0.1176 |
+-----------+----------+
Finding Elimination Order: : : 0it [00:00, ?it/s]
0it [00:00, ?it/s]
```

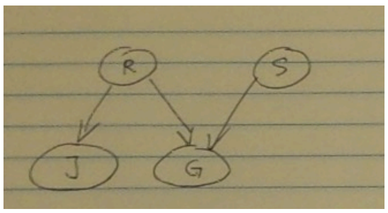
```
{'R': 'Rain', 'S': 'Off'}
```

Given that the grass is wet, P(NotRain|Wet) = 0.0196 and P(Rain|Wet) = 0.9804.

Given that the grass is wet, for the sprinkler, P(Off|Wet) = 0.8824 and P(LeftOn|Wet) = 0.1176.

Therefore, if Tracey sees that the grass is wet, it was most likely due to an overnight rain and the sprinkler was off.

e.



P(R = 0) NotRain	P(R = 1) Rain
0.8	0.2

P(S = 0) Off	P(S = 1) LeftOn
0.9	0.1

	P(G = 0) Dry	P(G =1) Wet
R = 0, S = 0	1	0
R = 0, S = 1	0.95	0.05
R = 1, S = 0	0	1
R = 1, S = 1	0	1

	P(J = 0) JDry	P(J =1) JWet
R = 0	0.85	0.15
R = 1	0	1

```python
# With additional information about John's Grass
# Defining the model structure.
model3 = BayesianModel([('R', 'G'), ('S', 'G'), ('R', 'J')])

# Defining individual CPDs.
# Only one new CPD needed.
cpd_j = TabularCPD(variable='J', variable_card=2,
                   values=[[0.85, 0],
                           [0.15, 1]],
                  evidence=['R'],
                  evidence_card=[2],
                  state_names={'J': ['JDry', 'JWet'], 'R': ['NotRain', 'Rain']})

# Associating the CPDs with the network
model3.add_cpds(cpd_r, cpd_s, cpd_g2, cpd_j)

# check_model checks for the network structure and CPDs and verifies that the CPDs are correctly
# defined and sum to 1.
model3.check_model()
```

f. Since P(X_1, … , X_n) = $\prod_{i=1}^{n} P(X_i|(parents(X_i)))$, P(J,R, G, S) = P(R)P(S)P(J|R)P(G|R,S)

P(S|G = 1, J = 1) = $\alpha P(S, G = 1, J = 1)$

= $\alpha \sum_R P(J = 1, R, G = 1, S)$

= $\alpha(P(J = 1, R = 0, G = 0, S) + P(J = 1, R = 1, G = 1, S))$

= $\alpha(< P(J = 1, R = 0, G = 1, S = 0), P(J = 1, R = 0, G = 1, S = 1) > +< P(J = 1, R = 1, G = 1, S = 0), P(J = 1, R = 1, G = 1, S = 1) >)$

= $\alpha(< P(R = 0)P(S = 0)P(J = 1|R = 0)P(G = 1|R = 0, S = 0), P(R = 0)P(S = 1)P(J = 1|R = 0)P(G = 1|R = 0, S = 1) > +< P(R = 1)P(S = 0)P(J = 1|R = 1)P(G = 1|R = 1, S = 0), P(R = 1)P(S = 1)P(J = 1|R = 1)P(G = 1|R = 1, S = 1) >)$

For simplicity, rewrite P(S|G = 1, J = 1) = $\alpha(< A, B > +< C, D >)$

Therefore, $P(S = 1|G = 1, J = 1) = \frac{A+C}{A+B+C+D}$ = 0.103 which is consistent with the value from toolbox.

```python
# Exercise 2 f)
infer3 = VariableElimination(model3)
print("E2 f: P(S = 1 | G = 1, J = 1) = ")
print(infer3.query(['S'], evidence={'G': 'Wet', 'J': 'JWet'}))
```

```
E2 f: P(S = 1 | G = 1, J = 1) =
Finding Elimination Order: : 100%|██████████| 1/1 [00:00<00:00, 1302.17it/s]
Eliminating: R: 100%|██████████| 1/1 [00:00<00:00, 1253.53it/s]
+-----------+----------+
```

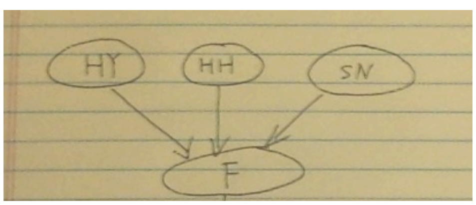
```
| S          |    phi(S) |
+==========+==========+
| S(Off)     |    0.8973 |
+----------+----------+
| S(LeftOn) |    0.1027 |
+----------+----------+
```

g. We have additional information about the condition of John's grass. Since J is only dependent on R, the probability that there was overnight rain increases. Since G is the common evidence(effect) of R and S and G has been observed, the influences flows from R to S. Therefore, S decreases. Namely, Tracery's belief that she left the sprinkler on decreases.

3. Extra

Chris in on his way to Quad and he sees a girl fainted. He wonders: Was the faint due to hypoglycemia? Or she was hit on the head? Or she heard an extremely sad news?

The noisy-OR model of the Bayes net that models this scenario is shown below. Using the following variables: HY = the girl has hypoglycemia; HH = the girl's head was hit; SN = the girl heard an extremely sad news; F = the girl falls in a faint.



Assumes that the parent causes of effect F contribute independently. Therefore, the probability that none of them caused effect F is simply the product of the probabilities that each one did not cause F. The probability that any of them caused F is just one minus the above.

P(HY = 0) NoHypoglycemia	P(HY = 1) Hypoglycemia
0.8	0.2

P(HH = 0) NoHeadHit	P(HH = 1) HeadHit

0.95	0.05

P(SN = 0) NoSadNews	P(SN = 1) SadNews
0.9	0.1

Conditional probability table for P(F|HY, HH, SN) as calculated from the noisy-OR model.

HY	HH	SN	P(F)	P(¬F)
0	0	0	0	1
0	0	1	0.2	0.8
0	1	0	0.9	0.1
0	1	1	1-(1-0.2)(1-0.9)= 0.92	0.08
1	0	0	0.7	0.3
1	0	1	1-(1-0.7)(1-0.2)= 0.76	0.24
1	1	0	1-(1-0.7)(1-0.9)= 0.97	0.03
1	1	1	1-(1-0.2)(1-0.9)(1-0.7))= 0.976	0.024