Instantiation to Support the Integration of Logical and Probabilistic Knowledge

Jingsong Wang and Marco Valtorta

Department of Computer Science and Engineering
University of South Carolina, Columbia SC 29208, USA

First Workshop on Grounding and Transformations for Theories with Variables

UNIVERSITY OF SOUTH CAROLINA

Outline

UNIVERSITY OF SOUTH CAROLINA

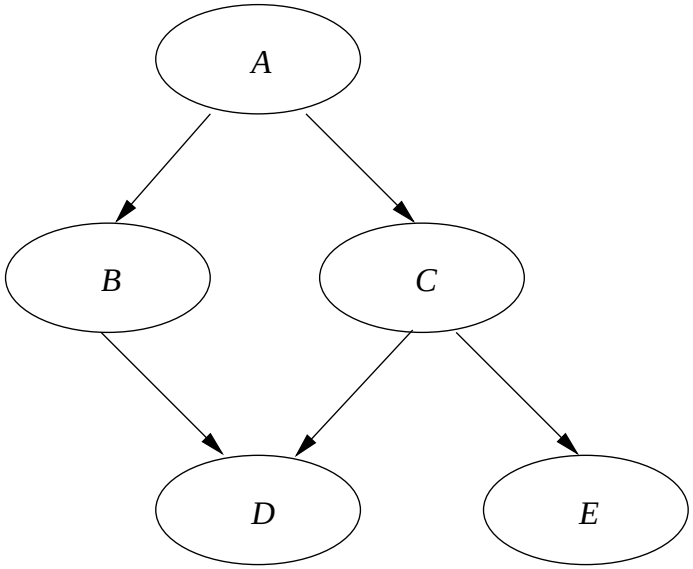
Outline

Knowledge-based systems

Knowledge-based systems normally comprise two components:

- A knowledge base (KB)
- An inference engine

Bayesian networks

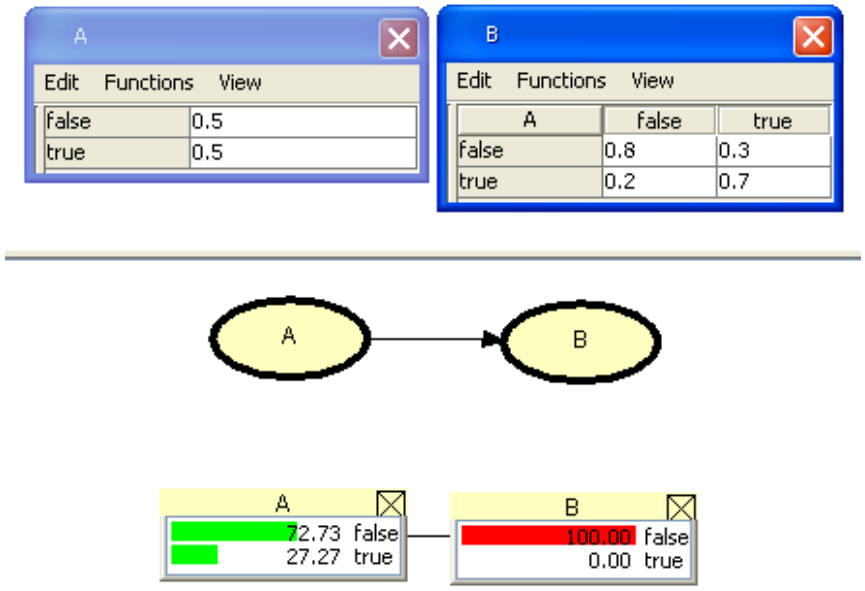


	A_1	A_2
B_1	0.2	0.1
B_2	0.6	0.6
B_3	0.2	0.3

- $\mathbf{Pr}(\mathcal{V}) = \prod_{V \in \mathcal{V}} \mathbf{Pr}(V \mid \pi(V))$
- The directed acyclic graph constrains $\mathbf{Pr}(\mathcal{V})$
- Few parameters are needed to descrive $\mathbf{Pr}(\mathcal{V})$, because of the independence constraints
- Probability update is fast

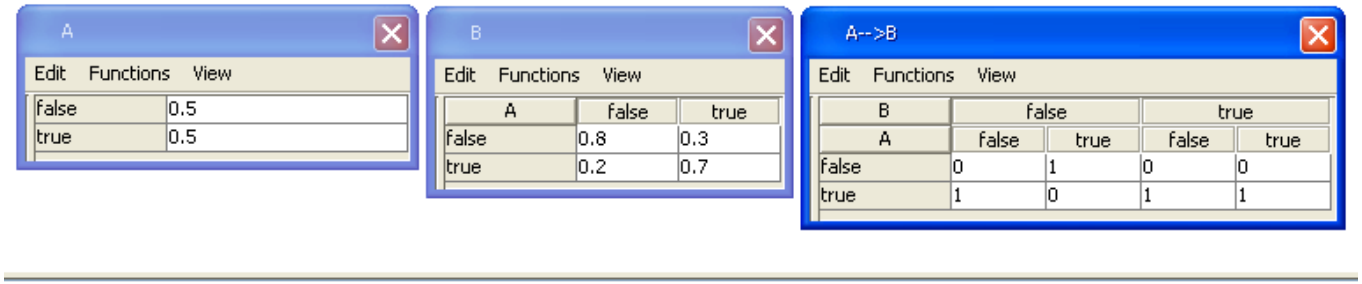
A Bayesian network with its CPTs and reasoning result given evidence

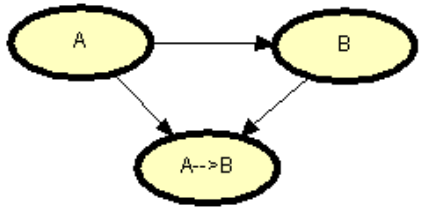
CPT stands for Conditional Probability Table

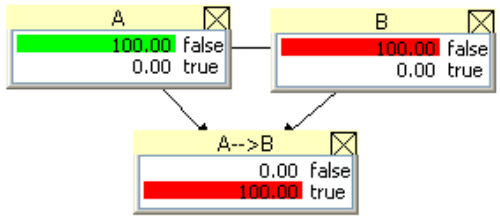


An extra node representing logical relations is added

Suppose we also have some logical knowledge about the relation between A and B in KB, such as $A \rightarrow B$.

