THE BAYESIAN ONTOLOGY REASONER IS BORN!

İsmail İlkan Ceylan, Julian Mendez and Rafael Peñaloza

Athens, 6th June 2015
Bayesian Networks

Probabilistic graphical models that can compactly represent the joint probability distribution

\[
P(\neg x, \neg y, z) = P(z | \neg x, \neg y) P(\neg y | \neg x) P(\neg x) = 0.9 \times 0.5 \times 0.3 = 0.135
\]
Bayesian Networks

Probabilistic graphical models that can compactly represent the joint probability distribution

\[
P(\neg x, \neg y, z) = P(z|\neg x, \neg y)P(\neg y|\neg x)P(\neg x)
\]
\[
= 0.9 \times 0.5 \times 0.3
\]
\[
= 0.135
\]
The Description Logic \( \mathcal{BEL} \)

Extends \( \mathcal{EL} \) by defining a joint probability distribution over the axioms of an \( \mathcal{EL} \) ontology.
The Description Logic $\mathcal{BEL}$

Extends $\mathcal{EL}$ by defining a joint probability distribution over the axioms of an $\mathcal{EL}$ ontology

$$\mathcal{EL} \quad \text{concept language} \quad V \quad \text{of propositional variables} \quad \text{contextual TBox}$$

$$C, D \quad \kappa = \{x, \neg y\} \quad \langle C \sqsubseteq D : \kappa \rangle$$
The Description Logic \(\textbf{BEL}\)

Extends \(\textbf{EL}\) by defining a joint probability distribution over the axioms of an \(\textbf{EL}\) ontology

\[
\begin{align*}
\text{\(\text{EL}\) concept language} & \quad \text{\(V\) of propositional variables} & \quad \text{contextual TBox} \\
C, D & \quad \kappa = \{x, \neg y\} & \quad \langle C \sqsubseteq D : \kappa \rangle
\end{align*}
\]

\(\textbf{BEL}\) Knowledge Base \(\mathcal{K} = (\mathcal{T}, \mathcal{B})\)

\[
\begin{align*}
\mathcal{T} & \quad \text{a (contextual) TBox} \\
\mathcal{B} & \quad \text{a Bayesian network}
\end{align*}
\]

\} \quad \text{defined over the same variables} \quad V
The Description Logic $\mathcal{BEL}$

Extends $\mathcal{EL}$ by defining a joint probability distribution over the axioms of an $\mathcal{EL}$ ontology

\[
\mathcal{EL} \text{ concept language } V \text{ of propositional variables contextual TBox } \\
C, D \quad \kappa = \{x, \neg y\} \quad \langle C \sqsubseteq D : \kappa \rangle
\]

$\mathcal{BEL}$ Knowledge Base $\mathcal{K} = (\mathcal{T}, \mathcal{B})$

\[
\mathcal{T} \text{ a (contextual) TBox } \quad \mathcal{B} \text{ a Bayesian network } \quad \text{defined over the same variables } V
\]

Contextual interpretations

\[
\mathcal{I} = (\Delta^{\mathcal{I}}, \mathcal{I}, \nu^{\mathcal{I}}) \text{ with } \mathcal{I} \models \langle C \sqsubseteq D : \kappa \rangle \text{ iff } \nu^{\mathcal{I}} \not\models \kappa \text{ or } C^{\mathcal{I}} \subseteq D^{\mathcal{I}}
\]
The Description Logic $\mathcal{BEL}$

Extends $\mathcal{EL}$ by defining a joint probability distribution over the axioms of an $\mathcal{EL}$ ontology

$\mathcal{EL}$ concept language $V$ of propositional variables contextual TBox

$C, D$ \hspace{1cm} $\kappa = \{x, \neg y\}$ \hspace{1cm} $\langle C \sqsubseteq D : \kappa \rangle$

$\mathcal{BEL}$ Knowledge Base $\mathcal{K} = (\mathcal{T}, \mathcal{B})$

$\mathcal{T}$ a (contextual) TBox \hspace{1cm} $\mathcal{B}$ a Bayesian network \hspace{1cm} defined over the same variables $V$

Contextual interpretations

$\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I}, \nu^\mathcal{I})$ with $\mathcal{I} \models \langle C \sqsubseteq D : \kappa \rangle$ iff $\nu^\mathcal{I} \not\models \kappa$ or $C^\mathcal{I} \subseteq D^\mathcal{I}$

