Towards a (Model) Theory for Probabilistic Logical Models

Manfred Jaeger

Aalborg University
Progic questions:

*Should probability and logic be combined at all?*
Progic questions:

*Should probability and logic be combined at all?*

Yes.
Progic questions:

*Should probability and logic be combined at all?*

Yes.

*How can probabilistic networks be used to simplify probabilistic logics?*
Progic questions:

*Should probability and logic be combined at all?*

Yes.

*How can probabilistic networks be used to simplify probabilistic logics?*

- How do probabilistic networks relate to probabilistic logic?
- How do “first-order probabilistic networks” relate to first-order logic?
- what are “first-order probabilistic networks”?
"First-order relational probabilistic logic programming models"
"First-order relational probabilistic logic programming models"

$\vdash PL-Models$
Introduction

Logics vs. Models

Semantics of PL-Models

MLNs and RBNs

Expressivity

Complexity

Non-Elementary Inference
Logics vs. Models

The Propositional Case

Logic:

Propositional probabilistic logic (Boole, Nilsson, . . .):

Graphical Models:

Bayesian networks (Lauritzen, Spiegelhalter, Pearl, . . .):
Logics vs. Models

The Propositional Case

**Logic:**

Propositional probabilistic logic (Boole, Nils-son, ...):

Syntax:

\[
P(A \mid B) = 0.4
\]
\[
P(C \mid \neg A \land B) \leq 0.7
\]

... 

**Graphical Models:**

Bayesian networks (Lauritzen, Spiegelhalter, Pearl, ...):

Syntax:

\[
\begin{array}{c}
\text{A} \\
\text{B} \\
\text{C}
\end{array}
\]
Logic:

Propositional probabilistic logic (Boole, Nilsson, ...):

Syntax:

\[
P(A \mid B) = 0.4 \\
P(C \mid \neg A \land B) \leq 0.7 \\
\ldots
\]

Semantics:
Set of probability distributions over possible worlds.

Graphical Models:

Bayesian networks (Lauritzen, Spiegelhalter, Pearl, ...):

Syntax:

Semantics:
Unique probability distribution over possible worlds.
The Propositional Case

Logic:

Propositional probabilistic logic (Boole, Nilsson, ...):

Syntax:

\[ P(A \mid B) = 0.4 \]
\[ P(C \mid \neg A \land B) \leq 0.7 \]
\[ \ldots \]

Semantics:
Set of probability distributions over possible worlds.

Inference:
Theorem proving

Graphical Models:

Bayesian networks (Lauritzen, Spiegelhalter, Pearl, ...):

Syntax:

Semantics:
Unique probability distribution over possible worlds.

Inference:
Model checking

[Halpern, Vardi: Model-checking vs. Theorem proving: a manifesto. 1991]
Logics vs. Models

The Propositional Case

Logic:
Propositional probabilistic logic (Boole, Nilsson, ...):
Syntax:
\[ P(A \mid B) = 0.4 \]
\[ P(C \mid \neg A \land B) \leq 0.7 \]
\[ \ldots \]
Semantics:
Set of probability distributions over possible worlds.
Inference:
Theorem proving

Graphical Models:
Bayesian networks (Lauritzen, Spiegelhalter, Pearl, ...):
Syntax:
\[
\begin{tikzpicture}
  \node (A) at (0,0) {A};
  \node (B) at (-1,-1) {B};
  \node (C) at (1,-1) {C};
  \draw (A) -- (B);
  \draw (B) -- (C);
\end{tikzpicture}
\]
Semantics:
Unique probability distribution over possible worlds.
Inference:
Model checking

[Halpern, Vardi: Model-checking vs. Theorem proving: a manifesto. 1991]

Complexity:
NP-complete

Complexity:
NP-complete
Logic:

First-order probabilistic logic (Halpern, Bacchus, …):

Syntax:
\[ P(\exists x f(x) = a) > 0.6 \]

Semantics:
Set of probability distributions over possible worlds

Inference:
Theorem proving

Complexity:
\( \Pi^1_{\infty} \)-complete
Logics vs. Models

The First-Order Case

Logic:

First-order probabilistic logic (Halpern, Bacchus, ...):

Syntax:

\[ P(\exists x f(x) = a) > 0.6 \]

Semantics:
Set of probability distributions over possible worlds

Inference:
Theorem proving

Complexity:
\( \Pi_1^1 \)-complete

Graphical Models:

PL-models

Syntax:

Semantics:

Inference:
Model checking

Complexity:

\[ \text{Progic-07, Canterbury, Sep.5 2007} \]
Terminology

“Model”...