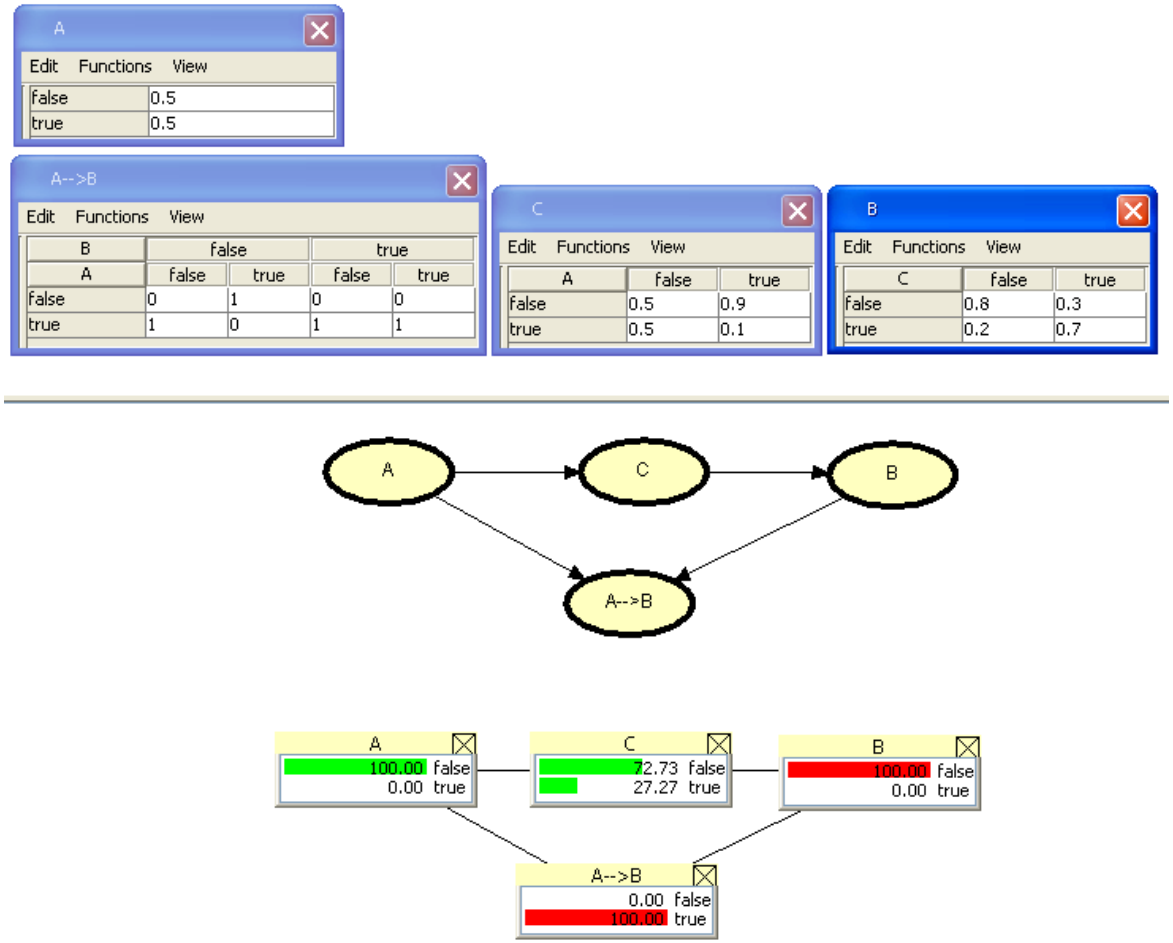






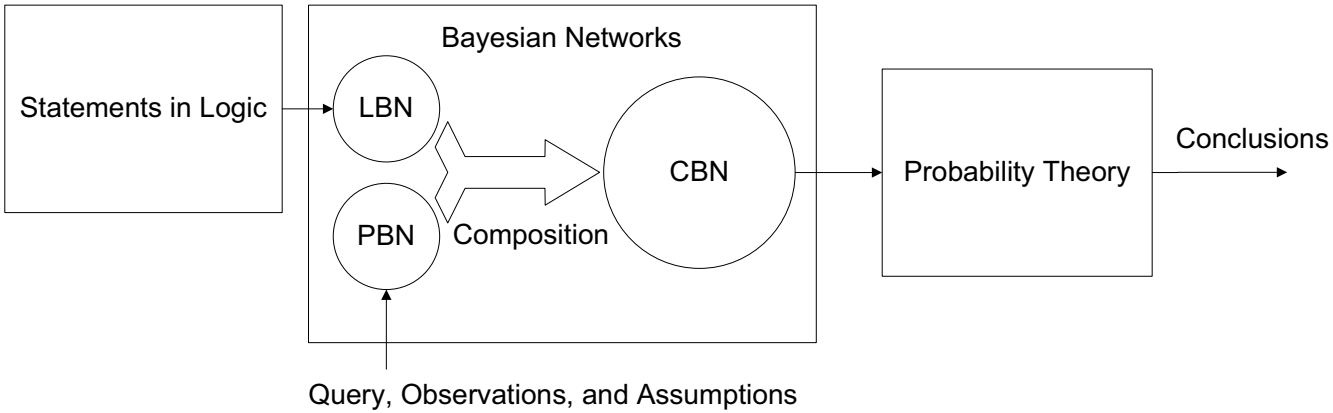
A more complex example

$\mathbf{Pr}(C|\mathbf{e}) = ?$



Reasoning systems with the combination of logic and Bayesian networks

1. Specify special knowledge using the most suitable languages, while reasoning in an uniform engine
2. Make use of pre-existing logical knowledge bases for probabilistic reasoning (to complete the model or minimize potential inconsistencies).



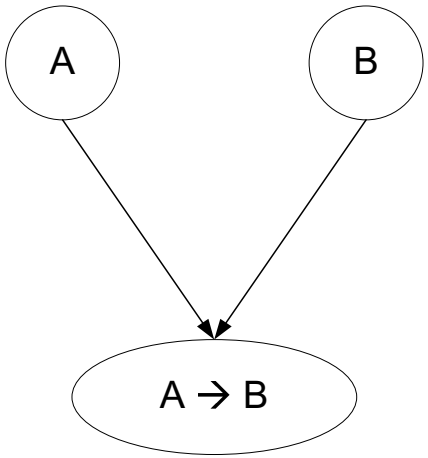
Outline

Relation search

Require: the probabilistic atom set $\mathcal{P}$ from PBN containing query, observations, and assumptions, the set of logical formulas KB $= \{R_1, R_2, \ldots, R_z\}$, $z = |\text{KB}|$, and the sets $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_z$. $\mathcal{A}_i$ comprises all the atoms appearing in its corresponding logical formula $R_i \in$ KB, where $1 \leq i \leq z$.

