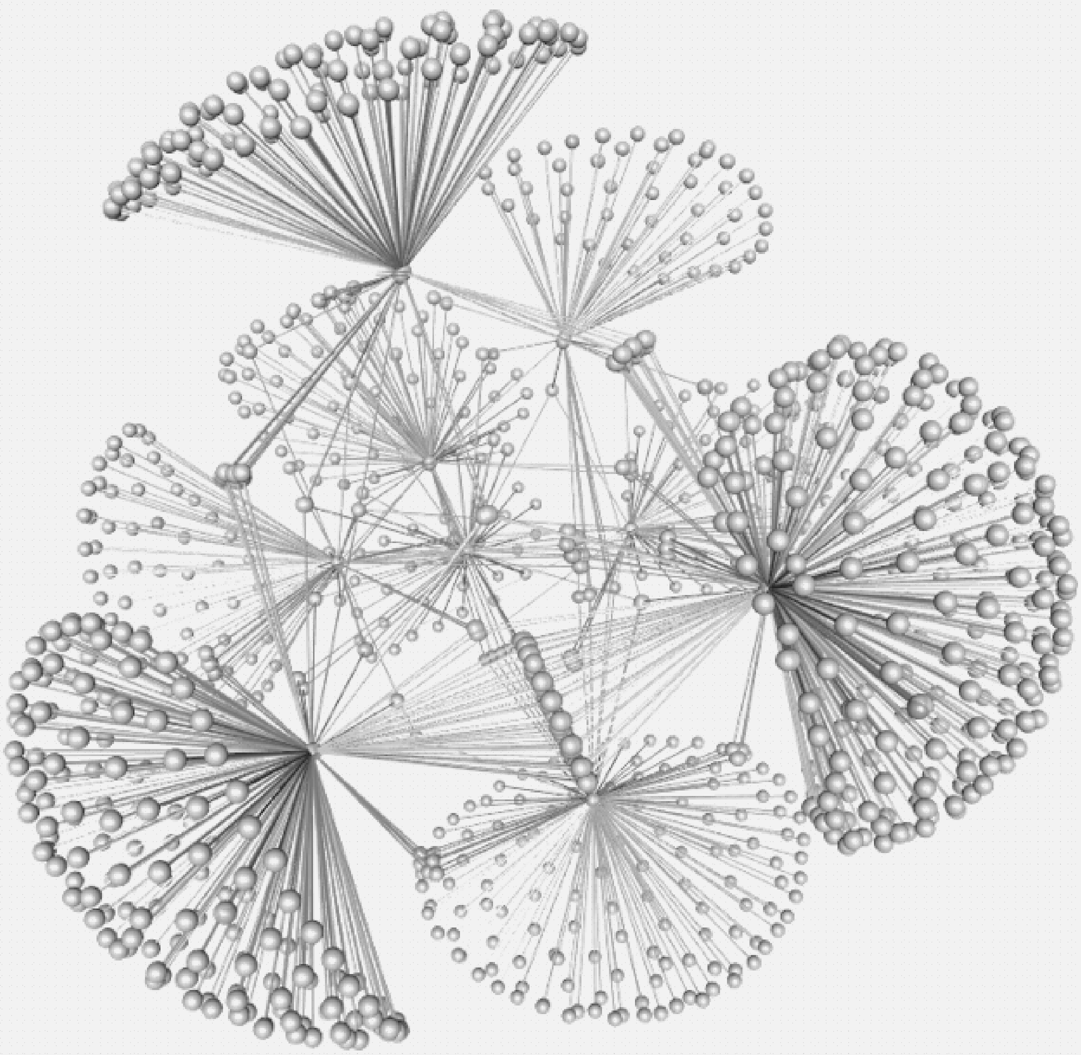
Probabilistic Programming

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 without intensional probabilistic facts :

```
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 :

$$
\begin{aligned}
&(smokes(a) \leftrightarrow \quad (stress(a) \land p(a)) \\
&\quad \lor (friends(a,b) \land smokes(b) \land p(a,b)) \\
&\quad \lor (friends(a,c) \land smokes(c) \land p(a,c))) \\
&\qquad\qquad \land \\
&(smokes(b) \leftrightarrow \quad (stress(b) \land p(b)) \\
&\lor (friends(b,c) \land stress(c) \land p(c) \land p(b,c))) \\
&\qquad\qquad \land \\
&(smokes(c) \leftrightarrow \quad (stress(c) \land p(c)) \\
