On the Completeness of First-Order Knowledge Compilation for Lifted Probabilistic Inference

Guy Van den Broeck

NIPS11
December 13, 2011
Outline

- Probabilistic Logic
- Lifted Inference
- Compilation Algorithm
- Completeness
- Conclusions
Outline

- Probabilistic Logic
- Lifted Inference
- Compilation Algorithm
- Completeness
- Conclusions
First-Order Logic

- Example: FOL

- Logical variables have domain of constants
  e.g., X,Y range over domain People = \{alice, bob\}

- Ground formula has no logical variables
  e.g., \text{friends}(alice, bob) \land \text{smokes}(alice) \Rightarrow \text{smokes}(bob)
Probabilistic Logic

- Example: Markov Logic Network (MLN)

Ground atom = random variable in \{true, false\}
  e.g., \text{smokes(alice)}, \text{friends(alice, bob)}

Ground formula = factor in propositional factor graph
Lifted Probabilistic Inference

- Factor graph explodes
- **Propositional** inference is intractable
- **Solution:** lifted inference
  
  - Exploit symmetries
  - Reason at first-order level
  - Reason about groups of objects as a whole
  - Avoid repeated computations
  - Mimic **resolution in theorem proving**

- There is a common understanding but **no formal definition** of lifted inference!
Questions?

- What is commonly understood as lifted inference?
  - Contribution: A formal framework for lifted inference (definition + complexity considerations) ~ PAC-learnability (Valiant)

- When can a model be lifted?
  - Contribution: Extended first-order knowledge compilation
  - Contribution: Completeness result

Take-away message: **Probabilistic models with 2 logical variables per formula are liftable.**
Outline

- Probabilistic Logic
- **Lifted Inference**
- Compilation Algorithm
- Completeness
- Conclusions
### Lifted Inference by First-Order Knowledge Compilation

<table>
<thead>
<tr>
<th></th>
<th>Variable Elimination</th>
<th>Belief Propagation</th>
<th>Knowledge Compilation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Propositional</td>
<td>[Zhang94]</td>
<td>[Pearl82]</td>
<td>[Darwiche03]</td>
</tr>
<tr>
<td>Lifted</td>
<td>[Poole03]</td>
<td>[Singla08]</td>
<td>[VdB11]</td>
</tr>
</tbody>
</table>

- **MLN**
- **Parfactor Graph**
- **WFOMC in FOL**
- **FO d-DNNF Circuit**
- **Evaluate Circuit for Domain**

[Van den Broeck, Guy; Taghipour, Nima; Meert, Wannes; Davis, Jesse; De Raedt, Luc. Lifted probabilistic inference by first-order knowledge compilation, IJCAI11]
Weighted First-Order Model Counting

- A logical theory

Possible worlds
Logical interpretations
Weighted First-Order Model Counting

- A **logical theory**

Interpretations that satisfy the theory

Models
Weighted First-Order Model Counting

- A logical theory and a weight function for predicates

<table>
<thead>
<tr>
<th></th>
<th>theory</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{smokes(\textit{alice})}$</td>
<td>0</td>
<td>$0 \cdot 0 \cdot 1 \cdot 1$</td>
</tr>
<tr>
<td>$\text{smokes(\textit{bob})}$</td>
<td>0</td>
<td>$0 \cdot 0 \cdot 1 \cdot 1$</td>
</tr>
<tr>
<td>$\text{friends(\textit{alice, bob})}$</td>
<td>0</td>
<td>$0 \cdot 0 \cdot 1 \cdot 1$</td>
</tr>
<tr>
<td>$\text{friends(\textit{bob, alice})}$</td>
<td>0</td>
<td>$0 \cdot 0 \cdot 1 \cdot 1$</td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
</tr>
<tr>
<td>$1$</td>
<td>1</td>
<td>$1 \cdot 1 \cdot 1 \cdot 1$</td>
</tr>
<tr>
<td>$0$</td>
<td>0</td>
<td>$0 \cdot 0 \cdot 1 \cdot 1$</td>
</tr>
<tr>
<td>$1$</td>
<td>1</td>
<td>$1 \cdot 1 \cdot 1 \cdot 1$</td>
</tr>
<tr>
<td>$1$</td>
<td>1</td>
<td>$1 \cdot 1 \cdot 1 \cdot 1$</td>
</tr>
</tbody>
</table>
Weighted First-Order Model Counting

