Symmetry in Probabilistic Databases

Guy Van den Broeck
KU Leuven

Joint work with

Dan Suciu, Paul Beame, Eric Gribkoff,
Wannes Meert, Adnan Darwiche

Based on NIPS 2011, KR 2014, and upcoming PODS 2015 paper
Overview

• Motivation and convergence of
  – The artificial intelligence story (*recap*)
  – The machine learning story (*recap*)
  – The probabilistic database story
  – The database theory story
• Main theoretical results and proof outlines
• Discussion and conclusions
• Dessert
Overview

• Motivation and convergence of
  – The artificial intelligence story (recap)
  – The machine learning story (recap)
  – The probabilistic database story
  – The database theory story
• Main theoretical results and proof outlines
• Discussion and conclusions
• Dessert
A Simple Reasoning Problem

Probability that Card1 is Hearts?

[Van den Broeck; AAAI-KRR’15]
A Simple Reasoning Problem

?  

Probability that Card1 is Hearts? 1/4

[Van den Broeck; AAAI-KRR’15]
A Simple Reasoning Problem

Probability that Card52 is Spades given that Card1 is QH?

[Van den Broeck; AAAI-KRR’15]
A Simple Reasoning Problem

Probability that Card52 is Spades given that Card1 is QH?

\[
\frac{13}{51}
\]

[Van den Broeck; AAAI-KRR’15]
Automated Reasoning

Let us automate this:

1. Probabilistic graphical model (e.g., factor graph)
2. Probabilistic inference algorithm (e.g., variable elimination or junction tree)
Automated Reasoning

Let us automate this:

1. Probabilistic graphical model (e.g., factor graph) is fully connected!

2. Probabilistic inference algorithm (e.g., variable elimination or junction tree) builds a table with $52^{52}$ rows

[Van den Broeck; AAAI-KRR’15]
What's Going On Here?

Probability that Card52 is Spades given that Card1 is QH?

[Van den Broeck; AAAI-KRR'15]
What's Going On Here?

Probability that Card52 is Spades given that Card1 is QH?

13/51

[Van den Broeck; AAAI-KRR’15]
What's Going On Here?

Probability that Card52 is Spades given that Card2 is QH?

[Van den Broeck; AAAI-KRR’15]
What's Going On Here?

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

[Van den Broeck; AAAI-KRR’15]
What's Going On Here?

Probability that Card52 is Spades given that Card3 is QH?

[Van den Broeck; AAAI-KRR’15]
What's Going On Here?

Probability that Card52 is Spades given that Card3 is QH? 13/51

[Van den Broeck; AAAI-KRR'15]
Tractable Probabilistic Inference

Which property makes inference tractable?
Traditional belief: Independence
What's going on here?

[Niepert, Van den Broeck; AAAI’14], [Van den Broeck; AAAI-KRR’15]
Tractable Probabilistic Inference

Which property makes inference tractable?

Traditional belief: Independence

What's going on here?

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

⇒ Lifted Inference

[Niepert, Van den Broeck; AAAI’14], [Van den Broeck; AAAI-KRR’15]
Let us automate this:

- **Relational** model

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

- **Lifted** probabilistic inference algorithm
Playing Cards Revisited

Let us automate this:

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

Playing Cards Revisited

Let us automate this:

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

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

Let us automate this:

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

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

Computed in time polynomial in \( n \)

Model Counting

- Model = solution to a propositional logic formula $\Delta$
- Model counting = $\#\text{SAT}$

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

<table>
<thead>
<tr>
<th>Rain</th>
<th>Cloudy</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>

$\#\text{SAT} = 3$

[Valiant] $\#P$-hard, even for 2CNF
First-Order Model Counting

Model = solution to \textit{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>

$$\text{FOMC} = 3$$
# 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>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>F</td>
<td>Yes</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>Yes</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>
<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>Yes</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>
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>

$\text{FOMC} = 9$
FOMC Inference: Example 1
FOMC Inference: Example 1

3. $\Delta = \forall x, (\text{Stress}(x) \Rightarrow \text{Smokes}(x))$

Domain = \{n people\}
FOMC Inference: Example 1

3. \( \Delta = \forall x, (\text{Stress}(x) \Rightarrow \text{Smokes}(x)) \) 

\( \rightarrow 3^n \text{ models} \)

Domain = \{n people\}
FOMC Inference: Example 1

3. $\Delta = \forall x, \ (\text{Stress}(x) \Rightarrow \text{Smokes}(x))$
   \hspace{1cm} Domain = \{n \text{ people}\}

$\rightarrow 3^n \text{ models}$

2. $\Delta = \forall y, \ (\text{ParentOf}(y) \land \text{Female} \Rightarrow \text{MotherOf}(y))$
   \hspace{1cm} D = \{n \text{ people}\}$
FOMC Inference: Example 1

2. \[ \Delta = \forall y, (\text{ParentOf}(y) \land \text{Female} \Rightarrow \text{MotherOf}(y)) \quad \text{D} = \{n \text{ people}\} \]

If Female = true? \[ \Delta = \forall y, (\text{ParentOf}(y) \Rightarrow \text{MotherOf}(y)) \quad \rightarrow 3^n \text{ models} \]

