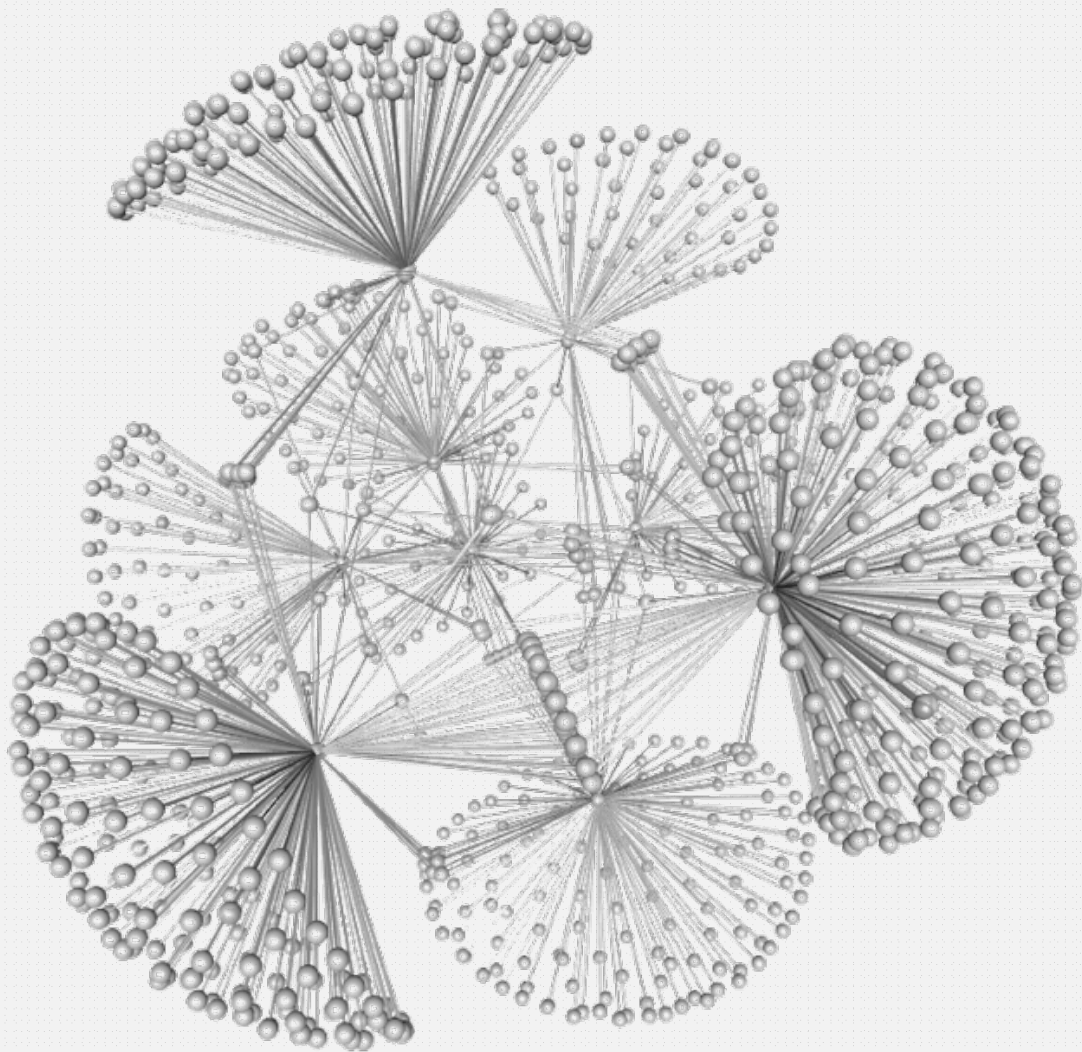


Probabilistic Programming

Michiel Janssen & Bruno Vandekerkhove



ACADEMISCH JAAR 2020

H05N0A: CAPITA SELECTA : ARTIFICIAL
INTELLIGENCE

Contents

Probabilistic Inference Using Weighted Model Counting	1
1.1 SRL to CNF	1
1.2 SRL to PGM	3
1.3 PGM to CNF	3
1.4 Weighted Model Counting	3
Lifted Inference	3
Parameter Learning	3

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

```
1 person(a).
2 person(b).
3 person(c).
4 0.2::stress(X) :- person(X).
5 0.1::friends(X,Y) :- person(X), person(Y).
6 0.3::smokes(X) :- stress(X).
7 0.4::smokes(X) :- friends(X,Y), smokes(Y).
8 query(smokes(a)).
```

Code snippet 1: PROBLOG program used throughout.

