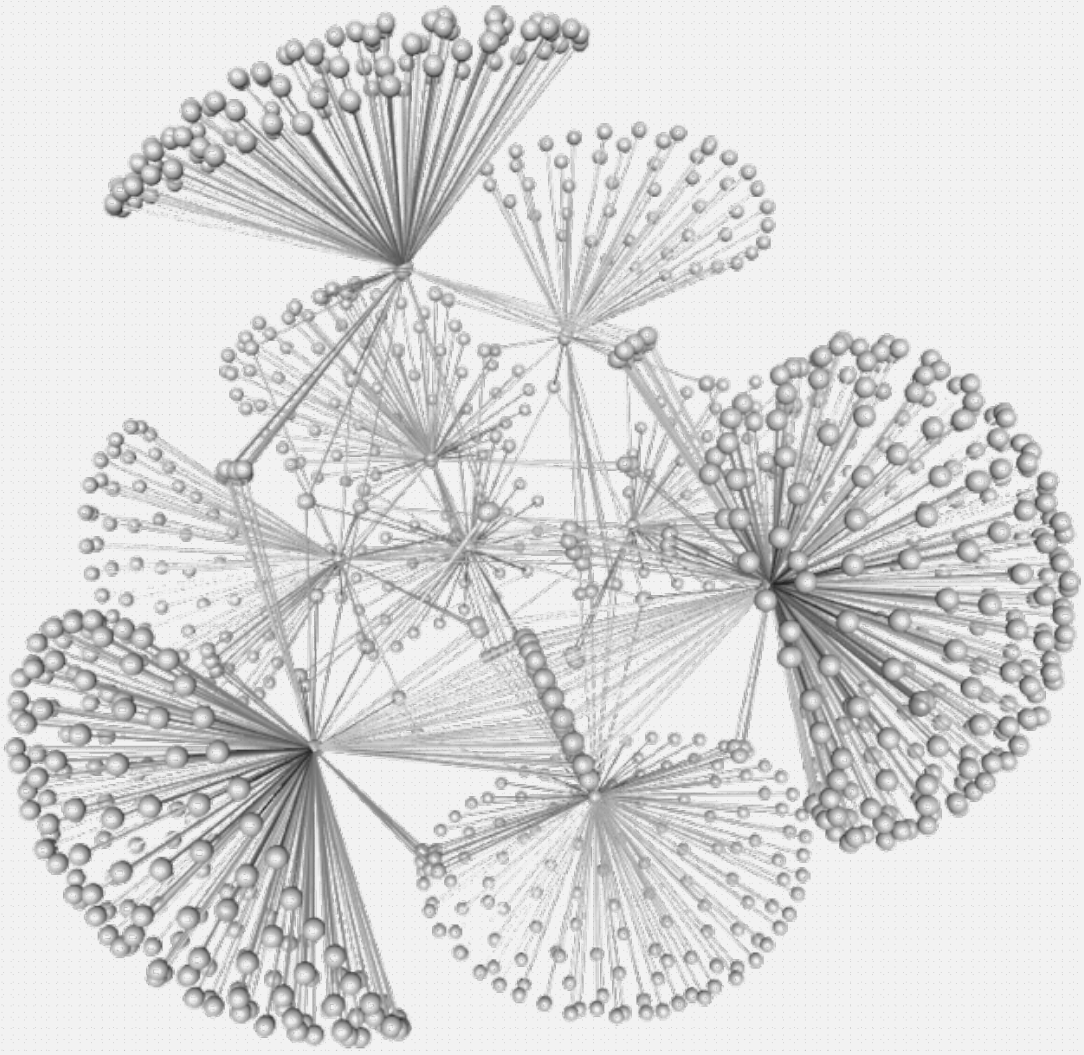
Probabilistic Programming

Michiel Janssen & Bruno Vandekerkhove



Academisch Jaar 2020

H05N0A: Capita Selecta : Artificial Intelligence

Contents

Below's our solution for the given challenges. The questions in each section of the original assignment are answered in a section having the same title.

```
person(a).
person(b).
person(c).
0.2::stress(X) :- person(X).
0.1::friends(X,Y) :- person(X), person(Y).
0.3::smokes(X) :- stress(X).
0.4::smokes(X) :- friends(X,Y), smokes(Y).
query(smokes(a)).
```

Code snippet 1: PROBLOG program used throughout the first two chapters of the report.