$\mathcal{V} = \mathcal{P}$;
$\mathcal{E} = \emptyset$;
$\mathcal{L} = \emptyset$;
for $i = 1$ to z **do**
    Tag R_i as not visited;
end for
while true do
    Changed $\Leftarrow$ **false**;
    for $i = 1$ to z **do**
        if R_i is not visited **then**
            if $\mathcal{A}_i \cap \mathcal{V} \neq \emptyset$ **then**
                $\mathcal{V} = \mathcal{V} \cup \mathcal{A}_i \cup \{R_i\}$;
                for all $A \in \mathcal{A}_i$ **do**
                    $\mathcal{E} = \mathcal{E} \cup \{(A, R_i)\}$;
                end for
                Build CPT Θ_i for R_i based on its logical structure;
                $\mathcal{L} = \mathcal{L} \cup \{\Theta_i\}$;
                Tag R_i as visited;
                Changed $\Leftarrow$ **true**;
            end if
        end if
    end for
    if *Changed* = **false then**
        break;
    end if
end while
return $BN(\mathcal{V}, \mathcal{E}, \mathcal{L})$;

Creating the CPT by a formula's logical structure



if $(A \rightarrow B)$				
A	true		false	
B	true	false	true	false
true	1	0	1	1
false	0	1	0	0

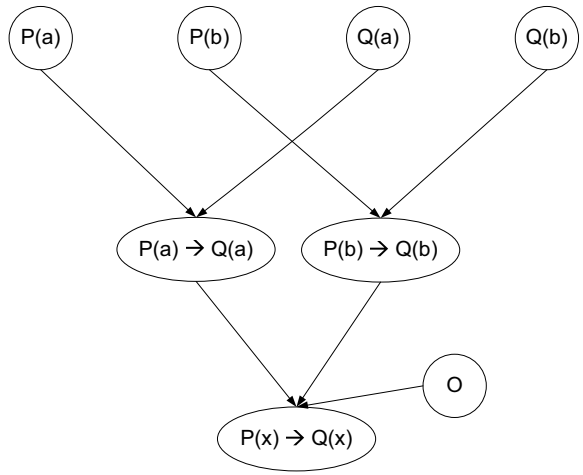
A FOL case example

Probabilistic atom set

$$\mathcal{P} = \{P(a), P(b)\}$$

The original FOL KB's formulas

1	$\forall x P(x) \rightarrow Q(x)$,
2	$\forall x H(x) \rightarrow L(x)$,
3	$H(c)$.



Outline

Theorem

Theorem

For a BN resulting from relation search, $\mathbf{G} = (\mathcal{V}, \mathcal{E}, \mathcal{L})$*, for any* $U \in \mathcal{V}$ *and any* $\mathcal{V}' \subseteq \mathcal{V}$*, if*

$$\mathcal{V}' \models U,$$

then

$$\mathbf{Pr}(U = true | \mathcal{V}' = true) = 1$$

in **G***, where* $\mathcal{V}'$ *= true means that all the nodes in* $\mathcal{V}'$ *of* **G** *are set to true.*

Weighted model count

Definition (Darwiche, 2009)

Let Δ be a propositional sentence over Boolean variables $X_1, ..., X_n$ and let Wt be a function that assigns a weight $Wt(x_i) \geq 0$ to each value x_i of variable X_i. The weighted model count (WMC) of Δ is defined as the sum of weights assigned to its models:

$$\textbf{\textit{WMC}}(\Delta) \stackrel{def}{=} \sum_{x_1,\ldots,x_n \models \Delta} Wt(x_1, ..., x_n),$$

where

$$Wt(x_1, ..., x_n) \stackrel{def}{=} \prod_{i=1}^{n} Wt(x_i).$$

CNF encoding

At first we define two types of Boolean variables: indicator variables and parameter variables. In particular, for each network variable X with parents $\mathbf{U}$, we have

- A Boolean variable I_x, called an *indicator variable*, for each value x of network variable X.
- A Boolean variable $P_{x|\mathbf{u}}$, called a *parameter variable*, for each instantiation $x\mathbf{u}$ of the family $X\mathbf{U}$.

Indicator clauses and parameter clauses

- A set of *indicator clauses* are generated for each variable X with values $x_1, x_2, ..., x_k$ as follows:

$$I_{x_1} \vee I_{x_2} \vee ... \vee I_{x_k}$$

$$\neg I_{x_i} \vee \neg I_{x_j}, \text{ for } i < j.$$

  These clauses ensure that exactly one indicator variable for variable X will be true.

- A set of *parameter clauses* are generated for each variable X and its parameter variable $P_{x|u_1,u_2,...,u_m}$. These include an *IP* clause,

$$I_{u_1} \wedge I_{u_2} \wedge ... \wedge I_{u_m} \wedge I_x \Rightarrow P_{x|u_1,u_2,...,u_m},$$

  and a set of *PI* clauses,

$$P_{x|u_1,u_2,...,u_m} \Rightarrow I_x$$

$$P_{x|u_1,u_2,...,u_m} \Rightarrow I_{u_i}, \text{ for } i = 1, ..., m.$$

  These clauses ensure that a parameter variable is true if and only if the corresponding indicator variables are true. We typically write these parameter clauses as one equivalence:

$$I_{u_1} \wedge I_{u_2} \wedge ... \wedge I_{u_m} \wedge I_x \Leftrightarrow P_{x|u_1,u_2,...,u_m}.$$

Correspondence lemma

Lemma (Darwiche, 2009)

Let **G** *be a Bayesian network inducing probability distribution* **Pr** *and let* $\Delta_{\mathbf{G}}$ *be its CNF encoding given by its indicator clauses and parameter clauses. For any evidence* $\mathbf{e} = e_1, ..., e_k$*, we have*

$$\mathbf{Pr(e)} = \mathbf{WMC}(\Delta_{\mathbf{G}} \wedge I_{e_1} \wedge \ldots \wedge I_{e_k}),$$

given the following weights:

$$Wt(I_x) = Wt(\neg I_x) = Wt(\neg P_{x|\mathbf{u}}) = 1$$

and

$$Wt(P_{x|\mathbf{u}}) = \theta_{x|\mathbf{u}}.$$

Proof - Main idea

- Main idea:
  1. Use weighted model counting, over the Bayesian network **G**, to conclude the joint probability of evidence $\mathbf{e} = \{U = \mathit{false}\} \cup \mathcal{V}' = \mathit{true}$ to be 0
  2. Follow Bayes' rule to conclude the posterior probability $\mathbf{Pr}(U = \mathit{true} | \mathcal{V}' = \mathit{true}) = 1$

