CS267A: Homework #5

Peter Racioppo (103953689)

May 18, 2020

Acknowledgements: If you have discussed with other students in the class regarding this homework, please acknowledge their names. See the syllabus for detailed policies about collaboration and academic honesty.

Problem 1

Topic: Learning in ProbLog.
The classic "earthquake, burglary" example is back. Please use ProbLog to model it. Consider we are the police station, which means, we are able to collect data regarding whether there is burglary/earthquake and whether Mary/John calls, but we don't know whether the alarms in their home sound or not. Please update the conditional probabilities (i.e. parameters) in your ProgLog program through Bayesian learning after observing the following examples.

- *No Burglary, No earthquake, Mary did not call, John did not call*
- *Burglary, No earthquake, John called, Mary did not call*
- *No Burglary, No earthquake, Mary did not call, John did not call*
- *No Burglary, no earthquake, John did not call, Mary called*
- *Burglary, No earthquake, John called, Mary called*
- *Burglary, earthquake, John called, Mary called*
- *No Burglary, No earthquake, Mary did not call, John did not call*
- *No Burglary, No earthquake, Mary called, John did not call*

Solution:
1. P(burglary $\vee$ earthquake | John calls $\wedge$ Mary calls) = 1.0000.

2. P(earthquake | $\neg$ burglary $\wedge$ Mary calls) = 0.0024.

I used the 'Learning' feature of ProbLog to update the initial weights with new evidence:

```
0.001::burglary.
0.002::earthquake.

t(0.95)::p_alarm1.
t(0.94)::p_alarm2.
t(0.29)::p_alarm3.
t(0.001)::p_alarm4.

t(0.90)::p_john1.
t(0.05)::p_john2.
```

```
t(0.70)::p_mary1.
t(0.01)::p_mary2.

alarm :- burglary, earthquake, p_alarm1.
alarm :- burglary, \+earthquake, p_alarm2.
alarm :- \+burglary, earthquake, p_alarm3.
alarm :- \+burglary, \+earthquake, p_alarm4.

john :- alarm, p_john1.
john :- \+alarm, p_john2.
mary :- alarm, p_mary1.
mary :- \+alarm, p_mary2.

%%% The data:
%% No Burglary, No earthquake, Mary did not call, John did not call
%% Burglary, No earthquake, John called, Mary did not call
%% No Burglary, No earthquake, Mary did not call, John did not call
%% No Burglary, no earthquake, John did not call, Mary called
%% Burglary, No earthquake, John called, Mary called
%% Burglary, earthquake, John called, Mary called
%% No Burglary, No earthquake, Mary did not call, John did not call
%% No Burglary, No earthquake, Mary called, John did not call

evidence(burglary,false).
evidence(earthquake,false).
evidence(mary,false).
evidence(john,false).
---
evidence(burglary,true).
evidence(earthquake,false).
evidence(john,true).
evidence(mary,false).
---
evidence(burglary,false).
evidence(earthquake,false).
evidence(mary,false).
evidence(mary,false).
---
evidence(burglary,false).
evidence(earthquake,false).
evidence(john,false).
evidence(mary,true).
---
evidence(burglary,true).
evidence(earthquake,false).
evidence(john,true).
evidence(mary,true).
---
evidence(burglary,true).
evidence(earthquake,true).
evidence(john,true).
evidence(mary,true).
---
evidence(burglary,false).
```

```
evidence(earthquake,false).
evidence(mary,false).
evidence(john,false).
```

```
evidence(burglary,false).
evidence(earthquake,false).
evidence(mary,true).
evidence(john,false).
```

ProbLog output the following model. I used this model to answer the two queries:

```
0.001::burglary.
0.002::earthquake.
0.999999999430665::p_alarm1.
0.999999999999999::p_alarm2.
0.29::p_alarm3.
0.0::p_alarm4.
1.0::p_john1.
0.0::p_john2.
0.666666666666667::p_mary1.
0.4::p_mary2.
alarm :- burglary, earthquake, p_alarm1.
alarm :- burglary, \+earthquake, p_alarm2.
alarm :- \+burglary, earthquake, p_alarm3.
alarm :- \+burglary, \+earthquake, p_alarm4.
john :- alarm, p_john1.
john :- \+alarm, p_john2.
mary :- alarm, p_mary1.
mary :- \+alarm, p_mary2.

burglary_or_earthquake :- \+ (\+burglary, \+earthquake).
```