Probabilistic interpretation $\mathcal{P} = (\mathcal{I}, P_\mathcal{I})$ \hspace{1cm} Probabilistic model $\mathcal{P} = (\mathcal{I}, P_\mathcal{I})$

$\mathcal{I}$ a set of contextual interpretations \hspace{1cm} $\mathcal{I} \models \mathcal{T}$ for all $\mathcal{I} \in \mathcal{I}$

$P_\mathcal{I}$ a probability distribution over $\mathcal{I}$ \hspace{1cm} $\sum_{\mathcal{I} \in \mathcal{I}} P_\mathcal{I}(\mathcal{I}) = P_B(\mathcal{W})$
Probabilistic Subsumption

\[ K = (T, B) \] a knowledge base, \( P \) a probabilistic interpretation

\[ \Pr(\text{subsumption} \mid K) = \inf_{P \models K} \Pr(\text{subsumption} \mid P) \]

\[ \Pr(\text{subsumption} \mid P) = \sum_{T \models \text{subsumption}} P(T) \]

\( \text{JELIA 2014} \) and is \( \text{P}-\text{complete} \)
Probabilistic Subsumption

$\mathcal{K} = (\mathcal{T}, \mathcal{B})$ a knowledge base, $\mathcal{P}$ a probabilistic interpretation

Probability of a subsumption w.r.t. $\mathcal{P}$

$$P_\mathcal{P}(C \sqsubseteq D) = \sum_{\mathcal{I} \in \mathcal{G}} P_\mathcal{P}(\mathcal{I})$$

$$\quad \text{s.t. } \mathcal{I} \models C \sqsubseteq D$$

$I_3 \sim W_3$

$I_2 \sim W_2$

$I_1 \sim W_1$
Probabilistic Subsumption

\( \mathcal{K} = (T, B) \) a knowledge base, \( \mathcal{P} \) a probabilistic interpretation

Probability of a subsumption w.r.t. \( \mathcal{P} \)

\[
P_{\mathcal{P}}(C \sqsubseteq D) = \sum_{\mathcal{I} \models C \sqsubseteq D} P_{\mathcal{J}}(\mathcal{I})
\]

Probability of a subsumption w.r.t. \( \mathcal{K} \)

\[
P_{\mathcal{K}}(C \sqsubseteq D) = \inf_{\mathcal{P} \models \mathcal{K}} P_{\mathcal{P}}(C \sqsubseteq D)
\]
Probabilistic Subsumption

\( \mathcal{K} = (\mathcal{T}, B) \) a knowledge base, \( \mathcal{P} \) a probabilistic interpretation

Probability of a subsumption w.r.t. \( \mathcal{P} \)

\[
P_\mathcal{P}(C \sqsubseteq D) = \sum_{\exists I \in \mathcal{I} \mid I \models C \sqsubseteq D} P_\mathcal{I}(I)
\]

Probability of a subsumption w.r.t. \( \mathcal{K} \)

\[
P_\mathcal{K}(C \sqsubseteq D) = \inf_{\mathcal{P} \models \mathcal{K}} P_\mathcal{P}(C \sqsubseteq D)
\]
Probabilistic Subsumption

\( \mathcal{K} = (\mathcal{T}, \mathcal{B}) \) a knowledge base, \( \mathcal{P} \) a probabilistic interpretation

Probability of a subsumption w.r.t. \( \mathcal{P} \)

\[
P_{\mathcal{P}}(C \sqsubseteq D) = \sum_{\mathcal{I} \models \mathcal{K} \models C \sqsubseteq D} \mathcal{P}(\mathcal{I})
\]

Probability of a subsumption w.r.t. \( \mathcal{K} \)

\[
P_{\mathcal{K}}(C \sqsubseteq D) = \inf_{\mathcal{P} \models \mathcal{K}} P_{\mathcal{P}}(C \sqsubseteq D)
\]

\( P_{\mathcal{K}}(C \sqsubseteq D) \) can be computed by

\[
\sum_{\mathcal{I} \models \mathcal{K} \models C \sqsubseteq D} \mathcal{P}_B(\mathcal{W}) \quad \text{where} \quad \mathcal{W}(\kappa)=1
\]

