On First-Order Knowledge Compilation

Guy Van den Broeck

UCLA

Beyond NP Workshop
Feb 12, 2016
Overview

1. Why first-order model counting?
2. Why first-order model counters?
3. What first-order circuit languages?
4. How first-order knowledge compilation?
5. Perspectives …
Why do we need first-order model counting?
Uncertainty in AI

Probability Distribution

= Qualitative + Quantitative
Probabilistic Graphical Models

Probability Distribution = Graph Structure + Parameterization
Probabilistic Graphical Models

Probability Distribution = Graph Structure + Parameterization

| rain | Pr(sun | rain) |
|------|---------|
| T    | 0.1     |
| F    | 0.6     |

<table>
<thead>
<tr>
<th>rain</th>
<th>sun</th>
<th>Pr(rainbow)</th>
<th>rain, sun</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>0.9</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>0.05</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>0.05</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Pr(rain)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
</tr>
</tbody>
</table>
Weighted Model Counting

Probability Distribution

= SAT Formula + Weights

[Chavira 2008, Sang 2005]
Weighted Model Counting

Probability Distribution

\[ \text{SAT Formula} + \text{Weights} \]

Rain \Rightarrow \text{Cloudy}

\[ \text{Sun} \land \text{Rain} \Rightarrow \text{Rainbow} \]

\[ w(\text{Rain}) = 1 \]
\[ w(\neg\text{Rain}) = 2 \]
\[ w(\text{Cloudy}) = 3 \]
\[ w(\neg\text{Cloudy}) = 5 \]

\[ \ldots \]

[Chavira 2008, Sang 2005]
Beyond NP Pipeline for #P

Generalized Perspective

Probability Distribution

= Logic + Weights
Generalized Perspective

Probability Distribution

= Logic

+ Weights

Logical Syntax Model-theoretic Semantics

+ Weight function $w(.)$

Factorized

$Pr(model) \propto \prod_i w(x_i)$
First-Order Model Counting

Probability Distribution = First-Order Logic + Weights

[Van den Broeck 2011, 2013, Gogate 2011]
First-Order Model Counting

Probability Distribution

= First-Order Logic

+ Weights

Smokes(x) \land \text{Friends}(x,y) \Rightarrow \text{Smokes}(y)

+ 

w(\text{Smokes}(a))=1

w(\neg\text{Smokes}(a))=2

w(\text{Smokes}(b))=1

w(\neg\text{Smokes}(b))=2

w(\text{Friends}(a,b))=3

w(\neg\text{Friends}(a,b))=5

... 

[Van den Broeck 2011, 2013, Gogate 2011]
Probabilistic Programming

Probability Distribution

\[ \equiv \]

Logic Programs

\[ + \]

Weights

[Fierens 2015]
Probabilistic Programming

Probability Distribution

\[ \text{Logic Programs} \]

+ Weights

\[
\begin{align*}
\text{path}(X,Y) & : \neg \text{edge}(X,Y). \\
\text{path}(X,Y) & : \neg \text{edge}(X,Z), \text{path}(Z,Y).
\end{align*}
\]
Weighted Model Integration

\[ \text{Probability Distribution} = \text{SMT}(\text{LRA}) + \text{Weights} \]

[Belle 2015]
Weighted Model Integration

Probability Distribution

\[ \text{SMT(LRA)} \]

+ \text{Weights}

0 \leq \text{height} \leq 200
0 \leq \text{weight} \leq 200
0 \leq \text{age} \leq 100
\text{age} < 1 \Rightarrow \text{height+weight} \leq 90

\[ w(\text{height}) = \text{height}-10 \]
\[ w(\neg \text{height}) = 3\times \text{height}^2 \]
\[ w(\neg \text{weight}) = 5 \]

[\text{Belle 2015}]
Beyond NP Pipeline for $\#P/\#P_1$

- Parfactor graphs
- Probabilistic logic programs
- Markov Logic
- Relational Bayesian networks
- Probabilistic databases