For the first query, I added the following code:

```
%% 1. What is the probability that there is a burglary or earthquake
%% given both John and Mary call?

evidence(john,true).
evidence(mary,true).

query(burglary_or_earthquake).
```

For the second query, I added the following code:

```
%% 2. What is the probability that there is an earthquake
%% given no burglary and Mary calls?

evidence(burglary,false).
evidence(mary,true).

query(earthquake).
```

Problem 2

Consider a collection of deterministic databases with a single table called T having a single column, and a domain of A, B, and C. Then we define a probabilistic database by assigning a probability to each of the above deterministic databases:
Pr(ω_1) = 0.16 ; Pr(ω_2) = 0.16; Pr(ω_3) = 0.24 ;Pr(ω_4) = 0.04; Pr(ω_5) = 0.04; Pr(ω_6) = 0.24; Pr(ω_7) = 0.06; Pr(ω_8) = 0.06
1. What is $Pr(\exists x.T(x))$ for the above probabilistic database?
2. Fill in the probabilities of the following tuple-independent database so that this tuple-independent database describes the same probabilistic database as above:

Solution:
1. Pr$(\exists x.T(x)) = 1-$ Pr $(\omega_7) = 1 - 0.06 = 0.94.$

2. Pr(A) = Pr(ω_1) + Pr(ω_2) + Pr(ω_4) + Pr(ω_5) = 0.16 + 0.16 + 0.04 + 0.04 = 0.40.
Pr(B) = Pr(ω_1) + Pr(ω_2) + Pr(ω_3) + Pr(ω_6) = 0.16 + 0.16 + 0.24 + 0.24 = 0.80.
Pr(C) = Pr(ω_1) + Pr(ω_3) + Pr(ω_4) + Pr(ω_7) = 0.16 + 0.24 + 0.04 + 0.06 = 0.50.

As a check, note that:
Pr(A) $\times$ Pr(B) $\times$ Pr(C) = 0.40 $\times$ 0.80 $\times$ 0.50 = 0.16 = Pr(A, B, C)
Pr(A) $\times$ Pr(B) = 0.40 $\times$ 0.80 = 0.32 = 0.16 + 0.16 = Pr($A, B, \neg C$) + Pr(A, B, C)
Pr(A) $\times$ Pr(C) = 0.40 $\times$ 0.50 = 0.20 = 0.04 + 0.16 = Pr($A, \neg B, C$) + Pr(A, B, C)
Pr(B) $\times$ Pr(C) = 0.80 $\times$ 0.50 = 0.40 = 0.24 + 0.16 = Pr($\neg A, B, C$) + Pr(A, B, C)
etc...

Problem 3

Topics: Probabilistic Databases
Consider the following probabilistic database, which is similar to the one from class:
Let $Pr(\omega_1) = 1/8, Pr(\omega_2) = 1/4, Pr(\omega_3) = 3/8$, and $Pr(\omega_4) = 1/4$. Assume that the domain of the "Name" column is $\{Alice, Carol\}$ and that the domain of the "Assoc." column is $\{Pixar, UPenn, Brown, INRIA\}$. Under the domain-closure assumption, answer the following:
1. Compute each of the following queries on the database:
(a) Pr(T(Alice,Pixar)).
(b) Pr($\exists x$. T(Alice,x)).
(c) Pr($\exists x, y$. T(x, y)).
(d) Pr(x. T(x,Brown)).
2. Is this probabilistic database tuple-independent? Why or why not?

Solution:
(1a.) Pr(T(Alice,Pixar)) = Pr(ω_1) + Pr(ω_2) = 1/8 + 1/4 = 3/8.
(1b.) Pr($\exists x$. T(Alice,x)) = Pr(ω_1) + Pr(ω_2) + Pr(ω_3) = 1/8 + 1/4 + 3/8 = 3/4.
(1c.) Pr($\exists x, y$. T(x, y)) = Pr(ω_1) + Pr(ω_2) + Pr(ω_3) = 1/8 + 1/4 + 3/8 = 3/4.
(1d.) Pr(x. T(x,Brown)) = Pr(T(Alice,Brown) $\wedge$ T(Carol,Brown)) = 0.

