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

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

[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

Friends model interpretations that satisfy the theory:

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

- Models
Weighted First-Order Model Counting

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

<table>
<thead>
<tr>
<th></th>
<th>alice</th>
<th>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>0</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>
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>0 0 0 0</td>
<td>1</td>
<td>2 • 2 • 1 • 1</td>
</tr>
<tr>
<td>... ... ... ...</td>
<td>... ... ... ...</td>
<td>... ... ... ...</td>
</tr>
<tr>
<td>1 0 1 0 0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>... ... ... ...</td>
<td>... ... ... ...</td>
<td>... ... ... ...</td>
</tr>
<tr>
<td>1 1 1 1 1</td>
<td>1</td>
<td>1 • 1 • 4 • 4</td>
</tr>
</tbody>
</table>

\[ \sum \]

Weighted first-order model count
～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 \xrightarrow{\text{WFOMC in FOL}} \text{WFOMC in FOL}

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

\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 \cdot e^2</td>
<td>1</td>
</tr>
</tbody>
</table>
Lifted Inference by First-Order Knowledge Compilation

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>$\bar{w}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>friends</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>smokers</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$f$</td>
<td>$e^2$</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

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

<table>
<thead>
<tr>
<th>Predicate</th>
<th>( w )</th>
<th>( \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>( e^2 )</td>
<td>1</td>
</tr>
</tbody>
</table>

Evaluate Circuit for Domain
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

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

- Example theory: \( \text{friends}(X, Y) \Rightarrow \text{friends}(Y, X) \)
- Split up domain People into \( \{c\} \cup \text{People'} \)
- 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 $\{c\} \cup \text{People}'$
- 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 $Y\neq c$: $\text{friends}(X, Y) \Rightarrow \text{friends}(Y, X), X \neq c \land Y \neq c$
New Rule: Domain Recursion

- Example theory: \[ \text{friends}(X, Y) \Rightarrow \text{friends}(Y, X) \]
- Split up domain \textbf{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 \(Y\neq c\):
     \[ \text{friends}(X, Y) \Rightarrow \text{friends}(Y, X), X \neq c \land Y \neq c \]
  3) where \((X\neq c\) and \(Y=c\)) or \((X=c\) and \(Y\neq c\)):
     \[
     \begin{align*}
     \text{friends}(c, Y) & \Rightarrow \text{friends}(Y, c), Y \neq c \\
     \text{friends}(X, c) & \Rightarrow \text{friends}(c, X), X \neq c
     \end{align*}
     \]
Experiments

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

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

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 *complete* for $M$.

*All models in $M$ are “liftable”*

No completeness result so far for existing algorithms.

*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., $\text{friends}(X, Y) \Rightarrow \text{friends}(Y, X)$, $\text{parent}(X, Y) \Rightarrow \neg \text{parent}(Y, X)$, $X \neq Y$, $\leq (X, Y) \lor \leq (Y, X)$

- **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

Website & Implementation: http://dtai.cs.kuleuven.be/ml/systems/wfomc

Poster today!

Probabilistic Logic
- Logic with Probabilities: e.g. Markov Logic
- Weight: Probability
- Formula in First-Order Logic
- 2 Friends(X, Y) \land \text{smokes}(X) \Rightarrow \text{smokes}(Y)

Domain of constants: e.g. X \in \{ak⚡️, bob\}

Literals Variable
Random variable in (true/false) for each X

* Represents factor graph for given domain (aka, test)

Lifted Inference
- Factor graph explodes: e.g., 10 parents, 2500 factor variables
- Propositional inference is intractable
- Solution: Lifted inference

Can we talk about groups of objects as whole?
- Avoid repeated computations
- Simpler reasoning

Complete Symmetrical
Explicit Symmetry

However, there is no formal framework (similar to PAC for learning)

Research Questions
- What is commonly understood as lifted inference?
  - Contribution: A formal framework for lifted inference (definition + complexity considerations) = PAC
  - When can a model be lifted?
    - Contribution: Extended first-order knowledge compilation with a new operator
    - Contribution: Completeness result

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)

Compilation Algorithm
- There are 6 existing compilation rules:
  - Input: logical theory, Output: FO-DNNF Circuit
    - Compilation rules successively compile "simpler" theories
  - We add a 7th: domain recursion

- Experiments show improvement

- cf. propositional knowledge compilation
- CR1 = existing FO knowledge compilation
- CR2 = CR1 w/ domain recursion

Completeness
- Definition: Domain-Lifted Probabilistic Inference
  - Complexity of computing P(x|e) in model M is
    - Polynomial time in the domain sizes of the logical variables i.e. \#e
    - Possibly exponential in the size of \#e
  - Definition: A procedure that is domain-lifted for all models in a class is called \textit{complete} for M.

  - There is no completeness result for lifted inference methods.
    - If you give me a model, I cannot say if grounding will be needed, until run the inference algorithm fast.

- Definition: \textbf{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.