- in logic: a possible world
- in probabilistic logic: a probability distribution over possible worlds
- in statistics: a (parametric) class of probability distributions over possible worlds
“Model”...

In logic: a possible world
In probabilistic logic: a probability distribution over possible worlds
In statistics: a (parametric) class of probability distributions over possible worlds
Introduction

Logics vs. Models

Semantics of PL-Models

MLNs and RBNs

Expressivity

Complexity

Non-Elementary Inference
“... graphical decision-modeling languages are still quite limited ... while they can describe the relationships among particular event instances, they cannot capture general knowledge about probabilistic relationships across classes of events.” [Breese, Goldman, Wellman 1994]
“... graphical decision-modeling languages are still quite limited ... while they can describe the relationships among particular event instances, they cannot capture general knowledge about probabilistic relationships across classes of events.” [Breese, Goldman, Wellman 1994]
“... graphical decision-modeling languages are still quite limited ... while they can describe the relationships among particular event instances, they cannot capture general knowledge about probabilistic relationships across classes of events.” [Breese, Goldman, Wellman 1994]
Semantics of PL-Models

Probabilistic Logic Model

Possible world $w$

(input domain)

$+$

Probability distribution over expansions of $w$
A PL-model is a mapping

\[ w \mapsto P \]

defined for an (infinite) class of input domains \( w \).
Random Graphs

Markov Chains
Introduction

Logics vs. Models

Semantics of PL-Models

MLNs and RBNs

Expressivity

Complexity

Non-Elementary Inference
Markov Logic Networks [Richardson, Domingos 2006]
Software: http://alchemy.cs.washington.edu/
Markov Logic Networks [Richardson, Domingos 2006]
Software: http://alchemy.cs.washington.edu/

Hard constraints: $\exists x \text{green}(x)$
Markov Logic Networks [Richardson, Domingos 2006]

Software: http://alchemy.cs.washington.edu/

Hard constraints: \( \exists x \ green(x) \)
Weighted formulas: \( green(x) : 0.5 \)
Markov Logic Networks [Richardson,Domingos 2006]
Software: http://alchemy.cs.washington.edu/

Hard constraints: \( \exists x \ green(x) \)
Weighted formulas: 
- \( green(x) : 0.5 \)
- \( green(x) \land edge(x, y) \land \neg green(y) : -1.0 \)
Markov Logic Networks [Richardson, Domingos 2006]
Software: http://alchemy.cs.washington.edu/

Hard constraints:
\[ \exists x \ green(x) \]

Weighted formulas:
\[ \green(x) : 0.5 \]
\[ \green(x) \land edge(x, y) \land \neg \green(y) : -1.0 \]
Relational Bayesian Networks [Jaeger 1997]

Software: http://www.cs.aau.dk/~jaeger/Primula/

logic definition:

$$\exists y (\text{green}(y) \land \text{edge}(y, x))$$
Relational Bayesian Networks [Jaeger 1997]

Software: http://www.cs.aau.dk/~jaeger/Primula/

logic definition:

$$\exists y (\text{green}(y) \land \text{edge}(y, x)) \rightarrow \text{green}(x)$$
Relational Bayesian Networks [Jaeger 1997]

**Software:** http://www.cs.aau.dk/~jaeger/Primula/

**Logic definition:**

\[ \exists y (\text{green}(y) \land \text{edge}(y, x)) \rightarrow \text{green}(x) \]

**Probabilistic definition:**

\[ 0.3 + 0.7 \cdot \text{noisy-or}[0.6].y(\text{green}(y) \land \text{edge}(y, x)) =: P(\text{green}(x)) \]
Relational Bayesian Networks [Jaeger 1997]

Software: [http://www.cs.aau.dk/~jaeger/Primula/](http://www.cs.aau.dk/~jaeger/Primula/)

**logic definition:**

\[ \exists y \left( \text{green}(y) \land \text{edge}(y, x) \right) \iff \text{green}(x) \]

**probabilistic definition:**

\[ 0.3 + 0.7 \cdot \text{noisy-or}[0.6].y \left( \text{green}(y) \land \text{edge}(y, x) \right) =: P(\text{green}(x)) \]
Relational Bayesian Networks [Jaeger 1997]