(2.) In order to have tuple-independence, we must have that:
Pr(T(Alice,Pixar) $\wedge$ T(Carol,UPenn)) = Pr(T(Alice,Pixar)) $\times$ Pr(T(Carol,UPenn)) and
Pr(T(Alice,Pixar) $\wedge$ T(Carol,INRIA)) = Pr(T(Alice,Pixar)) $\times$ Pr(T(Carol,INRIA)).
But Pr(T(Alice,Pixar)) = Pr(ω_1) + Pr(ω_2) = 1/8 + 1/4 = 3/8, Pr(T(Carol,UPenn)) = Pr(ω_1) = 1/8, and
Pr(T(Carol,INRIA)) = Pr(ω_2) = 1/4. Thus,
Pr(T(Alice,Pixar)) $\times$ Pr(T(Carol,UPenn)) = $(3/8)(1/8) = 3/64 \neq$ Pr(T(Alice,Pixar)$\wedge$ T(Carol,UPenn)) = 1/8,
Pr(T(Alice,Pixar)) $\times$ Pr(T(Carol,INRIA)) = $(3/8)(1/4) = 3/32 \neq$ Pr(T(Alice,Pixar)$\wedge$ T(Carol,INRIA)) = 1/4.
So this probabilistic database is not tuple-independent.

Problem 4

Topic: Probabilistic Database, ProbLog
Consider the following tuple-independent probabilistic database.

Part A
A tuple-independent probabilistic database describes a probability distribution on a collection of classical deterministic databases (the possible worlds). For just the table T_H , give (1) the set of all possible world databases of this table; (2) the probability of each such deterministic database.

Solution:
Let A = Alice, B = Bob, C = Charlie.
The set of all possible world databases of T_H is {{A,B,C},{A,B},{A,C},{B,C},{A},{B},{C},{}}. There are $2^3 = 8$ possible worlds. The probabilities of each world are:
Pr($\{A, B, C\}$) = Pr(A, B, C) = 0.7 $\times$ 0.4 $\times$ 0.9 = 0.252.
Pr($\{A, B\}$) = Pr($A, B, \neg C$) = 0.7 $\times$ 0.4 $\times$ 0.1 = 0.028.
Pr($\{A, C\}$) = Pr($A, \neg B, C$) = 0.7 $\times$ 0.6 $\times$ 0.9 = 0.378.
Pr($\{B, C\}$) = Pr($\neg A, B, C$) = 0.3 $\times$ 0.4 $\times$ 0.9 = 0.108.
Pr($\{A\}$) = Pr($A, \neg B, \neg C$) = 0.7 $\times$ 0.6 $\times$ 0.1 = 0.042.
Pr($\{B\}$) = Pr($\neg A, B, \neg C$) = 0.3 $\times$ 0.4 $\times$ 0.1 = 0.012.
Pr($\{C\}$) = Pr($\neg A, \neg B, C$) = 0.3 $\times$ 0.6 $\times$ 0.9 = 0.162.
Pr($\{\}$) = Pr($\neg A, \neg B, \neg C$) = 0.3 $\times$ 0.6 $\times$ 0.1 = 0.018.

Part B
Compute each of the following queries for the above probabilistic database without enumerating all possible worlds, and instead exploiting independence:
1. Pr($\exists x.H(x)$)
2. Pr($\exists x.H(x) \wedge E(x,$ Artificial Intelligence(AI)$)$)
3. Pr($\exists x \exists y.H(x) \wedge E(x, y)$)

Problem 4 continued on next page. . .