Weighted First-Order Model Counting

First-Order Model Counting

Model = solution to first-order logic formula $\Delta$

$\Delta = \forall d \ (\text{Rain}(d) \Rightarrow \text{Cloudy}(d))$

Days = \{Monday\}
First-Order Model Counting

Model = solution to first-order logic formula $\Delta$

$\Delta = \forall d \ (\text{Rain}(d) \Rightarrow \text{Cloudy}(d))$

Days = {Monday}

<table>
<thead>
<tr>
<th>Rain(M)</th>
<th>Cloudy(M)</th>
<th>Model?</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>Yes</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>Yes</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>Yes</td>
</tr>
</tbody>
</table>

$\text{FOMC} = 3$
Weighted First-Order Model Counting

Model = solution to first-order logic formula $\Delta$

$\Delta = \forall d \ (\text{Rain}(d) \Rightarrow \text{Cloudy}(d))$

<table>
<thead>
<tr>
<th>Days = {Monday, Tuesday}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Days = {Monday, Tuesday}</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rain(M)</th>
<th>Cloudy(M)</th>
<th>Rain(T)</th>
<th>Cloudy(T)</th>
<th>Model?</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>Yes</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>No</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>Yes</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>Yes</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>Yes</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>No</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>Yes</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>Yes</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>Yes</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>Yes</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Weighted First-Order Model Counting

Model = solution to first-order logic formula $\Delta$

$$\Delta = \forall d \ (\text{Rain}(d) \Rightarrow \text{Cloudy}(d))$$

Days = {Monday, Tuesday}

<table>
<thead>
<tr>
<th>Rain(M)</th>
<th>Cloudy(M)</th>
<th>Rain(T)</th>
<th>Cloudy(T)</th>
<th>Model?</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>Yes</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>No</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>Yes</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>Yes</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>Yes</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>No</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>Yes</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>Yes</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>Yes</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>Yes</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>Yes</td>
</tr>
</tbody>
</table>

$\#\text{SAT} = 9$
Weighted First-Order Model Counting

Model = solution to first-order logic formula $\Delta$

$$\Delta = \forall d \ (\text{Rain}(d) \Rightarrow \text{Cloudy}(d))$$

Days = {Monday, Tuesday}

<table>
<thead>
<tr>
<th>Rain(M)</th>
<th>Cloudy(M)</th>
<th>Rain(T)</th>
<th>Cloudy(T)</th>
<th>Model?</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>Yes</td>
<td>9</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>No</td>
<td>0</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>Yes</td>
<td>18</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>Yes</td>
<td>30</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>No</td>
<td>0</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>No</td>
<td>0</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>No</td>
<td>0</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>No</td>
<td>0</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>Yes</td>
<td>18</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>No</td>
<td>0</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>Yes</td>
<td>36</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>Yes</td>
<td>60</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>Yes</td>
<td>30</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>No</td>
<td>0</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>Yes</td>
<td>60</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>Yes</td>
<td>100</td>
</tr>
</tbody>
</table>

$\# \text{SAT} = 9$
# Weighted First-Order Model Counting

Model = solution to **first-order** logic formula $\Delta$

## Delta

$\Delta = \forall d \ (\text{Rain}(d) \Rightarrow \text{Cloudy}(d))$

## Days

Days = \{Monday, Tuesday\}

## Weights

\- $w(\text{R}) = 1$
\- $w(\neg \text{R}) = 2$
\- $w(\text{C}) = 3$
\- $w(\neg \text{C}) = 5$