**Software:** [http://www.cs.aau.dk/~jaeger/Primula/](http://www.cs.aau.dk/~jaeger/Primula/)

**Logic definition:**

\[ \exists y (\text{green}(y) \land \text{edge}(y, x)) \leftrightarrow \text{green}(x) \]

**Probabilistic definition:**

\[ 0.3 + 0.7 \cdot \text{noisy-or}[0.6].y(\text{green}(y) \land \text{edge}(y, x)) =: P(\text{green}(x)) \]
Relational Bayesian Networks [Jaeger 1997]

**Software:** [http://www.cs.aau.dk/~jaeger/Primula/](http://www.cs.aau.dk/~jaeger/Primula/)

![Diagram of RBNs]

**Logic definition:**

\[ \exists y (\text{green}(y) \land \text{edge}(y, x)) \leftarrow \text{green}(x) \]

**Probabilistic definition:**

\[ 0.3 + 0.7 \cdot \text{noisy-or}[0.6].y(\text{green}(y) \land \text{edge}(y, x)) =: P(\text{green}(x)) \]
Relational Bayesian Networks [Jaeger 1997]

Software: [http://www.cs.aau.dk/~jaeger/Primula/](http://www.cs.aau.dk/~jaeger/Primula/)

**logic definition:**

\[ \exists y (\text{green}(y) \land \text{edge}(y, x)) \leftrightarrow \text{green}(x) \]

**probabilistic definition:**

\[ 0.3 + 0.7 \cdot \text{noisy-or}[0.6].y(\text{green}(y) \land \text{edge}(y, x)) =: P(\text{green}(x)) \]
Relational Bayesian Networks [Jaeger 1997]

Software: http://www.cs.aau.dk/~jaeger/Primula/

\[ \exists y (\text{green}(y) \land \text{edge}(y, x)) \leftrightarrow \text{green}(x) \]

\[ 0.3 + 0.7 \cdot \text{noisy-or}[0.6].y(\text{green}(y) \land \text{edge}(y, x)) =: P(\text{green}(x)) \]
Relational Bayesian Networks [Jaeger 1997]

Software: http://www.cs.aau.dk/~jaeger/Primula/

logic definition:

$$\exists y ( \text{green}(y) \land \text{edge}(y, x)) \Leftarrow \text{green}(x)$$

probabilistic definition:

$$0.3 + 0.7 \cdot \text{noisy-or}[0.6].y(\text{green}(y) \land \text{edge}(y, x)) =: P(\text{green}(x))$$

Dependency of ground atoms:

- green(A)
- green(B) → green(C)
### MLN and RBN Comparison

<table>
<thead>
<tr>
<th>MLN</th>
<th>RBN</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Declarative” use of F-O formulas</td>
<td>“Definitional” use of F-O formulas</td>
</tr>
<tr>
<td>Undirected dependencies between ground atoms</td>
<td>Directed dependencies between ground atoms (acyclicity conditions!)</td>
</tr>
<tr>
<td>Best for descriptive/declarative modeling</td>
<td>Best for causal or (incrementally) generative modeling</td>
</tr>
</tbody>
</table>
Introduction

Logics vs. Models

Semantics of PL-Models

MLNs and RBNs

Expressivity

Complexity

Non-Elementary Inference
Expressivity

Formal Framework

General idea:
\( XYZ \) is at least as expressive as \( ABC \) if every PL-model (i.e. mapping \( w \mapsto P \)) definable in \( ABC \) is definable in \( XYZ \).

Issues

- Need to fit the semantics of \( XYZ \), \( ABC \) into PL-model framework
- Asking for identity of models is usually too much – only feasible: \( ABC \)-models can be \textit{embedded} in \( XYZ \)-models.
Every MLN-definable model is *conditionally embedded* in a RBN-definable model:
Every MLN-definable model is *conditionally embedded* in a RBN-definable model:
Every MLN-definable model is \textit{conditionally embedded} in a RBN-definable model:
Every MLN-definable model is *conditionally embedded* in a RBN-definable model:
Every MLN-definable model is *conditionally embedded* in a RBN-definable model:
Every MLN-definable model is \textit{conditionally embedded} in a RBN-definable model:
Introduction