Probabilistic Inference Using Weighted Model Counting

SRL to CNF

First the program is grounded. This is a matter of collecting all atoms involved in all proofs of the query.

```
0.2::stress(a).
0.2::stress(b).
0.2::stress(c).

0.1::friends(a,a).
0.1::friends(a,b).
0.1::friends(a,c).

0.1::friends(b,a).
0.1::friends(b,b).
0.1::friends(b,c).

0.1::friends(c,a).
0.1::friends(c,b).
0.1::friends(c,c).

0.3::smokes(a) :- stress(a).
0.3::smokes(b) :- stress(b).
0.3::smokes(c) :- stress(c).

0.4::smokes(a) :- friends(a,a), smokes(a).
0.4::smokes(a) :- friends(a,b), smokes(b).
0.4::smokes(a) :- friends(a,c), smokes(c).
0.4::smokes(b) :- friends(b,a), smokes(a).
0.4::smokes(b) :- friends(b,b), smokes(b).
0.4::smokes(b) :- friends(b,c), smokes(c).

0.4::smokes(c) :- friends(c,a), smokes(a).
0.4::smokes(c) :- friends(c,b), smokes(b).
0.4::smokes(c) :- friends(c,c), smokes(c).
```

Code snippet 2: Relevant ground program.

The proofs of the query make for a trie as shown in figure 1, where colourings indicate the presence of cycles. Any proof involving an atom `friends(X,X)` or `friends(Y,a)` (with $Y \in \{b, c\}$) is non-minimal and doesn't affect the final probability. These atoms are disregarded. For the remaining cycles (involving `friends(b,c)` and `friends(c,b)`) auxiliary variables can be used to obtain a cycle-free program :

```
0.2::stress(a).
0.2::stress(b).
0.2::stress(c).

```

```
0.1::friends(a,b).
0.1::friends(a,c).
0.1::friends(b,c).
0.1::friends(c,b).

0.3::p(a).
0.3::p(b).
0.3::p(c).

0.4::p(a,b).
0.4::p(a,c).
0.4::p(b,c).
0.4::p(c,b).

smokes(a) :- stress(a), p(a).
smokes(b) :- stress(b), p(b).
smokes(c) :- stress(c), p(c).

smokes(a) :-
    friends(a,b), smokes(b), p(a,b).
smokes(a) :-
    friends(a,c), smokes(c), p(a,c).
smokes(b) :-
    friends(b,c), stress(c), p(c), p(b,c).
smokes(c) :-
    friends(c,b), stress(b), p(b), p(c,b).

query(smokes(a)).
```

Code snippet 3: Relevant ground program without cycles.

The above logic program is equivalent to the following propositional formula :

$$
\begin{aligned}
& smokes(a) \leftrightarrow && stress(a) \\
& && \vee\ (friends(a,b) \wedge smokes(b)) \\
& && \vee\ (friends(a,c) \wedge smokes(c)) \\
& \wedge\ (smokes(b) \leftrightarrow && p(b) \vee (friends(b,c) \wedge p(c))) \\
& \wedge\ (smokes(c) \leftrightarrow && p(c) \vee (friends(c,b) \wedge p(b))) \\
& \wedge\ (p(b) \leftrightarrow && stress(b)) \\
& \wedge\ (p(c) \leftrightarrow && stress(c))
\end{aligned}
$$

Which gives the following `CNF` :