## Model Evaluation

<table>
<thead>
<tr>
<th>Rain(M)</th>
<th>Cloudy(M)</th>
<th>Rain(T)</th>
<th>Cloudy(T)</th>
<th>Model?</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>Yes</td>
<td>9</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>No</td>
<td>0</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>Yes</td>
<td>18</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>Yes</td>
<td>30</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>No</td>
<td>0</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>No</td>
<td>0</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>No</td>
<td>0</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>No</td>
<td>0</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>Yes</td>
<td>18</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>No</td>
<td>0</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>Yes</td>
<td>36</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>Yes</td>
<td>60</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>Yes</td>
<td>30</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>No</td>
<td>0</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>Yes</td>
<td>60</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>Yes</td>
<td>100</td>
</tr>
</tbody>
</table>

$\#\text{SAT} = 9$

$\text{WFOMC} = 361$
Why do we need first-order model counters?
A Simple Reasoning Problem

- 52 playing cards
- Let us ask some simple questions

[Van den Broeck 2015]
A Simple Reasoning Problem

Probability that Card1 is Hearts?
A Simple Reasoning Problem

Probability that Card1 is Hearts? 1/4

[Van den Broeck 2015]
A Simple Reasoning Problem

Probability that Card1 is Hearts given that Card1 is red?

[Van den Broeck 2015]
A Simple Reasoning Problem

Probability that Card1 is Hearts given that Card1 is red? 1/2
A Simple Reasoning Problem

Probability that Card52 is Spades given that Card1 is QH?

[Van den Broeck 2015]
A Simple Reasoning Problem

Probability that Card52 is Spades given that Card1 is QH? 13/51

[Van den Broeck 2015]
A Simple Reasoning Problem

Probability that Card1 is Hearts?
A Simple Reasoning Problem

Probability that Card1 is Hearts?  1/4

[Van den Broeck 2015]
A Simple Reasoning Problem

Probability that Card52 is Spades given that Card1 is QH?

[Van den Broeck 2015]
A Simple Reasoning Problem

Probability that Card52 is Spades given that Card1 is QH? 13/51

[Van den Broeck 2015]
Model distribution by FOMC:

\[ \Delta = \forall p, \exists c, \text{Card}(p,c) \]
\[ \forall c, \exists p, \text{Card}(p,c) \]
\[ \forall p, \forall c, \forall c', \text{Card}(p,c) \land \text{Card}(p,c') \Rightarrow c = c' \]

[Van den Broeck 2015]
Beyond NP Pipeline for #P

Reduce to propositional model counting:

[Van den Broeck 2015]
Beyond NP Pipeline for #P

Reduce to propositional model counting:

\[ \Delta = \text{Card}(A\heartsuit, p_1) \lor \ldots \lor \text{Card}(2\spadesuit, p_1) \]
\[ \text{Card}(A\heartsuit, p_2) \lor \ldots \lor \text{Card}(2\spadesuit, p_2) \]
\[ \ldots \]
\[ \text{Card}(A\heartsuit, p_1) \lor \ldots \lor \text{Card}(A\heartsuit, p_{52}) \]
\[ \text{Card}(K\heartsuit, p_1) \lor \ldots \lor \text{Card}(K\heartsuit, p_{52}) \]
\[ \ldots \]
\[ \neg \text{Card}(A\heartsuit, p_1) \lor \neg \text{Card}(A\heartsuit, p_2) \]
\[ \neg \text{Card}(A\heartsuit, p_1) \lor \neg \text{Card}(A\heartsuit, p_3) \]
\[ \ldots \]

[Van den Broeck 2015]
Beyond NP Pipeline for \#P

Reduce to propositional model counting:

$$\Delta = \text{Card}(A\heartsuit, p_1) \lor \ldots \lor \text{Card}(2\spadesuit, p_1)$$
$$\text{Card}(A\heartsuit, p_2) \lor \ldots \lor \text{Card}(2\spadesuit, p_2)$$
$$\ldots$$
$$\text{Card}(A\heartsuit, p_1) \lor \ldots \lor \text{Card}(A\heartsuit, p_{52})$$
$$\text{Card}(K\heartsuit, p_1) \lor \ldots \lor \text{Card}(K\heartsuit, p_{52})$$
$$\ldots$$
$$\neg\text{Card}(A\heartsuit, p_1) \lor \neg\text{Card}(A\heartsuit, p_2)$$
$$\neg\text{Card}(A\heartsuit, p_1) \lor \neg\text{Card}(A\heartsuit, p_3)$$
$$\ldots$$

What will happen?