Logics vs. Models

Semantics of PL-Models

MLNs and RBNs

Expressivity

Complexity

Non-Elementary Inference
Elementary Inference Problem

**Given**
- a **model** $M$
- an (input) **domain** (possible world) $D$
- truth assignments for ground atoms in expansions of $D$ (**evidence atoms**):\[
  r_1(d_1) = \tau_1, \ldots, r_n(d_n) = \tau_n
\]
- truth assignments for ground atoms in expansions of $D$ (**query atoms**):\[
  r_{n+1}(d_{n+1}) = \tau_{n+1}, \ldots, r_{n+k}(d_{n+k}) = \tau_n
\]

**compute**
\[
P(r_{n+1}(d_{n+1}) = \tau_{n+1}, \ldots, r_{n+k}(d_{n+k}) \mid r_1(d_1) = \tau_1, \ldots, r_n(d_n) = \tau_n)
\]
for $P$ defined by $M$ on expansions of $D$.
(or **decide** whether $P(\ldots \mid \ldots) > 0$).

**Example:** given that $responding(server003)=false$, what is the probability that $cable_intact(terminal152,server003)=false$?
Complexity issues:

- complexity in terms of complexity of $M$
- complexity in terms of size of $D$
Complexity issues:
- complexity in terms of complexity of $\mathcal{M}$
- complexity in terms of size of $D$

Results:
- Elementary inference (decision version) is $NP$-hard (in size of $D$) for any modeling language with [expressivity requirement]
- ... $NP$-complete for MLNs and RBNs
Boolean satisfiability encoding (cf. [Cooper 1990]):

Boolean function as input domain:

Uniform distribution over consistent value assignments:

Formula satisfiable $\iff P(\text{value(outp)} = \text{true}) > 0$

MLNs and RBNs can encode Boolean satisfiability!
Introduction

Logics vs. Models

Semantics of PL-Models

MLNs and RBNs

Expressivity

Complexity

Non-Elementary Inference
MLN defines a distribution over expansions of $D$ iff

the hard constraints are satisfiable in expansions of $D$

RBN defines a distribution over expansions of $D$ iff

the dependency relation induced by RBN is acyclic

“Verification” Problem:

Given

- a model $\mathcal{M}$
- a class $\mathcal{D}$ of input domains (e.g. axiomatized in some logic)

decide

whether $\mathcal{M}$ defines for all $D \in \mathcal{D}$ a distribution over expansions of $D$. 
\( \mathcal{D}_\emptyset \): finite input domains without relations.

**Result for MLNs**

It is undecidable whether a MLN defines a distribution for each input \( D \in \mathcal{D}_\emptyset \).

**Proof:** it is undecidable whether a first-order sentence \( \phi \) is satisfiable over all finite cardinalities.
\( \mathcal{D}_\emptyset \): finite input domains without relations.

**Result for MLNs**

It is undecidable whether a MLN defines a distribution for each input \( D \in \mathcal{D}_\emptyset \).

**Proof:** it is undecidable whether a first-order sentence \( \phi \) is satisfiable over all finite cardinalities.

**Conjectures for RBNs**

\( \mathcal{D}_{un} \): finite input domains with only unary relations.

It is decidable whether a RBN defines a distribution for each input \( D \in \mathcal{D}_{un} \).

acyclicity of dependencies can be expressed in monadic transitive closure logic

\( \mathcal{D}_{bin} \): finite input domains with one binary relation.

It is undecidable whether a RBN defines a distribution for each input \( D \in \mathcal{D}_{bin} \).
“Global Inference” Problem:

Given

- a model $\mathcal{M}$
- a first-order sentence $\phi$

decide

whether $P(\phi) > 0$ for all input domains for which a distribution is defined by $\mathcal{M}$
“Global Inference” Problem:

Given

- a model $\mathcal{M}$
- a first-order sentence $\phi$

decide

whether $P(\phi) > 0$ for all input domains for which a distribution is defined by $\mathcal{M}$

Results

The global inference problem is undecidable for MLNs and RBNs
How can probabilistic networks be used to simplify probabilistic logics?

They can direct our attention to tractable (model checking) problems!
Summary:

To better understand the plethora of probabilistic logical modeling languages:
- introduced semantic concept of PL-models
- obtained first expressivity result
- obtained some complexity results

Next:
- extend expressivity analysis to other languages (Bayesian Logic Programs, Prism, …)
- investigate learnability issues

Acknowledgements:

Kristian Kersting, Luc De Raedt, …, discussion groups at SRL and Dagstuhl workshops