- Notation:
  - $\mathcal{V}' = \{S_1, S_2, \ldots, S_k\}$, where $k \leq |\mathcal{V}|$
  - $\mathbf{e} = \{\overline{u}, s_1, s_2, \ldots, s_k\}$

Notation

- Suppose that after relation search, the set of partial formulas $\mathcal{R} = \{R_1, R_2, ..., R_m\}$ of KB are selected according to probabilistic atoms $\mathcal{P}$. $\mathcal{R} \subseteq$ KB.
- The set of atoms appearing in $\mathcal{R}$ is denoted by $\mathcal{A} = \{A_1, A_2, ...A_n\}$. Note that, for some $1 \leq i \leq n$ and $1 \leq j \leq m$, A_i and R_j might be exactly the same atomic formula.
- The set of atoms appearing in $\{U\} \cup \mathcal{V}'$ is denoted by $\mathcal{A}' = \{A_1, A_2, ...A_l\}$, where $l \leq n$. Note that S_i and A_j might be the same formula, for some $1 \leq i \leq k$ and $1 \leq j \leq l$, as S_i could also be an atomic formula.

Notation, ctd.

Suppose V is any node of **G**.

- If V represents an atomic formula, then it is a parent of some other nodes. (Note that there is no isolated node in **G**.)
- If V represents a compound formula, then it is a child of some atomic formulas that compose V.
- A compound formula represented by V contains some of the connectives $\neg$, $\vee$, $\wedge$, or $\rightarrow$.

Suppose $\{B_1, B_2, ..., B_h\}$ are atoms appearing in V, with $1 \leq h \leq n$.

- We use $\Theta_{V|B_1,B_2,\ldots,B_h}$ to represent the CPT of node V, one specific entry of which is denoted by θ, given parents $B_1, B_2, ..., B_h$.
- We use Π_{true} to represent any assignment over $\{B_1, B_2, ..., B_h\}$ that make formula V evaluate to true, and Π_{false} to represent any assignment over $\{B_1, B_2, ..., B_h\}$ that make formula V evaluate to false.

CPT entries

Because we build **G**'s CPTs purely based on the logical structure of each formula node, it is obvious that

$$\begin{aligned}\theta_{V=true|\Pi_{true}} &= 1,\\ \theta_{V=true|\Pi_{false}} &= 0,\\ \theta_{V=false|\Pi_{true}} &= 0,\\ \theta_{V=false|\Pi_{false}} &= 1.\end{aligned}$$

In another word, the probability of any state value of a compound formula node is 1 if this value is consistent with the formula's evaluation under its parents' value configuration. Otherwise, it is 0.

Evidence probability

Now consider the evidence $\mathbf{e} = \{U = \mathit{false}\} \cup \mathcal{V}' = \mathit{true}$, i.e., $\mathbf{e} = \{\overline{u}, s_1, s_2, ..., s_k\}$, and calculate its probability, represented by $\mathbf{Pr}(\mathbf{e})$.

The weighted model counting method shows that each instantiation of **G** corresponds to a model of **G**'s CNF encoding. (All the variables/literals of clauses corresponding to the instantiated values of nodes of **G** and related CPTs are set to true and all others are set to false.)

Compatible instantiations

- If we assume there are totally t nodes in $\mathbf{G}$, i.e., $|\mathbf{G}| = t$, then the list of all the nodes is $\{U, S_1, S_2, ..., S_k, Q_1, Q_2, ..., Q_{t-k-1}\}$, where Q nodes represent the other nodes that are not evidence.
- Note that a Q node could be either an atomic formula node or a compound formula node. The number of instantiations is 2^t, as each node's value is binary.
- However, we only need to consider the instantiations compatible to evidence $\{\overline{u}, s_1, s_2, ..., s_k\}$. This means, no matter what values it assigns to other nodes, the key instantiations, based on which we will calculate the joint probability, must assign nodes U = false and S_1 = true, S_2 = true, ..., S_k = true. The number of these instantiations is 2^{t-k-1}.

Logic perspective

Consider the set of formulas $\{U\} \cup \mathcal{V}' = \{U, S_1, S_2, ..., S_k\}$ and the set of atoms appearing in it $\mathcal{A}' = \{A_1, A_2, ...A_l\}$:

- Because $\mathcal{V}' \models U$, $\{\neg U\} \cup \mathcal{V}'$ is unsatisfiable. Thus an assignment of $\{U\} \cup \mathcal{V}'$, i.e., $\{U, S_1, S_2, ..., S_k\}$, over $\mathcal{A}' = \{A_1, A_2, ...A_l\}$ that is restricted to evaluate U to false while satisfying $\{S_1, S_2, ..., S_k\}$ does not exist.
- This means that for any assignment over $\mathcal{A}'$, as long as it makes U = false, there is at least one formula S_x that logically evaluates to false, where $1 \leq x \leq k$.

CNF encoding perspective

Because $\mathcal{V} = \mathcal{R} \cup \mathcal{A}$, any instantiation of **G** includes exactly one assignment over $\mathcal{A}$, the set of atomic formulas appearing in compound formulas of **G**.

For any instantiation c of **G** compatible to evidence $\{\overline{u}, s_1, s_2, ..., s_k\}$, it might evaluate formula U to either true or false.

CNF encoding perspective, ctd.

If U = false, we can always find at least one formula $S_X \in \mathcal{V}'$ that logically evaluates to false. We use c_X to represent the partial assignment over atoms only related to S_X ($c_X \subseteq c$, as c is an assignment over not only atomic formula nodes, but also compound formula nodes). Then considering one model ω of the CNF encoding corresponding to this instantiation c, the weight of one parameter variable/literal that is related to the evidential value S_X = true, $P_{s_X|c_X}$, is

$$Wt(P_{s_X|c_X}) = \theta_{s_X|c_X} = 0.$$

So the weight of the model ω is

$$Wt(\omega) = Wt(P_{s_X|c_X}) * \prod_{i=1}^{t-1} Wt(P_i) = 0,$$

where P_i represents another parameter variable/literal of the model ω that is not $P_{s_X|c_X}$.