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.

```
1 0.2::stress(a).
2 0.2::stress(b).
3 0.2::stress(c).
4
5 0.1::friends(a,a).
6 0.1::friends(a,b).
7 0.1::friends(a,c).
8
9 0.1::friends(b,a).
10 0.1::friends(b,b).
11 0.1::friends(b,c).
12
13 0.1::friends(c,a).
14 0.1::friends(c,b).
15 0.1::friends(c,c).
16
17 0.3::smokes(a) :- stress(a).
18 0.3::smokes(b) :- stress(b).
19 0.3::smokes(c) :- stress(c).
20
21 0.4::smokes(a) :- friends(a,a), smokes(a).
22 0.4::smokes(a) :- friends(a,b), smokes(b).
23 0.4::smokes(a) :- friends(a,c), smokes(c).
```

```

24 0.4::smokes(b) :- friends(b,a), smokes(a).
25 0.4::smokes(b) :- friends(b,b), smokes(b).
26 0.4::smokes(b) :- friends(b,c), smokes(c).
27
28 0.4::smokes(c) :- friends(c,a), smokes(a).
29 0.4::smokes(c) :- friends(c,b), smokes(b).
30 0.4::smokes(c) :- friends(c,c), smokes(c).

```

Code snippet 2: Relevant ground program.

The proofs of the query make for the following (nested) trie, where colourings indicate the presence of loops :

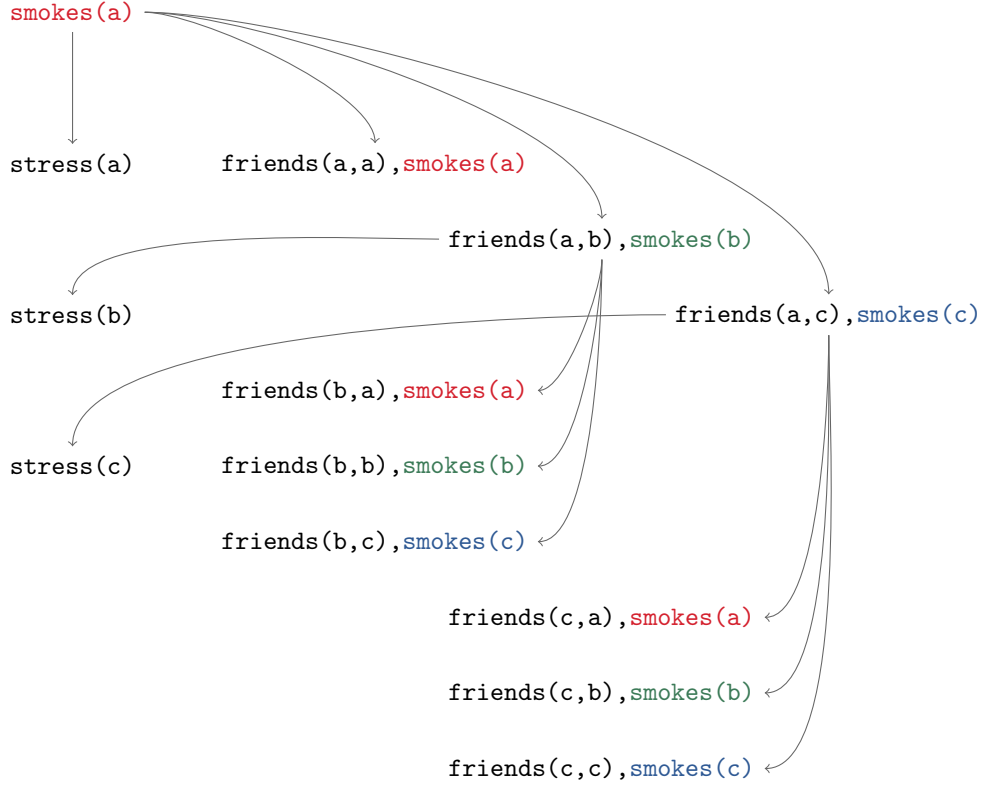


Figure 1: SLG-tree produced while turning the ground program into a boolean formula. Coloured atoms refer to loops.

The loops can be removed while retaining the semantics through the use of auxiliary variables. For example :

```

1 0.2::stress(a).
2 0.2::stress(b).
3 0.2::stress(c).
4
5 0.1::friends(a,b).
6 0.1::friends(a,c).
7 0.1::friends(b,c).
8 0.1::friends(c,b).
9
10 0.3::p(b) :- stress(b).
11 0.3::p(c) :- stress(c).
12
13 0.3::smokes(a) :- stress(a).
14 smokes(b) :- p(b).
15 smokes(c) :- p(c).
16
17 0.4::smokes(a) :- friends(a,b), smokes(b).
18 0.4::smokes(a) :- friends(a,c), smokes(c).

```

```

19  0.4::smokes(b) :- friends(b,c), p(c).
20  0.4::smokes(c) :- friends(c,b), p(b).
21
22  query(smokes(a)).

```

Code snippet 3: Relevant ground program without cycles.

This is equivalent to the given propositional formula :

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

Which, in CNF form, gives :

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

As a weighted CNF we end up with the following weights :

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

Table 1: Weights for the CNF.

SRL to PGM

PGM to CNF

Weighted Model Counting

Lifted Inference

Parameter Learning