$$
\begin{aligned}
& (\neg smokes(a) \vee stress(a) \vee friends(a,b) \vee friends(a,c)) \\
& \wedge (\neg smokes(a) \vee stress(a) \vee friends(a,b) \vee smokes(c)) \\
& \wedge (\neg smokes(a) \vee stress(a) \vee smokes(b) \vee friends(a,c)) \\
& \wedge\ (\neg smokes(a) \vee stress(a) \vee smokes(b) \vee smokes(c)) \\
& \wedge\ (\neg stress(a) \vee smokes(a)) \\
& \wedge\ (\neg friends(a,b) \vee \neg smokes(b) \vee smokes(a)) \\
& \wedge\ (\neg friends(a,c) \vee \neg smokes(c) \vee smokes(a)) \\
& \wedge\ (\neg smokes(b) \vee p(b) \vee friends(b,c)) \\
& \wedge\ (\neg smokes(b) \vee p(b) \vee p(c)) \\
& \wedge\ (\neg p(b) \vee smokes(b)) \\
& \wedge\ (\neg friends(b,c) \vee \neg p(c) \vee smokes(b)) \\
& \wedge\ (\neg smokes(c) \vee p(c) \vee friends(c,b)) \\
& \wedge\ (\neg smokes(c) \vee p(c) \vee p(b)) \\
& \wedge\ (\neg p(c) \vee smokes(c)) \\
& \wedge\ (\neg friends(c,b) \vee \neg p(b) \vee smokes(c)) \\
& \wedge\ (\neg p(b) \vee stress(b)) \\
& \wedge\ (\neg stress(b) \vee p(b)) \\
& \wedge\ (\neg p(c) \vee stress(c)) \\
& \wedge\ (\neg stress(c) \vee p(c))
\end{aligned}
$$

The probabilistic literals `CNF` are assigned weights (derived literals have a weight of 1) :

Atom	Weight
stress(a)	0.2
¬stress(a)	0.8
stress(b)	0.2
¬stress(b)	0.8
stress(c)	0.2
¬stress(c)	0.8
friends(a,b)	0.1
¬friends(a,b)	0.9
friends(a,c)	0.1
¬friends(a,c)	0.9
friends(b,c)	0.1
¬friends(b,c)	0.9
friends(c,b)	0.1
¬friends(c,b)	0.9

SRL to PGM

PGM to CNF

Weighted Model Counting

Lifted Inference

Parameter Learning

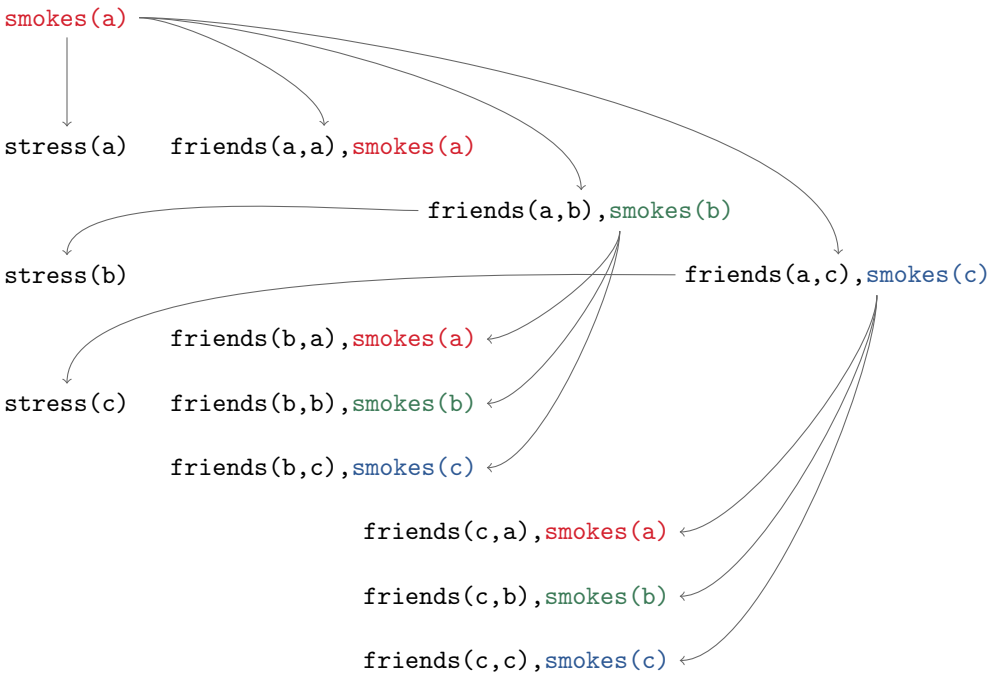


Figure 1: SLG-tree produced while turning the ground program into a boolean formula. Coloured atoms indicate the presence of cycles.