[Van den Broeck 2015]
Deck of Cards Graphically

[Van den Broeck 2015]
Deck of Cards Graphically

One model/\textit{perfect matching}

[Van den Broeck 2015]
Deck of Cards Graphically

[Van den Broeck 2015]
Deck of Cards Graphically

Card(K♥, p_{52})

[Van den Broeck 2015]
Deck of Cards Graphically

Model counting: How many perfect matchings?

Card(K♥,p_{52})
Deck of Cards Graphically

[Van den Broeck 2015]
What if I add the unit clause 
\( \neg \text{Card}(K\heartsuit, p_{52}) \) to my CNF?
Deck of Cards Graphically

What if I add the unit clause \(-\text{Card(K♥,p}_{52})\) to my CNF?
Deck of Cards Graphically

What if I add unit clauses to my CNF?
Observations

• Deck of cards = complete bigraph
• Unit clause removes edge
  Encode any bigraph

• Counting models = perfect matchings
• Problem is \#P-complete! 😞

• All solvers efficiently handle unit clauses
• No solver can do cards problem efficiently!

[Van den Broeck 2015]
What's Going On Here?

Probability that Card52 is Spades given that Card1 is QH?
What's Going On Here?

Probability that Card52 is Spades given that Card1 is QH? 13/51

[Van den Broeck 2015]
What's Going On Here?

Probability that Card52 is Spades given that Card2 is QH?
What's Going On Here?

Probability that Card52 is Spades given that Card2 is QH? 13/51

[Van den Broeck 2015]
What's Going On Here?

Probability that Card52 is Spades given that Card3 is QH?
What's Going On Here?

Probability that Card52 is Spades given that Card3 is QH?

13/51

[Van den Broeck 2015]
Tractable Reasoning

What's going on here?
Which property makes reasoning tractable?

[Niepert 2014, Van den Broeck 2015]
Tractable Reasoning

What's going on here?
Which property makes reasoning tractable?

- High-level (first-order) reasoning
- Symmetry
- Exchangeability

⇒ Lifted Inference

[Niepert 2014, Van den Broeck 2015]
What are first-order circuit languages?
Negation Normal Form

[Darwiche 2002]
Decomposable NNF

[Darwiche 2002]
Deterministic Decomposable NNF

[Darwiche 2002]
Deterministic Decomposable NNF

Weighted Model Counting

[Darwiche 2002]
Deterministic Decomposable NNF

Weighted Model Counting and much more!

[Darwiche 2002]
First-Order NNF

∀X, X ∈ People : belgian(X) ⇒ likes(X, chocolate)
First-Order Decomposability

\[ \forall X, X \in \text{People} : \text{belgian}(X) \Rightarrow \text{likes}(X, \text{chocolate}) \]
First-Order Decomposability

\[ \forall X, X \in \text{People} : \text{belgian}(X) \Rightarrow \text{likes}(X, \text{chocolate}) \]
First-Order Determinism

$$\forall X, X \in \text{People} : \text{belgian}(X) \implies \text{likes}(X, \text{chocolate})$$
Deterministic Decomposable FO NNF

∀X, X ∈ People : belgian(X) ⇒ likes(X, chocolate)

Weighted Model Counting

[Van den Broeck 2013]
Deterministic Decomposable FO NNF

\[ \forall X, X \in \text{People} : \text{belgian}(X) \Rightarrow \text{likes}(X, chocolate) \]