&\lor (friends(c,b) \land stress(b) \land p(b) \land p(c,b)))
\end{aligned}
$$

Which corresponds to the following `CNF` :

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

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

Literal	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
p(a)	0.3
¬p(a)	0.7
p(b)	0.3
¬p(b)	0.7
p(c)	0.3
¬p(c)	0.7
p(a,b)	0.4
¬p(a,b)	0.6
p(a,c)	0.4
¬p(a,c)	0.6
p(b,c)	0.4
¬p(b,c)	0.6
p(c,b)	0.4
¬p(c,b)	0.6

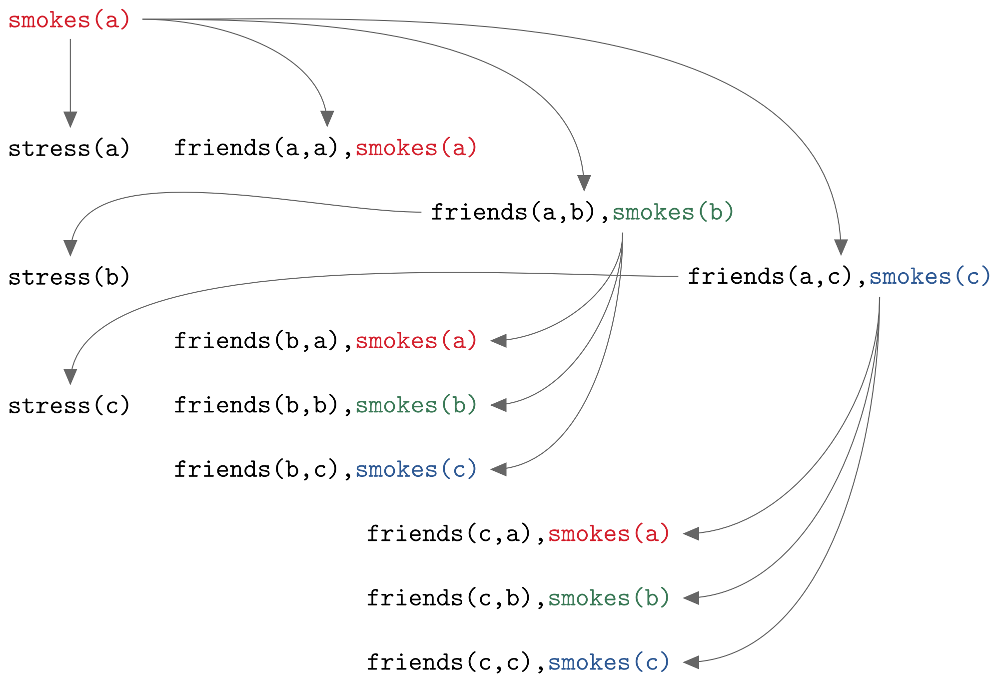


Figure 1: Trie representing proofs of the query. Coloured atoms indicate the presence of cycles.

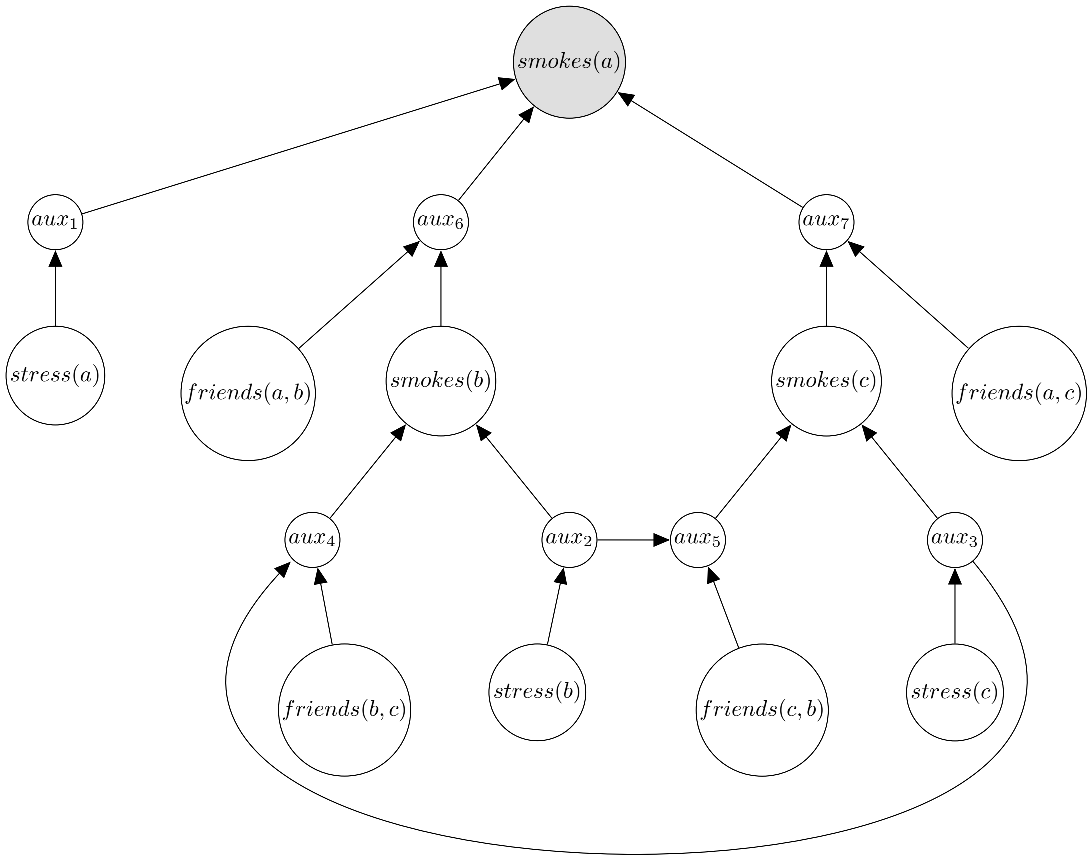


Figure 2: Bayesian network corresponding to the ground acyclic program.