- A **logical theory** and a **weight function** for predicates

<table>
<thead>
<tr>
<th></th>
<th>smokes(alice)</th>
<th>smokes(bob)</th>
<th>friends(alice, bob)</th>
<th>friends(bob, alice)</th>
<th>theory</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2 \cdot 2 \cdot 1 \cdot 1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1 \cdot 1 \cdot 4 \cdot 4</td>
</tr>
</tbody>
</table>

\[ \sum \text{Weighted first-order model count} \]

\[ \sim \text{Partition function} \]
Lifted Inference by First-Order Knowledge Compilation

MLN → WFOMC in FOL → FO d-DNNF Circuit → Evaluate Circuit for Domain
Lifted Inference by First-Order Knowledge Compilation

MLN

\[ 2 \text{ friends}(X, Y) \land \text{smokes}(X) \Rightarrow \text{smokes}(Y) \]
Lifted Inference by First-Order Knowledge Compilation

MLN \rightarrow \text{WFOMC in FOL}

\text{2 friends}(X, Y) \land \text{smokes}(X) \Rightarrow \text{smokes}(Y)

\text{WFOMC in FOL equivalent to partition function of MLN}

\text{smokes}(Y) \lor \neg \text{smokes}(X)
\lor \neg \text{friends}(X, Y) \lor \neg f(X, Y)
\text{friends}(X, Y) \lor f(X, Y)
\text{smokes}(X) \lor f(X, Y)
\neg \text{smokes}(Y) \lor f(X, Y).

<table>
<thead>
<tr>
<th>Predicate</th>
<th>w</th>
<th>\bar{w}</th>
</tr>
</thead>
<tbody>
<tr>
<td>friends</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>smokes</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>f</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>f</td>
<td>\epsilon^2</td>
<td>1</td>
</tr>
</tbody>
</table>
Lifted Inference by First-Order Knowledge Compilation

MLN → WFOMC in FOL → d-DNNF Circuit → Evaluate Circuit

Ground to propositional logic

Logical d-DNNF circuit

Inducing an arithmetic circuit

\[
\begin{align*}
\text{smokes}(Y) \lor \neg \text{smokes}(X) \\
\lor \neg \text{friends}(X, Y) \lor \neg f(X, Y) \\
\text{friends}(X, Y) \lor f(X, Y) \\
\text{smokes}(X) \lor f(X, Y) \\
\neg \text{smokes}(Y) \lor f(X, Y).
\end{align*}
\]

<table>
<thead>
<tr>
<th>Predicate</th>
<th>(w)</th>
<th>(\overline{w})</th>
</tr>
</thead>
<tbody>
<tr>
<td>friends</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>smokes</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>(f)</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Circuit for domain \{alice\}

Circuit for domain \{alice, bob\}

Circuit for domain \{alice, bob, charlie\}
Lifted Inference by First-Order Knowledge Compilation

**First-Order d-DNNF circuit**

- Independent of domain size

**Example Predicate Formulas**

\[
\text{smokes}(Y) \vee \neg \text{smokes}(X) \\
\neg \text{friends}(X, Y) \vee \neg f(X, Y) \\
\text{friends}(X, Y) \vee f(X, Y) \\
\text{smokes}(X) \vee f(X, Y) \\
\neg \text{smokes}(Y) \vee f(X, Y).
\]

**Predicate Table**

<table>
<thead>
<tr>
<th>Predicate</th>
<th>$w$</th>
<th>$\overline{w}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>friends</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>smokes</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$f$</td>
<td>$e^2$</td>
<td>1</td>
</tr>
</tbody>
</table>
Lifted Inference by First-Order Knowledge Compilation