Weighted Model Counting

\[ \Pr(\text{belgian}) \times \Pr(\text{likes}) + \Pr(\neg \text{belgian}) \]

[Van den Broeck 2013]
Deterministic Decomposable FO NNF

\[ \forall X, X \in \text{People} : \text{belgian}(X) \Rightarrow \text{likes}(X, \text{chocolate}) \]

Weighted Model Counting

\[ |\text{People}| \left( \Pr(\text{belgian}) \times \Pr(\text{likes}) + \Pr(\neg\text{belgian}) \right) \]

[Van den Broeck 2013]
How to do first-order knowledge compilation?
Deterministic Decomposable FO NNF

\[ \Delta = \forall x, y \in \text{People}, (\text{Smokes}(x) \land \text{Friends}(x,y) \Rightarrow \text{Smokes}(y)) \]

[Van den Broeck 2013]
$\Delta = \forall x, y \in \text{People}, (\text{Smokes}(x) \land \text{Friends}(x,y) \Rightarrow \text{Smokes}(y))$
Deterministic Decomposable FO NNF

$\Delta = \forall x, y \in \text{People}, (\text{Smokes}(x) \land \text{Friends}(x,y) \Rightarrow \text{Smokes}(y))$

[Van den Broeck 2013]
Deterministic Decomposable FO NNF

\[ \Delta = \forall x, y \in \text{People}, (\text{Smokes}(x) \land \text{Friends}(x, y) \Rightarrow \text{Smokes}(y)) \]
Deterministic Decomposable FO NNF

$$\Delta = \forall x, y \in \text{People}, \ (\text{Smokes}(x) \land \text{Friends}(x, y) \Rightarrow \text{Smokes}(y))$$

[Van den Broeck 2013]
Deterministic Decomposable FO NNF

\[ \Delta = \forall x, y \in \text{People}, \ (\text{Smokes}(x) \land \text{Friends}(x,y) \Rightarrow \text{Smokes}(y)) \]
First-Order Model Counting: Example

\[ \Delta = \forall x, y \in \text{People}, (\text{Smokes}(x) \land \text{Friends}(x,y) \Rightarrow \text{Smokes}(y)) \]
First-Order Model Counting: Example

\[ \Delta = \forall x, y \in \text{People}, \ (\text{Smokes}(x) \land \text{Friends}(x,y) \Rightarrow \text{Smokes}(y)) \]

- If we know \( D \) precisely: who smokes, and there are \( k \) smokers?

**Database:**
- Smokes(Alice) = 1
- Smokes(Bob) = 0
- Smokes(Charlie) = 0
- Smokes(Dave) = 1
- Smokes(Eve) = 0
- ... 

\[
\begin{align*}
\text{Smokes} & \quad \text{Friends} & \quad \text{Smokes} \\
\text{k} & \quad & \text{k} \\
\text{n-k} & \quad & \text{n-k}
\end{align*}
\]
First-Order Model Counting: Example

\[ \Delta = \forall x, y \in \text{People}, (\text{Smokes}(x) \land \text{Friends}(x, y) \Rightarrow \text{Smokes}(y)) \]

- If we know \( D \) precisely: who smokes, and there are \( k \) smokers?

**Database:**
- \( \text{Smokes}(\text{Alice}) = 1 \)
- \( \text{Smokes}(\text{Bob}) = 0 \)
- \( \text{Smokes}(\text{Charlie}) = 0 \)
- \( \text{Smokes}(\text{Dave}) = 1 \)
- \( \text{Smokes}(\text{Eve}) = 0 \)
- ...

![Diagram showing the relationship between Smokes and Friends with values for k and n-k]
First-Order Model Counting: Example

\[ \Delta = \forall x, y \in \text{People}, (\text{Smokes}(x) \land \text{Friends}(x, y) \Rightarrow \text{Smokes}(y)) \]

- If we know \( \mathbf{D} \) precisely: who smokes, and there are \( k \) smokers?