SRL to PGM

A Bayesian network is shown in figure 2. The conditional probability tables (CPTs) for the nodes are given below. Note that every table represents multiple identical tables in the network (like those of the $stress(a)$, $stress(b)$ and $stress(c)$ nodes for example).

	$stress(\{a,b,c\})$
$\top$	0.2
$\bot$	0.8

	$friends(\{a,b,c\},\{b,c\})$
$\top$	0.1
$\bot$	0.9

$stress(\{a,b,c\})$	$aux_{\{1,2,3\}}$ $\top$	$aux_{\{1,2,3\}}$ $\bot$
$\top$	0.3	0.7
$\bot$	0.0	1.0

$friends(\{b,c\},\{c,b\})$	$aux_{2,3}$	$aux_{4,5}$ $\top$	$aux_{4,5}$ $\bot$
$\top$	$\top$	0.4	0.6
$\top$	$\bot$	0.0	1.0
$\bot$	$\top$	0.0	1.0
$\bot$	$\bot$	0.0	1.0

$friends(\{a\},\{b,c\})$	$smokes(\{b,c\})$	$aux_{6,7}$ $\top$	$aux_{6,7}$ $\bot$
$\top$	$\top$	0.4	0.6
$\top$	$\bot$	0.0	1.0
$\bot$	$\top$	0.0	1.0
$\bot$	$\bot$	0.0	1.0

$aux_{4,5}$	$aux_{2,3}$	$smokes(\{b,c\})$ $\top$	$smokes(\{b,c\})$ $\bot$
$\top$	$\top$	1.0	0.0
$\top$	$\bot$	0.0	1.0
$\bot$	$\top$	0.0	1.0
$\bot$	$\bot$	0.0	1.0

aux_1	$smokes(b)$	$smokes(c)$	$smokes(a)$ $\top$	$smokes(a)$ $\bot$
$\top$	$\top$	$\top$	1.0	0.0
$\top$	$\top$	$\bot$	0.0	1.0
$\top$	$\bot$	$\top$	0.0	1.0
$\top$	$\bot$	$\bot$	0.0	1.0
$\bot$	$\top$	$\top$	0.0	1.0
$\bot$	$\top$	$\bot$	0.0	1.0
$\bot$	$\bot$	$\top$	0.0	1.0
$\bot$	$\bot$	$\bot$	0.0	1.0

PGM to CNF

The Bayesian network can be encoded as a logical formula. Encodings ENC1 and ENC2 discussed [1] both make use of the same indicator variables such as $\lambda_{stress(a)=true}$ and $\lambda_{stress(a)=false}$ (which introduces redundancy considering the fact that all network variables are boolean).

In `ENC1` each row in each CPT is encoded by a parameter clause.
In `ENC2` an order is assumed over each variable's values. Then, each row of each CPT is encoded by an equivalence.

For both of these a Python script was written to automate the creation of CNF files, since manual conversion to a `CNF` turned out to be cumbersome. The script also produces LaTeXoutput which is added to the appendix.

Since the noisy OR relations end up being encoded naively by considering each row of the CPT separately, a more compact encoding is obtained by replacing this part of the encoding with a one-liner. For example :

$$smokes(a) \Leftrightarrow aux_1 \vee aux_6 \vee aux_7$$

This can be converted to a `CNF` in the usual way :

$smokes(a) \vee \neg aux_1 \vee \neg aux_6 \vee \neg aux_7$
$\wedge(\neg smokes(a) \vee aux_1)$
$\wedge(\neg smokes(a) \vee aux_6)$
$\wedge(\neg smokes(a) \vee aux_7)$

Weighted Model Counting

Lifted Inference

Parameter Learning

References

[1] Mark Chavira and Adnan Darwiche. On probabilistic inference by weighted model counting. *Artificial Intelligence*, 172(6):772 – 799, 2008.