CNF encoding perspective, ctd.

If U = true, and if we use c_u to denote the partial assignment over atoms only related to U, then

$$Wt(P_{\overline{u}|c_u}) = \theta_{\overline{u}|c_u} = 0.$$

The weight of the model ω is still

$$Wt(\omega) = Wt(P_{\overline{u}|c_u}) * \prod_{i=1}^{t-1} Wt(P_i) = 0.$$

CNF encoding perspective, ctd.

This conclusion holds for all the other models. Thus

$$\mathbf{Pr}(\mathbf{e}) = \mathbf{Pr}(\overline{u}, \mathcal{V}' = true) = \mathbf{Pr}(\overline{u}, s_1, s_2, ..., s_k) = \sum_{i=1}^{2^{t-k-1}} Wt(\omega_i) = 0,$$

where ω_i represents a model corresponding to the different instantiation of **G** compatible to evidence $\{\overline{u}, s_1, s_2, ..., s_k\}$.

CNF encoding perspective: conclusion

Finally

$$\begin{aligned}
\mathbf{Pr}(u|\mathcal{V}' = true) &= \frac{\mathbf{Pr}(u, \mathcal{V}' = true)}{\mathbf{Pr}(u, \mathcal{V}' = true) + \mathbf{Pr}(\overline{u}, \mathcal{V}' = true)} \\
&= \frac{\mathbf{Pr}(u, \mathcal{V}' = true)}{\mathbf{Pr}(u, \mathcal{V}' = true) + 0} \\
&= 1.
\end{aligned}$$

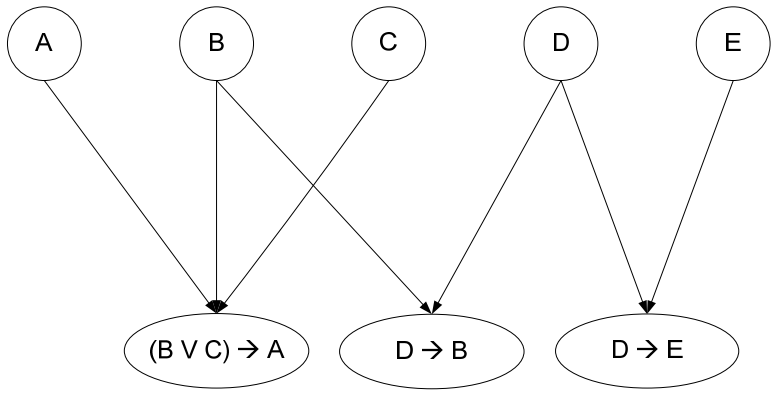
Example of use of WMC in the proof

Query

$$D$$

The original KB's formulas

1	$(B \vee C) \rightarrow A$
2	$D \rightarrow B$
3	$D \rightarrow E$,
4	$F \rightarrow G$



Search result

$$\mathcal{R} = \{(B \vee C) \rightarrow A, D \rightarrow B, D \rightarrow E\},$$
$$\mathcal{V} = \{A, B, C, D, E, (B \vee C) \rightarrow A, D \rightarrow B, D \rightarrow E\}.$$

CPTs for the example

A	B	C	$(B \vee C) \rightarrow A$	$\Theta_{(B \vee C) \rightarrow A \mid A,B,C}$
T	T	T	T	1
			F	0
T	T	F	T	1
			F	0
T	F	T	T	1
			F	0
T	F	F	T	1
			F	0
F	T	T	T	0
			F	1
F	T	F	T	0
			F	1
F	F	T	T	0
			F	1
F	F	F	T	1
			F	0

B	D	$D \rightarrow B$	$\Theta_{D \rightarrow B \mid B,D}$
T	T	T	1
		F	0
T	F	T	1
		F	0
F	T	T	0
		F	1
F	F	T	1
		F	0

D	E	$D \rightarrow E$	$\Theta_{D \rightarrow E \mid D,E}$
T	T	T	1
		F	0
T	F	T	0
		F	1
F	T	T	1
		F	0
F	F	T	1
		F	0

Example, ctd.

- For some set of formulas $\mathcal{V}' \subseteq \mathcal{V}$, e.g.,
$$\mathcal{V}' = \{D, (B \vee C) \rightarrow A, D \rightarrow B\},$$
it is obvious that
$$\mathcal{V}' \models A.$$
- We want to show:
$$\mathbf{Pr}(A = \mathit{true} | D = \mathit{true}, (B \vee C) \rightarrow A = \mathit{true}, D \rightarrow B = \mathit{true}) = 1,$$
which can be simplified as:
$$\mathbf{Pr}(a | d, (b \vee c) \rightarrow a, d \rightarrow b) = 1.$$
- We start by following the method of weighted model counting to calculate
$$\mathbf{Pr}(\overline{a}, d, (b \vee c) \rightarrow a, d \rightarrow b),$$
the probability of evidence $\{A = \mathit{false}, D = \mathit{true}, (B \vee C) \rightarrow A = \mathit{true}, D \rightarrow B = \mathit{true}\}$ in the Bayesian network **G**.