MLN → WFOMC in FOL → FO d-DNNF Circuit → Evaluate Circuit for Domain
Outline

- Probabilistic Logic
- Lifted Inference
- **Compilation Algorithm**
- Completeness
- Conclusions
Compilation Algorithm CR1 [VdB11]

- 6 compilation rules:
  - Input: FO logic theory; Output: FO d-DNNF circuit
  - Compilation rule recursively compiles 'simpler' theories
- Example: Independence compilation rule

\[
\neg \text{friends}(bob, X) \\
\text{smokes}(X) \Rightarrow \neg \text{friends}(alice, X)
\]
Compilation Algorithm CR1 [VdB11]

- 6 compilation rules:
  - Input: FO logic theory; Output: FO d-DNNF circuit
  - Compilation rule recursively compiles 'simpler' theories
- Example: Independence compilation rule

\[
\neg \text{friends}(bob, X) \\
\text{smokes}(X) \Rightarrow \neg \text{friends}(alice, X)
\]
New Rule: Domain Recursion

- Example theory: \( \text{friends}(X, Y) \Rightarrow \text{friends}(Y, X) \)
- Split up domain \textbf{People} into \{c\} U \textbf{People}'
- Split up theory into 3 independent subtheories
New Rule: Domain Recursion

- Example theory: $\text{friends}(X, Y) \Rightarrow \text{friends}(Y, X)$
- Split up domain **People** into $\{c\} \cup \text{People}'$
- Split up theory into 3 independent subtheories
  1) where $X=c$ and $Y=c$: $\text{friends}(c, c) \Rightarrow \text{friends}(c, c)$
New Rule: Domain Recursion

- **Example theory:** \( \text{friends}(X, Y) \Rightarrow \text{friends}(Y, X) \)
- **Split up domain** People **into** \{c\} U People'
- **Split up theory into 3 independent subtheories**
  1) where \( X = c \) and \( Y = c \):
     \( \text{friends}(c, c) \Rightarrow \text{friends}(c, c) \)
  2) where \( X \neq c \) and \( Y \neq c \):
     \( \text{friends}(X, Y) \Rightarrow \text{friends}(Y, X), X \neq c \land Y \neq c \)
New Rule: Domain Recursion

- Example theory: $\text{friends}(X,Y) \Rightarrow \text{friends}(Y,X)$
- Split up domain People into $\{c\} \cup \text{People'}$
- Split up theory into 3 independent subtheories:
  1) where $X=c$ and $Y=c$: $\text{friends}(c,c) \Rightarrow \text{friends}(c,c)$
  2) where $X \neq c$ and $Y \neq c$: $\text{friends}(X,Y) \Rightarrow \text{friends}(Y,X), X \neq c \land Y \neq c$
  3) where $(X \neq c$ and $Y=c)$ or $(X=c$ and $Y \neq c)$:
     $\text{friends}(c,Y) \Rightarrow \text{friends}(Y,c), Y \neq c$
     $\text{friends}(X,c) \Rightarrow \text{friends}(c,X), X \neq c$
Experiments

- **c2d**: Propositional knowledge compilation
- **CR1**: Existing FO knowledge compilation
- **CR2**: CR1 with domain recursion

The graph shows the runtime in seconds as a function of the number of people. The runtime increases with the number of people, with **c2d** being the fastest and **CR2** showing the slowest growth.
Outline

- Probabilistic Logic
- Lifted Inference
- Compilation Algorithm
- Completeness
- Conclusions
Definition:
Complexity of computing $P(q|e)$ in model $m$ is \textbf{polynomial} time in the \textbf{domain sizes} of the logical variables in $q,e,m$.

Possibly exponential in the size of $q,e,m$

- \# predicates, \# parfactors, \# atoms,
- \# arguments, \# formulas, \# constants in model

Motivation: Large domains lead to intractable propositional inference.
Completeness

A procedure that is domain-lifted for all models in a class $M$ is called \textbf{complete} for $M$.