**Database:**
- Smokes(Alice) = 1
- Smokes(Bob) = 0
- Smokes(Charlie) = 0
- Smokes(Dave) = 1
- Smokes(Eve) = 0
- ...

[Van den Broeck 2015]
First-Order Model Counting: Example

\[ \Delta = \forall x, y \in \text{People}, (\text{Smokes}(x) \land \text{Friends}(x,y) \Rightarrow \text{Smokes}(y)) \]

- If we know \( D \) precisely: who smokes, and there are \( k \) smokers?

**Database:**
- Smokes(Alice) = 1
- Smokes(Bob) = 0
- Smokes(Charlie) = 0
- Smokes(Dave) = 1
- Smokes(Eve) = 0
- ...

[Van den Broeck 2015]
First-Order Model Counting: Example

\[ \Delta = \forall x, y \in \text{People}, \ (\text{Smokes}(x) \land \text{Friends}(x,y) \Rightarrow \text{Smokes}(y)) \]

- If we know D precisely: who smokes, and there are \( k \) smokers?

**Database:**
- Smokes(Alice) = 1
- Smokes(Bob) = 0
- Smokes(Charlie) = 0
- Smokes(Dave) = 1
- Smokes(Eve) = 0
- ...

[Van den Broeck 2015]
First-Order Model Counting: Example

\[ \Delta = \forall x, y \in \text{People}, (\text{Smokes}(x) \land \text{Friends}(x, y) \Rightarrow \text{Smokes}(y)) \]

- If we know D precisely: who smokes, and there are k smokers?

**Database:**
- Smokes(Alice) = 1
- Smokes(Bob) = 0
- Smokes(Charlie) = 0
- Smokes(Dave) = 1
- Smokes(Eve) = 0
- ...

[Van den Broeck 2015]
First-Order Model Counting: Example

\[ \Delta = \forall x, y \in \textbf{People}, (\text{Smokes}(x) \land \text{Friends}(x,y) \Rightarrow \text{Smokes}(y)) \]

- If we know \( D \) precisely: who smokes, and there are \( k \) smokers?

**Database:**
- \( \text{Smokes}(\text{Alice}) = 1 \)
- \( \text{Smokes}(\text{Bob}) = 0 \)
- \( \text{Smokes}(\text{Charlie}) = 0 \)
- \( \text{Smokes}(\text{Dave}) = 1 \)
- \( \text{Smokes}(\text{Eve}) = 0 \)
- ...

[Van den Broeck 2015]
First-Order Model Counting: Example

$\Delta = \forall x, y \in \text{People}, (\text{Smokes}(x) \land \text{Friends}(x,y) \Rightarrow \text{Smokes}(y))$

- If we know $D$ precisely: who smokes, and there are $k$ smokers?

**Database:**

- Smokes(Alice) = 1
- Smokes(Bob) = 0
- Smokes(Charlie) = 0
- Smokes(Dave) = 1
- Smokes(Eve) = 0
- ...

[Van den Broeck 2015]
First-Order Model Counting: Example

\[ \Delta = \forall x, y \in \text{People}, (\text{Smokes}(x) \land \text{Friends}(x, y) \Rightarrow \text{Smokes}(y)) \]

- If we know \( D \) precisely: who smokes, and there are \( k \) smokers?

**Database:**

| Smokes(Alice) = 1 |
| Smokes(Bob) = 0  |
| Smokes(Charlie) = 0 |
| Smokes(Dave) = 1  |
| Smokes(Eve) = 0   |
| ...               |

[Van den Broeck 2015]
First-Order Model Counting: Example

\[ \Delta = \forall x, y \in \text{People}, (\text{Smokes}(x) \land \text{Friends}(x,y) \Rightarrow \text{Smokes}(y)) \]

- If we know D precisely: who smokes, and there are \( k \) smokers?