Solution:
1. $\Pr(\exists x.H(x))$
$= 1- \Pr(\forall x.\neg H(x))$
$= 1 - \prod_i (1 - H(x))$
$= 1 - (1 - H(A))(1 - H(B))(1 - H(C))$
$= 1 - (1 - 0.7)(1 - 0.4)(1 - 0.9)$
$= 1 - 0.3 \times 0.6 \times 0.1 = 1 - 0.018 = 0.982.$

2. $\Pr(\exists x.H(x) \wedge E(x, \text{AI}))$
$= 1- \Pr(\forall x.\neg H(x) \vee \neg E(x, \text{AI}))$
$= 1 - \prod_i \Pr(\neg H(i) \vee \neg E(i, \text{AI}))$ (by tuple independence)
$= 1 - \prod_i (1 - \Pr(H(i) \wedge E(i, \text{AI})))$
$= 1 - \prod_i (1 - \Pr(H(i))\Pr(E(i, \text{AI})))$
$= 1 - (1 - \Pr(H(A))\Pr(E(A, \text{AI})))(1 - \Pr(H(B))\Pr(E(B, \text{AI})))(1 - \Pr(H(C))\Pr(E(C, \text{AI})))$
$= 1 - (1 - 0.7 \times 0.7)(1 - 0.4 \times 0.3)(1 - 0.9 \times 0)$
$= 1 - (1 - 0.49)(1 - 0.12) = 1 - 0.51 \times 0.88 = 1 - 0.45$
$= 0.55.$

3. $\Pr(\exists x \exists y.H(x) \wedge E(x, y))$
$= 1- \Pr(\forall x \forall y.\neg H(x) \vee \neg E(x, y))$
$= 1 - \prod_i (\Pr(\forall y.\neg H(i) \vee \neg E(i, y)))$
$= 1 - \prod_i (\Pr(\neg H(i) \vee \forall y.\neg E(i, y)))$
$= 1 - \prod_i (1- \Pr(H(i) \wedge \exists y.E(i, y)))$
$= 1 - \prod_i (1- \Pr(H(i)) \times \Pr(\exists y.E(i, y)))$
$= 1 - \prod_i (1- \Pr(H(i)) \times (1 - \Pr(\forall y.\neg E(i, y))))$
$= 1 - \prod_i (1- \Pr(H(i)) \times (1 - \prod_j (1 - \Pr(E(i, j)))))$
$= 1 - (1- \Pr(H(A)) \times (1 - \prod_j (1 - \Pr(E(A, j)))))(1- \Pr(H(B)) \times (1 - \prod_j (1 - \Pr(E(B, j)))))(1- \Pr(H(C)) \times (1 - \prod_j (1 - \Pr(E(C, j)))))$
$= 1 - (1- \Pr(H(A)) \times (1 - (1 - \Pr(E(A, AI)))(1 - \Pr(E(A, PL)))))(1- \Pr(H(B))$
$\times (1 - (1 - \Pr(E(B, AI)))(1 - \Pr(E(B, PL)))))(1- \Pr(H(C))$
$\times (1 - (1 - \Pr(E(C, AI)))(1 - \Pr(E(C, PL)))))$
$= 1 - (1 - 0.7 \times (1 - (1 - 0.7)(1 - 0.4)))(1 - 0.4 \times (1 - (1 - 0.3)(1 - 0)))$
$= 1 - (1 - 0.7 \times (1 - 0.3 \times 0.6))(1 - 0.4 \times (1 - 0.7))$
$= 1 - 0.426 \times 0.88$
$= 0.625$

Part C

Implement the queries above in ProbLog, and make sure they give you the same numbers as before. Include your code.

Solution:
(1.) 0.982

```
0.70::alice.
0.40::bob.
0.90::charlie.

clause1 :- \+(\+alice,\+bob,\+charlie).

query(clause1).
```

Solution:
(2.) 0.5512

```
0.70::alice.
0.40::bob.
0.90::charlie.

0.70::p_alice_ai.
0.40::p_alice_pl.
0.30::p_bob_ai.

or1 :- alice, p_alice_ai.
or2 :- bob, p_bob_ai.
clause2 :- \+(\+or1,\+or2).

query(clause2).
```

Solution:
(3.) 0.62512

```
0.70::alice.
0.40::bob.
0.90::charlie.

0.70::p_alice_ai.
0.40::p_alice_pl.
0.30::p_bob_ai.

or1 :- alice, p_alice_ai.
or2 :- alice, p_alice_pl.
or3 :- bob, p_bob_ai.

clause3 :- \+(\+or1,\+or2,\+or3).

query(clause3).
```