CNF encoding

Indicator Clauses			
A	$I_a \vee I_{\overline{a}}$	$\neg I_a \vee \neg I_{\overline{a}}$	
B	$I_b \vee I_{\overline{b}}$	$\neg I_b \vee \neg I_{\overline{b}}$	
... ...	... ...		
E	$I_e \vee I_{\overline{e}}$	$\neg I_e \vee \neg I_{\overline{e}}$	
$D \rightarrow B$	$I_{d\rightarrow b} \vee I_{\overline{d\rightarrow b}}$	$\neg I_{d\rightarrow b} \vee \neg I_{\overline{d\rightarrow b}}$	
$D \rightarrow E$	$I_{d\rightarrow e} \vee I_{\overline{d\rightarrow e}}$	$\neg I_{d\rightarrow e} \vee \neg I_{\overline{d\rightarrow e}}$	
... ...	... ...		
Parameter Clauses			
A	$I_a \Leftrightarrow P_a$	$I_{\overline{a}} \Leftrightarrow P_{\overline{a}}$	
B	$I_b \Leftrightarrow P_b$	$I_{\overline{b}} \Leftrightarrow P_{\overline{b}}$	
... ...	... ...		
E	$I_e \Leftrightarrow P_e$	$I_{\overline{e}} \Leftrightarrow P_{\overline{e}}$	
$D \rightarrow B$	$I_d \wedge I_b \wedge I_{d\rightarrow b} \Leftrightarrow P_{d\rightarrow b\|d,b}$		$I_d \wedge I_{\overline{b}} \wedge I_{d\rightarrow b} \Leftrightarrow P_{d\rightarrow b\|d,\overline{b}}$
	$I_{\overline{d}} \wedge I_b \wedge I_{d\rightarrow b} \Leftrightarrow P_{d\rightarrow b\|\overline{d},b}$		$I_{\overline{d}} \wedge I_{\overline{b}} \wedge I_{d\rightarrow b} \Leftrightarrow P_{d\rightarrow b\|\overline{d},\overline{b}}$
	$I_d \wedge I_b \wedge I_{\overline{d\rightarrow b}} \Leftrightarrow P_{\overline{d\rightarrow b}\|d,b}$		$I_d \wedge I_{\overline{b}} \wedge I_{\overline{d\rightarrow b}} \Leftrightarrow P_{\overline{d\rightarrow b}\|d,\overline{b}}$
	$I_{\overline{d}} \wedge I_b \wedge I_{\overline{d\rightarrow b}} \Leftrightarrow P_{\overline{d\rightarrow b}\|\overline{d},b}$		$I_{\overline{d}} \wedge I_{\overline{b}} \wedge I_{\overline{d\rightarrow b}} \Leftrightarrow P_{\overline{d\rightarrow b}\|\overline{d},\overline{b}}$
$D \rightarrow E$	$I_d \wedge I_e \wedge I_{d\rightarrow e} \Leftrightarrow P_{d\rightarrow e\|d,e}$		$I_d \wedge I_{\overline{e}} \wedge I_{d\rightarrow e} \Leftrightarrow P_{d\rightarrow e\|d,\overline{e}}$
	$I_{\overline{d}} \wedge I_e \wedge I_{d\rightarrow e} \Leftrightarrow P_{d\rightarrow e\|\overline{d},e}$		$I_{\overline{d}} \wedge I_{\overline{e}} \wedge I_{d\rightarrow e} \Leftrightarrow P_{d\rightarrow e\|\overline{d},\overline{e}}$
	$I_d \wedge I_e \wedge I_{\overline{d\rightarrow e}} \Leftrightarrow P_{\overline{d\rightarrow e}\|d,e}$		$I_d \wedge I_{\overline{e}} \wedge I_{\overline{d\rightarrow e}} \Leftrightarrow P_{\overline{d\rightarrow e}\|d,\overline{e}}$
	$I_{\overline{d}} \wedge I_e \wedge I_{\overline{d\rightarrow e}} \Leftrightarrow P_{\overline{d\rightarrow e}\|\overline{d},e}$		$I_{\overline{d}} \wedge I_{\overline{e}} \wedge I_{\overline{d\rightarrow e}} \Leftrightarrow P_{\overline{d\rightarrow e}\|\overline{d},\overline{e}}$
... ...	... ...		

Compatible instantiations

Given evidence

$$\{A = false, D = true, (B \vee C) \rightarrow A = true, D \rightarrow B = true\},$$

we can list all the compatible instantiations of the Bayesian network **G**:

	Evidence Nodes (fixed values)				Other Nodes			
	A	D	$(B \vee C) \rightarrow A$	$D \rightarrow B$	B	C	E	$D \rightarrow E$
1	$\bar{a}$	d	$(b \vee c) \rightarrow a$	$d \rightarrow b$	b	c	e	$d \rightarrow e$
2	$\bar{a}$	d	$(b \vee c) \rightarrow a$	$d \rightarrow b$	b	c	e	$\overline{d \rightarrow e}$
3	$\bar{a}$	d	$(b \vee c) \rightarrow a$	$d \rightarrow b$	b	c	$\bar{e}$	$d \rightarrow e$
4	$\bar{a}$	d	$(b \vee c) \rightarrow a$	$d \rightarrow b$	b	c	$\bar{e}$	$\overline{d \rightarrow e}$
...	...	...	...	...	...	...	...	...
16	$\bar{a}$	d	$(b \vee c) \rightarrow a$	$d \rightarrow b$	$\bar{b}$	$\bar{c}$	$\bar{e}$	$\overline{d \rightarrow e}$

Each instantiation corresponds to a model of the CNF encoding :

ω	Truth assignment sets these variables to true and all others to false
1	$I_{\bar{a}}, I_d, I_{(b\vee c)\rightarrow a}, I_{d\rightarrow b}, I_b, I_c, I_e, I_{d\rightarrow e}, P_{\bar{a}}, P_d, \boldsymbol{P_{(b\vee c)\rightarrow a\mid \bar{a},b,c}}, P_{d\rightarrow b\mid b,d}, P_b, P_c, P_e, P_{d\rightarrow e\mid d,e}$
2	$I_{\bar{a}}, I_d, I_{(b\vee c)\rightarrow a}, I_{d\rightarrow b}, I_b, I_c, I_e, I_{\overline{d\rightarrow e}}, P_{\bar{a}}, P_d, \boldsymbol{P_{(b\vee c)\rightarrow a\mid \bar{a},b,c}}, P_{d\rightarrow b\mid b,d}, P_b, P_c, P_e, \boldsymbol{P_{\overline{d\rightarrow e}\mid d,e}}$
3	$I_{\bar{a}}, I_d, I_{(b\vee c)\rightarrow a}, I_{d\rightarrow b}, I_b, I_c, I_{\bar{e}}, I_{d\rightarrow e}, P_{\bar{a}}, P_d, \boldsymbol{P_{(b\vee c)\rightarrow a\mid \bar{a},b,c}}, P_{d\rightarrow b\mid b,d}, P_b, P_c, P_{\bar{e}}, \boldsymbol{P_{d\rightarrow e\mid d,\bar{e}}}$
4	$I_{\bar{a}}, I_d, I_{(b\vee c)\rightarrow a}, I_{d\rightarrow b}, I_b, I_c, I_{\bar{e}}, I_{\overline{d\rightarrow e}}, P_{\bar{a}}, P_d, \boldsymbol{P_{(b\vee c)\rightarrow a\mid \bar{a},b,c}}, P_{d\rightarrow b\mid b,d}, P_b, P_c, P_{\bar{e}}, P_{\overline{d\rightarrow e}\mid d,\bar{e}}$
...	... ...
16	$I_{\bar{a}}, I_d, I_{(b\vee c)\rightarrow a}, I_{d\rightarrow b}, I_{\bar{b}}, I_{\bar{c}}, I_{\bar{e}}, I_{\overline{d\rightarrow e}}, P_{\bar{a}}, P_d, P_{(b\vee c)\rightarrow a\mid \bar{a},\bar{b},\bar{c}}, \boldsymbol{P_{d\rightarrow b\mid \bar{b},d}}, P_{\bar{b}}, P_{\bar{c}}, P_{\bar{e}}, P_{\overline{d\rightarrow e}\mid d,\bar{e}}$