(IJCAR 2014)

and is a PP-complete problem

(JELIA 2014)
Example

Given a \( \mathcal{BEL} \) KB is \( \mathcal{K}_0 = (\mathcal{T}_0, \mathcal{B}_0) \) where \( \mathcal{B}_0 \) is as depicted:

\[
\begin{array}{c|c}
  \bar{x} & y \\
  x & \bar{y} \\
  \bar{y} & 0.5 \\
\end{array}
\]

\[
\begin{array}{c|c}
  x & 0.7 \\
  \bar{x} & y \\
\end{array}
\]

and the contextual TBox \( \mathcal{T}_0 \) given as:

\[
\mathcal{T}_0 := \{ \langle A \sqsubseteq C : \{x, y\} \rangle, \langle A \sqsubseteq B : \{\bar{x}\} \rangle, \langle B \sqsubseteq C : \{\bar{x}\} \rangle \}.
\]
Example

Given a $\mathcal{BEL}$ KB is $\mathcal{K}_0 = (\mathcal{T}_0, \mathcal{B}_0)$ where $\mathcal{B}_0$ is as depicted:

$$
\begin{array}{c|c}
  x & 0.7 \\
  y & \\
\end{array}
$$

and the contextual TBox $\mathcal{T}_0$ given as:

$$
\mathcal{T}_0 := \{ \langle A \sqsubseteq C : \{x, y\} \rangle, \langle A \sqsubseteq B : \{\neg x\} \rangle, \langle B \sqsubseteq C : \{\neg x\} \rangle \}.
$$

Let $A \sqsubseteq C$ be a subsumption query, then $P_{\mathcal{K}}(A \sqsubseteq C) = 1$
Example

Given a $\mathcal{BEL}$ KB is $\mathcal{K}_0 = (\mathcal{T}_0, \mathcal{B}_0)$ where $\mathcal{B}_0$ is as depicted:

<table>
<thead>
<tr>
<th></th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x$</td>
<td>1</td>
</tr>
<tr>
<td>$\neg x$</td>
<td>0.5</td>
</tr>
</tbody>
</table>

and the contextual TBox $\mathcal{T}_0$ given as:

$$\mathcal{T}_0 := \{ \langle A \sqsubseteq C : \{x, y\} \rangle, \langle A \sqsubseteq B : \{\neg x\} \rangle, \langle B \sqsubseteq C : \{\neg x\} \rangle \}.$$ 

Let $A \sqsubseteq C$ be a subsumption query, then $P_{\mathcal{K}}(A \sqsubseteq C) = 1$

This is a well connected TBox, but are real ontologies really well-connected?
The Bayesian Ontology Reasoner BORN

**BORN** first computes a module

Consider **BEL** KB is $\mathcal{K}_0 = (\mathcal{T}_1, \mathcal{B}_0)$ where:

$$\mathcal{T}_1 := \{ \langle A \sqsubseteq C : \{x, y\} \rangle, \langle A \sqsubseteq B : \{\neg x\} \rangle, \langle B \sqsubseteq C : \{\neg x\} \rangle, \langle D \sqsubseteq E : \{\neg y\} \rangle, \langle E \sqsubseteq \exists r.F : \{y\} \rangle, \langle F \sqsubseteq G : \{x, y\} \rangle, \ldots \}$$
The Bayesian Ontology Reasoner BORN

BORN first computes a module

Consider $\mathcal{KB}$ is $\mathcal{K}_0 = (\mathcal{T}_1, \mathcal{B}_0)$ where:

\[
\mathcal{T}_1 := \{ \langle A \sqsubseteq C : \{x, y\} \rangle, \langle A \sqsubseteq B : \{\neg x\} \rangle, \langle B \sqsubseteq C : \{\neg x\} \rangle, \\
\langle D \sqsubseteq E : \{\neg y\} \rangle, \langle E \sqsubseteq \exists r.F : \{y\} \rangle, \langle F \sqsubseteq G : \{x, y\} \rangle \ldots \}
\]

New axioms do not carry information about the query $A \sqsubseteq C$.