**Database:**
- Smokes(Alice) = 1
- Smokes(Bob) = 0
- Smokes(Charlie) = 0
- Smokes(Dave) = 1
- Smokes(Eve) = 0
- ...

\[ 2^{n^2} - k(n-k) \] models
First-Order Model Counting: Example

\[ \Delta = \forall x, y \in \textbf{People}, (\text{Smokes}(x) \land \text{Friends}(x, y) \Rightarrow \text{Smokes}(y)) \]

- If we know D precisely: who smokes, and there are \( k \) smokers?

  **Database:**
  
  \[
  \begin{align*}
  \text{Smokes}(\text{Alice}) &= 1 \\
  \text{Smokes}(\text{Bob}) &= 0 \\
  \text{Smokes}(\text{Charlie}) &= 0 \\
  \text{Smokes}(\text{Dave}) &= 1 \\
  \text{Smokes}(\text{Eve}) &= 0 \\
  \vdots
  \end{align*}
  \]
  
  \[ \rightarrow 2^{n^2 - k(n-k)} \text{ models} \]

- If we know that there are \( k \) smokers?

[Van den Broeck 2015]
First-Order Model Counting: Example

\[ \Delta = \forall x, y \in \text{People}, (\text{Smokes}(x) \land \text{Friends}(x, y) \Rightarrow \text{Smokes}(y)) \]

- If we know \( \mathbf{D} \) precisely: who smokes, and there are \( k \) smokers?

\[
\text{Database:} \quad 
\begin{align*}
\text{Smokes}(\text{Alice}) &= 1 \\
\text{Smokes}(\text{Bob}) &= 0 \\
\text{Smokes}(\text{Charlie}) &= 0 \\
\text{Smokes}(\text{Dave}) &= 1 \\
\text{Smokes}(\text{Eve}) &= 0 \\
\end{align*}
\]

\[ \Rightarrow 2^{n^2 - k(n - k)} \quad \text{models} \]

- If we know that there are \( k \) smokers?

\[ \Rightarrow \binom{n}{k} 2^{n^2 - k(n - k)} \quad \text{models} \]

[Van den Broeck 2015]
First-Order Model Counting: Example

\[ \Delta = \forall x, y \in \text{People}, (\text{Smokes}(x) \land \text{Friends}(x,y) \Rightarrow \text{Smokes}(y)) \]

- If we know \( D \) precisely: who smokes, and there are \( k \) smokers?

  \[ \text{Database:} \]
  
  - Smokes(Alice) = 1
  - Smokes(Bob) = 0
  - Smokes(Charlie) = 0
  - Smokes(Dave) = 1
  - Smokes(Eve) = 0
  - ...

  \[ \Rightarrow 2^n^2 - k(n-k) \text{ models} \]

- If we know that there are \( k \) smokers?

  \[ \Rightarrow \binom{n}{k} 2^n^2 - k(n-k) \text{ models} \]

- In total...

[Van den Broeck 2015]
First-Order Model Counting: Example

\[ \Delta = \forall x, y \in \text{People}, (\text{Smokes}(x) \land \text{Friends}(x,y) \Rightarrow \text{Smokes}(y)) \]

- If we know \( D \) precisely: who smokes, and there are \( k \) smokers?

  Database:
  
  \[
  \begin{align*}
  &\text{Smokes}(\text{Alice}) = 1 \\
  &\text{Smokes}(\text{Bob}) = 0 \\
  &\text{Smokes}(\text{Charlie}) = 0 \\
  &\text{Smokes}(\text{Dave}) = 1 \\
  &\text{Smokes}(\text{Eve}) = 0 \\
  &\ldots
  \end{align*}
  \]

  \[ \Rightarrow 2^{n^2 - k(n - k)} \] models

- If we know that there are \( k \) smokers?

  \[ \Rightarrow \binom{n}{k} 2^{n^2 - k(n - k)} \] models

- In total...

  \[ \Rightarrow \sum_{k=0}^{n} \binom{n}{k} 2^{n^2 - k(n - k)} \] models

[Van den Broeck 2015]
Compilation Rules

• Standard rules
  – Shannon decomposition (DPLL)
  – Detect decomposability
  – Etc.