Model weight

- In each model, there is at least one variable/literal whose weight is equal to 0. For example, in model 2, ω_2, they are $P_{(b\vee c)\rightarrow a|\bar{a},b,c}$ and $P_{\overline{d\rightarrow e}|d,e}$, as we know from the logical relations:

$$Wt(P_{(b\vee c)\rightarrow a|\bar{a},b,c}) = \theta_{(b\vee c)\rightarrow a|\bar{a},b,c} = 0$$
and
$$Wt(P_{\overline{d\rightarrow e}|d,e}) = \theta_{\overline{d\rightarrow e}|d,e} = 0.$$

  Thus,

$$Wt(\omega_2) = 0,$$

  as the weight of model ω_2 is based on the product of the weights of all the variables/literals in ω_2 and there is at least one 0 weight among these factors. This holds for all the other models.

- Therefore even if there are 16 models, the sum of their weights, the actual probability of evidence, is still 0, i.e.,

$$\mathbf{Pr}(\bar{a}, d, (b \vee c) \rightarrow a, d \rightarrow b) = Wt(\omega_1) + Wt(\omega_2) + ... + Wt(\omega_{16}) = 0.$$

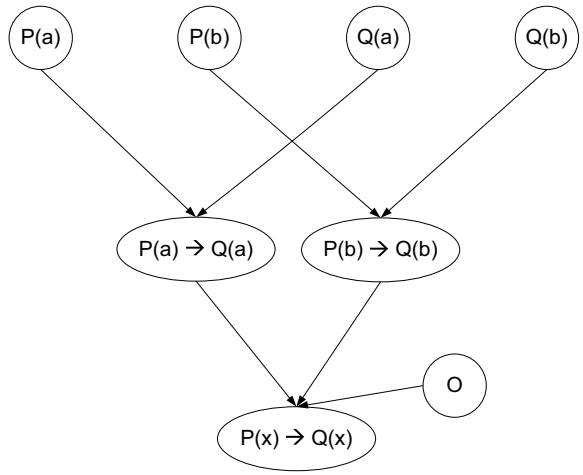
Propositional case example: conclusion

Now we can use this conclusion and Bayes' rule to calculate $\mathbf{Pr}(a|d,(b\vee c)\rightarrow a, d\rightarrow b)$:

$$\begin{aligned}\mathbf{Pr}(a|d,(b\vee c)\rightarrow a, d\rightarrow b) &= \frac{\mathbf{Pr}(a,d,(b\vee c)\rightarrow a, d\rightarrow b)}{\mathbf{Pr}(a,d,(b\vee c)\rightarrow a, d\rightarrow b)+\mathbf{Pr}(\bar{a},d,(b\vee c)\rightarrow a, d\rightarrow b)}\\ &= \frac{\mathbf{Pr}(a,d,(b\vee c)\rightarrow a, d\rightarrow b)}{\mathbf{Pr}(a,d,(b\vee c)\rightarrow a, d\rightarrow b)+0}\\ &= 1.\end{aligned}$$

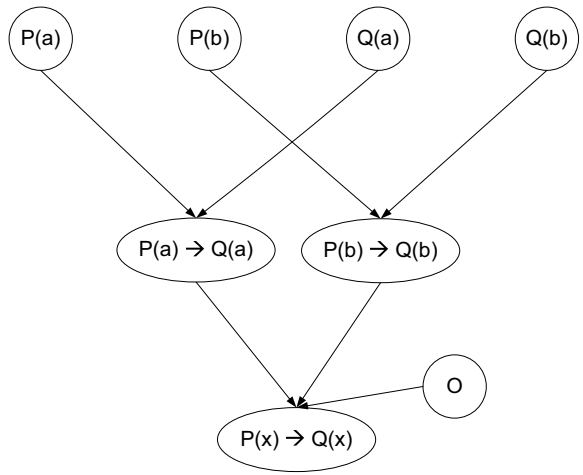
First-order case proof

- If $\mathcal{V}' \models U$, then FOL theory $\{\neg U\} \cup \mathcal{V}' = \{\neg U, S_1, S_2, ..., S_k\}$ is logically unsatisfiable. We define $F = U \wedge S_1 \wedge S_2 \wedge ... \wedge S_k$ and $F' = \neg U \wedge S_1 \wedge S_2 \wedge ... \wedge S_k$. Then a Herbrand structure of F is also the one of F' and vice versa. Note that every subformula of F has a corresponding node in **G**. We know that F' is unsatisfiable. This means that all interpretations (more commonly called structures in FOL) of F', including its Herbrand structures, that evaluate formula U to false will also evaluate at least one formula S_x $(1 \leq x \leq k)$ to false.



First-order case proof, ctd.

- For BN **G**, each instantiation includes one instantiation of **G**'s root nodes, which are all ground atomic formulas because of the way **G** is built. (The root nodes also include the *O* nodes, which are also atomic propositions.) Such an instantiation of root nodes in **G** has defined a Herbrand structure for the language of **G**. Thus given an instantiation **G**, there is a corresponding evaluation for any formula appearing in **G** and this evaluation result is fixed.
- In addition, for each quantified formula appearing in **G**, all its possible groundings based on constants available in the relation search result ($\mathcal{R} \cup \mathcal{P}$) also appear in **G**.



Proof, ctd.

- Consider the special CNF encoding of BN **G**. Any instantiation of **G** corresponds to a model of such a CNF encoding. To compute **Pr(e)** through the weighted model count, we only consider instantiations of **G** that are compatible with evidence $\mathbf{e} = \{\overline{u}, s_1, s_2, ..., s_k\}$.
- We choose one arbitrary instantiation of **G** compatible with **e**. The Herbrand structure contained in the current instantiation is denoted by H. We know that formula U can evaluate to true or false under H. We consider these two cases separately.