The idea is to compute a module w.r.t. the query and the TBox.
The Bayesian Ontology Reasoner BORN

BORN first computes a module

Consider $\mathcal{BEL}$ KB is $\mathcal{K}_0 = (\mathcal{T}_1, \mathcal{B}_0)$ where:

$$\mathcal{T}_1 := \{ \langle A \sqsubseteq C : \{x, y\} \rangle, \langle A \sqsubseteq B : \{-x\} \rangle, \langle B \sqsubseteq C : \{-x\} \rangle$$

$$\langle D \sqsubseteq E : \{-y\} \rangle, \langle E \sqsubseteq \exists r.F : \{y\} \rangle, \langle F \sqsubseteq G : \{x, y\} \rangle, ... \}$$

New axioms do not carry information about the query $A \sqsubseteq C$.

The idea is to compute a module w.r.t. the query and the TBox.

Module of $\mathcal{T}_1$ w.r.t. the query $\{A, C\}$ yields:

$$\mathcal{T}_0 := \{ \langle A \sqsubseteq C : \{x, y\} \rangle, \langle A \sqsubseteq B : \{-x\} \rangle, \langle B \sqsubseteq C : \{-x\} \rangle$$
The Bayesian Ontology Reasoner BORN

BORN converts the knowledge base into a probabilistic logic program:

\[
\text{con('a'). con('b'). role('r').}
\]

\[
\text{subs('a', exists('r', 'b')) :- x0.}
\]

the Bayesian network \( B_0 \) as:

\[
\begin{align*}
.7::x &.1::y:-x. \\
.5::y&:-\neg x. \\
.3::z&:-x,y. \\
.1::z&:-x,\neg y. \\
.0::z&:-\neg x,y. \\
.9::z&:-\neg x,\neg y.
\end{align*}
\]

and the query \( \langle A \sqsubseteq C \rangle \) as:

\[
\text{query(subs('a', 'c')).}
\]

This is the syntax of ProbLog, a probabilistic logic programming tool based on efficient techniques such as weighted model counting, etc...
BORN converts the knowledge base into a probabilistic logic program:

The subsumption $\langle A \sqsubseteq \exists r. B : x \rangle$ is represented as:

$$\text{con('a')}$. con('b'). role('r').
subs('a', exists('r', 'b')) :- x0.$$
The Bayesian Ontology Reasoner BORN

**BORN** converts the knowledge base into a probabilistic logic program:

The subsumption $\langle A \sqsubseteq \exists r.B : x \rangle$ is represented as:

```
con('a'). con('b'). role('r').
subs('a', exists('r', 'b')) :- x0.
```

the Bayesian network $B_0$ as:

```
.7::x.
 1::y:-x.
.5::y:-\+x.
.3::z:-x,y.
.1::z:-\+x,y.
.0::z:-\+x,\+y.
.9::z:-\+x,\+y.
```
**The Bayesian Ontology Reasoner BORN**

**BORN** converts the knowledge base into a probabilistic logic program:

The subsumption \( A \sqsubseteq \exists r.B : x \) is represented as:

\[
\text{con('a')}. \text{con('b')}. \text{role('r')}. \\
\text{subs('a', exists('r', 'b'))} :- x0.
\]

the Bayesian network \( B_0 \) as:

\[
.7::x. \\
1::y:-x. \\
.5::y:-\neg x. \\
.3::z:-x,y. \\
.1::z:-x,\neg y. \\
.0::z:-\neg x,y. \\
.9::z:-\neg x,\neg y.
\]

and the query \( A \sqsubseteq C \) as:

\[
\text{query(subs('a', 'c'))}.
\]
The Bayesian Ontology Reasoner BORN

BORN converts the knowledge base into a probabilistic logic program:

The subsumption $\langle A \sqsubseteq \exists r.B : x \rangle$ is represented as:

$$
\text{con(}'a'\text{'). con(}'b'\text{'). role(}'r'\text{').
subs(}'a'\text{'}, \text{exists(}'r'\text{' ,}'b'\text{') }:- x0.}
$$

the Bayesian network $B_0$ as:

$$
.7::x.
1::y:-x.
.5::y:-\text{\textbackslash }x.
.3::z:-x,y.
.1::z:-x,\text{\textbackslash }y.
.0::z:-\text{\textbackslash }x,y.
.9::z:-\text{\textbackslash }x,\text{\textbackslash }y.
$$

and the query $\langle A \sqsubseteq C \rangle$ as:

$$
\text{query(subs(}'a'\text{' ,}'c'\text{'}).}
$$

This is the syntax of ProbLog, a probabilistic logic programming tool based on efficient techniques such as weighted model counting, etc...
\textit{EL} completion rules added to the program:
**EL completion rules** added to the program:

\[
\text{sub}(X, B) :- \text{subs}(A, B), \text{sub}(X, A), \\
\quad \text{con}(X), \text{con}(A), \text{con}(B).
\]

\[
\text{sub}(X, B) :- \text{subs}(\text{and}(A1, A2), B), \text{sub}(X, A1), \text{sub}(X, A2), \\
\quad \text{con}(X), \text{con}(A1), \text{con}(A2), \text{con}(B).
\]

\[
\text{sub}(X, \exists(R, B)) :- \text{subs}(A, \exists(R, B)), \text{sub}(X, A), \\
\quad \text{con}(X), \text{con}(A), \text{con}(B), \text{role}(R).
\]

\[
\text{sub}(X, B) :- \text{subs}(\exists(R, A), B), \text{sub}(X, \exists(R, Y)), \text{sub}(Y, A), \\
\quad \text{con}(X), \text{con}(Y), \text{con}(A), \text{con}(B), \text{role}(R).
\]
Completion rules added to the program:

\[
\text{sub}(X, B) :- \text{subs}(A, B), \text{sub}(X, A), \\
\quad \text{con}(X), \text{con}(A), \text{con}(B).
\]

\[
\text{sub}(X, B) :- \text{subs}(\text{and}(A1, A2), B), \text{sub}(X, A1), \text{sub}(X, A2), \\
\quad \text{con}(X), \text{con}(A1), \text{con}(A2), \text{con}(B).
\]

\[
\text{sub}(X, \text{exists}(R, B)) :- \text{subs}(A, \text{exists}(R, B)), \text{sub}(X, A), \\
\quad \text{con}(X), \text{con}(A), \text{con}(B), \text{role}(R).
\]

\[
\text{sub}(X, B) :- \text{subs}(\text{exists}(R, A), B), \text{sub}(X, \text{exists}(R, Y)), \text{sub}(Y, A), \\
\quad \text{con}(X), \text{con}(Y), \text{con}(A), \text{con}(B), \text{role}(R).
\]

Inference is performed by ProbLog.
Initial Experiments on Ontologies

<table>
<thead>
<tr>
<th>Ontology</th>
<th>Size of the terminology</th>
<th>Size of the BN</th>
</tr>
</thead>
<tbody>
<tr>
<td>((\text{ABC}, \mathcal{B}_{\text{ABC}}))</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>((\text{ABC}, \mathcal{B}'_{\text{ABC}}))</td>
<td></td>
<td></td>
</tr>
<tr>
<td>((\text{DBPEDIA}, \mathcal{B}_{\text{DBPEDIA}}))</td>
<td>266</td>
<td>17</td>
</tr>
<tr>
<td>((\text{DBPEDIA}, \mathcal{B}'_{\text{DBPEDIA}}))</td>
<td></td>
<td></td>
</tr>
<tr>
<td>((\text{GO}, \mathcal{B}_{\text{GO}}))</td>
<td>23507</td>
<td>200</td>
</tr>
<tr>
<td>((\text{GO}, \mathcal{B}'_{\text{GO}}))</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Primitive Experimental Results

Queries vs Execution time in seconds

(ABC, \mathcal{B}_{ABC})

(ABC, \mathcal{B}'_{ABC})

Preprocessing

Queries

1

2

3

4

5

Execution time in seconds

0.25

0.3

0.35

0.4

6

6

4

2

1

Athens, 6th June 2015

The Bayesian Ontology Reasoner is BORN!
Primitive Experimental Results

Queries Execution time in seconds

(DBPEDIA, $B_{DB}$)
(DBPEDIA, $B'_{DB}$)
Preprocessing

Athens, 6th June 2015 The Bayesian Ontology Reasoner is BORN!
Primitive Experimental Results

![Graph showing execution time in seconds for different queries and ontologies. The x-axis represents queries numbered 1 to 5, while the y-axis represents execution time in seconds. The graph compares three different ontologies: \((\mathcal{G}_0, \mathcal{B}_0)\), \((\mathcal{G}_0, \mathcal{B}_0')\), and \((\mathcal{G}_0, \mathcal{B}_G)\). The preprocessing time is represented by a grey line.]

Athens, 6th June 2015
The Bayesian Ontology Reasoner is BORN!

9
Thanks