• FO Shannon decomposition:

[Van den Broeck 2013]
Playing Cards Revisited

Let us automate this:

\[
\forall p, \exists c, \text{Card}(p,c) \\
\forall c, \exists p, \text{Card}(p,c) \\
\forall p, \forall c, \forall c', \text{Card}(p,c) \land \text{Card}(p,c') \Rightarrow c = c'
\]

[Van den Broeck 2015]
Playing Cards Revisited

Let us automate this:

\[ \forall p, \exists c, \text{Card}(p,c) \]
\[ \forall c, \exists p, \text{Card}(p,c) \]
\[ \forall p, \forall c, \forall c', \text{Card}(p,c) \land \text{Card}(p,c') \Rightarrow c = c' \]

\[ \#\text{SAT} = \sum_{k=0}^{n} \left( \begin{array}{c} n \end{array} \right) \sum_{l=0}^{n} \left( \begin{array}{c} n \end{array} \right) (l + 1)^k (-1)^{2n-k-l} = n! \]

[Van den Broeck 2015]
Playing Cards Revisited

Let us automate this:

\[ \forall p, \exists c, \text{Card}(p, c) \]

\[ \forall c, \exists p, \text{Card}(p, c) \]

\[ \forall p, \forall c, \forall c', \text{Card}(p, c) \land \text{Card}(p, c') \implies c = c' \]

\[ \#\text{SAT} = \sum_{k=0}^{n} \binom{n}{k} \sum_{l=0}^{n} \binom{n}{l} (l + 1)^k (-1)^{2n-k-l} = n! \]

Computed in time polynomial in n

[Van den Broeck 2015]
Perspectives...
What I did not talk about… in KC

• Other queries and transformations
  (see Dan Olteanu poster)
• Other KC languages
  (FO-AODD)
• KC for logic programs
  (see Vlasselaer poster)

[Gogate 2010, Vlasselaer 2015]
What I did not talk about…in FOMC

- WFOMC for probabilistic databases
  (see Gribkoff poster)
- WFOMC for probabilistic programs
  (see Vlasselaer poster)
- Complexity theory (data or domain)
  - $\text{PTime}$ domain complexity for 2-var fragment
  - $\#P_1$ domain complexity for some 3-var CNFs

What I did not talk about...in FO

• Very related problems
  – Lifted inference in SRL

• Very related applications
  – Approximate lifted inference in Markov Logic
  – Learn Markov logic networks

• Classical first-order reasoning
  – Answer set programming,
  – SMT,
  – Theorem proving

[Kersting 2011]
Format for First-Order Beyond NP

- DIMACS contributed to SAT success
- Goals
  - Trivial to parse
  - Captures MLNs, Prob. Programs, Prob. DBs
  - *Not* a powerful representation language
- FO-CNF format under construction
- Vibhav?

```plaintext
p fo-cnf 2 1
d people 1000
r Friends(people,people)
r Smokes(people)
~Smokes(x) ~Friends(x,y) Smokes(y)
w Friends 3.5 1.2
w Smokes -0.5 4
```
Calendar

At IJCAI in New York on July 9-11

• StarAI 2016 (http://www.starai.org/2016/)
  Sixth International Workshop on Statistical Relational AI

• IJCAI Tutorial
  “Lifted Probabilistic Inference in Relational Models” with Dan Suciu
Conclusions

• FOMC is BeyondNP reduction target
• Existing solvers inadequate
  Exponential speedups from FO solvers
• FOKC is Elegant, more than FOMC
• Intersection of communities
  – Statistical relational learning (lifted inference)
  – Probabilistic databases
  – Automated reasoning (you!)
References


References