Proof: U evaluates to false in H

U **is false in** H. Because of the unsatisfiability of F', we know that F' does not have a Herbrand model. Therefore if U evaluates to false in H, there must be at least a S_x that evaluates to false in H. As we know that the nodes of **G** are in three levels, node S_x might appear in the first level (being an atomic formula node), in the second level (being a ground formula), or in the third level (being a quantified formula).

- Note that atomic formulas, appearing as root nodes in **G**, definitely evaluate to the same values as their state values in the instantiation. Remember that all the nodes $S_1, S_2, ..., S_k$ are in state true, as the current instantiation is compatible with evidence **e**. Since S_x is false in H, S_x could not be in the first level. This also excludes the case that S_x is an O node.
- If S_x is in the second level, then there is an inconsistency between its logical evaluation, which is false based on its parents' state values, and its state value, which is true as determined by **e**.
- If node S_x is in the third level, i.e., it is a quantified formula, there are two cases for possible instantiations of its parent nodes, which are groundings of S_x, in **G**:
  1. All the parent nodes of S_x are in state value true.
  2. There is at least one parent node of S_x in state false, including its O node.

  Therefore, if node S_x is in the third level of **G**, for any possible instantiation compatible with **e**, we can always find a node with logical inconsistency. This node might be S_x itself, or one of its parents S_{xg}.

Proof: U evaluates to true in H

U **is true in** H. There are three cases for the appearance of U: U is an O node, a ground formula that is not an O node, and a quantified formula.

1. U is an O node. This case is impossible.
   Being a root node, an O node should have the same evaluation as its state value false, which is compatible with evidence **e**.
2. U is a ground formula but not an O node.
   Still, U cannot be a root node. Thus U is a second level grounding. U has a state value false, which is compatible with evidence **e**, but it evaluates to true. Thus U itself has an inconsistency.
3. U is a quantified formula. We have a similar analysis as before.
   If all the parent nodes of U are in state value true, then there is an inconsistency in U, as we know that the CPT of U is an AND table, but its state value is false to be compatible with **e**. Otherwise, there is at least one parent in state value false. Based on the definition of the truth value of a quantified formula, all the parent formulas (node O or not) of U in **G** need to evaluate to true in H. Being a root node, the O node cannot be in state value false, as its state value should be the same value as its evaluation in H. Therefore there must be a grounding in state value false, which however evaluates to true. Then this grounding has an inconsistency.

Proof: model weight

Because of the way CPTs are built in **G**, as shown in the propositional case analysis, the entries of a CPT of a non-root node in **G** all have value 0, if the logical evaluation of its logical formula, given its parents' state values, is inconsistent with the node's real state value. For a node with such inconsistency, denoted by S, if we use c to represent the state values of its parent nodes in the current instantiation, and the corresponding truth assignment of CNF encoding is denoted by ω, we know that

$$Wt(P_{s|c}) = \theta_{s|c} = 0,$$

and the weight of the model ω is

$$Wt(\omega) = Wt(P_{s|c}) * \prod_{i=1}^{t-1} Wt(P_i) = 0,$$

where t is the number of nodes in **G**, i.e., $|\mathbf{G}| = t$, and P_i represents the other parameter literal of the model ω that is not $P_{s|c}$.

First-order case proof: evidence probability and conclusion

This result holds for all the other models. Thus

$$\mathbf{Pr}(\mathbf{e}) = \mathbf{Pr}(\overline{u}, \mathcal{V}' = true) = \mathbf{Pr}(\overline{u}, s_1, s_2, ..., s_k) = \sum_{i=1}^{2^{t-k-1}} Wt(\omega_i) = 0,$$

where ω_i represents the model corresponding to the different instantiation of **G** compatible with evidence $\{\overline{u}, s_1, s_2, ..., s_k\}$. Thus, as claimed in the statement of the theorem,

$$\mathbf{Pr}(u|\mathcal{V}' = true) = \frac{\mathbf{Pr}(u, \mathcal{V}' = true)}{\mathbf{Pr}(u, \mathcal{V}' = true) + \mathbf{Pr}(\overline{u}, \mathcal{V}' = true)} = \frac{\mathbf{Pr}(u, \mathcal{V}' = true)}{\mathbf{Pr}(u, \mathcal{V}' = true) + 0} = 1.$$

This concludes the proof of correctness for the first-order case.

Outline

Why partial instantiation?

In the FOL case, as the number of atoms containing different variables in a quantified formula increases, the number of groundings could increase exponentially. This might make the BN **G** very large. However, many nodes, including atoms and groundings, in this BN are not meaningful in real use.

Context predicates and context facts

In a well-defined KB, besides formulas representing rules, there are some facts that represent basic and fixed features of a few context constants in KB. For atoms describing such features, any groundings of them that are not explicitly specified in KB are regarded impossible. This is often referred as closed world assumption, i.e., for a feature F of a sequence of constants $\mathcal{C}$ in the domain, if $F(\mathcal{C})$ is not listed as a fact, then we believe $F(\mathcal{C})$ is false. We call the predicates like F *context predicates* and the set of related facts *context facts*.

Therefore, we can control the number of groundings of quantified formulas by discarding meaningless instantiations.

Implicative Normal Form

In addition, we assume that all the rules in KB are in Implicative Normal Form (INF). Note that any formula can be easily translated into INF.

For example, $(A_1 \wedge A_2) \vee (B_1 \wedge B_2) \rightarrow C$, where A_1, A_2, B_1, and B_2 are literals and C is a subformula, can be converted into $A_1 \wedge A_2 \rightarrow C$ and $B_1 \wedge B_2 \rightarrow C$.

One extra step

- We add one extra step for the instantiation process of quantified formulas in INF. We will still try all the possible instantiations of a quantified formula that contains context predicates. We discard a grounding unless all of the context predicates appearing in its body are context facts.
- The idea is simple. Each rule in KB naturally adds one constraint to the set of possible worlds. However, when one part of the body of the rule, which is in INF, is deterministically false, the rule will not constrain anything, based on the definition of logical implication. Then adding the rule is truly meaningless.
- Thus we can just ignore such groundings when building the BN. In applications, the users can define their context predicates and context facts flexibly for controlling the size of resulting BN.

Outline

Conclusion and future work

In this paper, we presented a new way of translating logical knowledge into Bayesian networks that supports a new approach to the integration problem of logical and probabilistic reasoning that is easy to understand, simple to implement, and efficient to execute.

Future work:

- The method presented in this paper can be used not only for the general integration problem but also the traditional BN learning problem.
- We are exploring how to use soft evidence for the improvement of accurate reasoning for many problems, such as classification.

Questions?