All models in $M$ are “liftable”

\textbf{No completeness result} so far for existing algorithms.

\textit{If you give me a model, I cannot say if grounding will be needed, until I run the inference algorithm itself.}
Completeness of CR1 and CR2

- **Definition:**
  - \(k\)-WFOMC consists of WFOMC theories with up to \(k\) logical variables per formula

- **Theorem:**
  - CR1 is complete domain-lifted for 1-WFOMC
    ... but not for e.g.,
    \[
    \begin{align*}
    \text{friends}(X, Y) & \Rightarrow \text{friends}(Y, X) \\
    \text{parent}(X, Y) & \Rightarrow \neg \text{parent}(Y, X), \ X \neq Y, \\
    \leq (X, Y) & \lor \leq (Y, X)
    \end{align*}
    \]

- **Theorem:**
  - CR2 is complete domain-lifted for 2-WFOMC
Importance of Completeness Results

- These are **sufficient** conditions for domain-lifted inference ("liftability")
- **First completeness result** so far for lifted probabilistic inference
- 2-WFOMC is a **non-trivial** class of models
  - (anti-)symmetric, total relations are useful concepts
  - CR1 could already lift more than previous methods
  - CR2 can lift even more, now all of 2-WFOMC
- Open question: other classes?
Outline

- Probabilistic Logic
- Lifted Inference
- Compilation Algorithm
- Completeness
- Conclusions
Conclusions

3 contributions:

1) A formal framework for lifted inference (definition + complexity considerations)
2) New compilation rule for first-order knowledge compilation
3) First completeness result in lifted inference

Take-away message: 2-WFOMC is liftable. This is the first non-trivial class of problems.
On the Completeness of First-Order Knowledge Compilation for Lifted Probabilistic Inference

Guy Van den Broeck

Compilation Algorithm
- There are 6 existing compilation rules:
  - Input: logical theory Output: FO-DONE Circuit
  - Compilation rules recursively compile "simpler" theories
- We add a 7th: domain recursion
- Experiments show improvement
  - call propositional knowledge compilation
  - FO: existing FO knowledge compilation
  - CR1, CR2 w.r.t. domain recursion

Completeness
- Definition: Domain-Lifted Probabilistic Inference
  The complexity of computing $P(c|e)$ in model $m$ is:
  - Polynomial time in the domain sizes of the logical variables in $e$,$a$,$m$
  - Possibly exponential in the size of $e$,$a$,$m$
- Definition: A procedure that is domain-lifted for all models in a class $M$ is called complete for $M$.
  Domain-lifted inference is "solved" for $M$.
- There is no completeness result for inference methods:
  - If you give me a model, I cannot say if grounding will be needed, until run the inference algorithm first.
- Definition: k-WFOMC consists of WFOMC theories with up to k logical variables per clause:
  - Theorem: CR1 is complete domain-lifted for 1-WFOMC ... but not for (anti-)symmetric and total relations:
  - Theorem: CR2 is complete domain-lifted for 2-WFOMC

Conclusions
- Sufficient conditions for domain-lifted inference
- First completeness result for lifted probabilistic inference
- 2-WFOMC is a non-trivial class of models and (anti-)symmetric, total relations are useful concepts
- 3 main contributions:
  1) A formal framework for lifted inference with a definition in terms of complexity considerations
  2) New compilation rule for first-order knowledge compilation
  3) New algorithm is a complete domain-lifted probabilistic inference algorithm
- Lifted inference is "solved" for 2-WFOMC, a first non-trivial class of problems.

First-Order Knowledge Compilation
- Lifted version of knowledge compilation
- Reduce probabilistic inference to WFOMC in logic
- Compile probabilistic model into a logical circuit where WFOMC inference is efficient (polynomial)

Poster today!
Website & Implementation: http://dtaic.cs.kuleuven.be/ml/systems/wfomc