3. \[ \Delta = \forall x, (\text{Stress}(x) \Rightarrow \text{Smokes}(x)) \quad \text{Domain} = \{n \text{ people}\} \]

\[ \rightarrow 3^n \text{ models} \]
FOMC Inference: Example 1

2. \[ \Delta = \forall y, (\text{ParentOf}(y) \land \text{Female} \implies \text{MotherOf}(y)) \]  
   \[ \text{D} = \{n \text{ people}\} \]  
   \[ \text{If Female = true?} \quad \Delta = \forall y, (\text{ParentOf}(y) \implies \text{MotherOf}(y)) \quad \rightarrow 3^n \text{ models} \]  
   \[ \text{If Female = false?} \quad \Delta = \text{true} \quad \rightarrow 4^n \text{ models} \]  

3. \[ \Delta = \forall x, (\text{Stress}(x) \implies \text{Smokes}(x)) \]  
   \[ \text{Domain} = \{n \text{ people}\} \]  
   \[ \rightarrow 3^n \text{ models} \]
FOMC Inference: Example 1

2. \( \Delta = \forall y, (\text{ParentOf}(y) \land \text{Female} \Rightarrow \text{MotherOf}(y)) \)
   \( D = \{n \text{ people}\} \)

   If Female = true? \( \Delta = \forall y, (\text{ParentOf}(y) \Rightarrow \text{MotherOf}(y)) \)
   \( \rightarrow 3^n \text{ models} \)

   If Female = false? \( \Delta = \text{true} \)
   \( \rightarrow 4^n \text{ models} \)

   \( \rightarrow 3^n + 4^n \text{ models} \)

3. \( \Delta = \forall x, (\text{Stress}(x) \Rightarrow \text{Smokes}(x)) \)
   \( D = \{n \text{ people}\} \)

   \( \rightarrow 3^n \text{ models} \)
FOMC Inference: Example 1

1. \[ \Delta = \forall x, y, (\text{ParentOf}(x,y) \land \text{Female}(x) \Rightarrow \text{MotherOf}(x,y)) \]  
   \[ \text{Domain} = \{n \text{ people}\} \]  
   \[ \rightarrow 3^n \text{ models} \]

2. \[ \Delta = \forall y, (\text{ParentOf}(y) \land \text{Female} \Rightarrow \text{MotherOf}(y)) \]  
   \[ \text{D} = \{n \text{ people}\} \]  
   If Female = true? \[ \Delta = \forall y, (\text{ParentOf}(y) \Rightarrow \text{MotherOf}(y)) \]  
   \[ \rightarrow 3^n \text{ models} \]  
   If Female = false? \[ \Delta = \text{true} \]  
   \[ \rightarrow 4^n \text{ models} \]  
   \[ \rightarrow 3^n + 4^n \text{ models} \]

3. \[ \Delta = \forall x, (\text{Stress}(x) \Rightarrow \text{Smokes}(x)) \]  
   \[ \text{Domain} = \{n \text{ people}\} \]  
   \[ \rightarrow 3^n \text{ models} \]
# FOMC Inference: Example 1

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1.</strong></td>
<td>$\Delta = \forall x, y, \text{ParentOf}(x,y) \land \text{Female}(x) \Rightarrow \text{MotherOf}(x,y)$</td>
<td>$D = {n \text{ people}}$</td>
</tr>
<tr>
<td></td>
<td>$\rightarrow (3^n + 4^n)^n$ models</td>
<td></td>
</tr>
<tr>
<td><strong>2.</strong></td>
<td>$\Delta = \forall y, (\text{ParentOf}(y) \land \text{Female}) \Rightarrow \text{MotherOf}(y)$</td>
<td>$D = {n \text{ people}}$</td>
</tr>
<tr>
<td>If Female = true?</td>
<td>$\Delta = \forall y, (\text{ParentOf}(y) \Rightarrow \text{MotherOf}(y))$</td>
<td>$\rightarrow 3^n$ models</td>
</tr>
<tr>
<td>If Female = false?</td>
<td>$\Delta = \text{true}$</td>
<td>$\rightarrow 4^n$ models</td>
</tr>
<tr>
<td></td>
<td>$\rightarrow 3^n + 4^n$ models</td>
<td></td>
</tr>
<tr>
<td><strong>3.</strong></td>
<td>$\Delta = \forall x, (\text{Stress}(x) \Rightarrow \text{Smokes}(x))$</td>
<td>$\text{Domain} = {n \text{ people}}$</td>
</tr>
<tr>
<td></td>
<td>$\rightarrow 3^n$ models</td>
<td></td>
</tr>
</tbody>
</table>
FOMC Inference : Example 2

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

Domain = \{n people\}
FOMC Inference : Example 2

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

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

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

\[ \text{Domain} = \{n \text{ people}\} \]
If we know precisely who smokes, and there are $k$ smokers?

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

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

Domain = \{n people\}
FOMC Inference : Example 2

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

Domain = \{ n \text{ people} \}

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

**Database:**

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

```
  Smokes
  ↓
k

  Friends
  →

  Smokes
  ↓
k
```

```
  Smokes
  ↓
n-k
```

```
  Smokes
  ↓
n-k
```
FOMC Inference : Example 2

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

Domain = \{n people\}

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

**Database:**

- Smokes(Alice) = 1
- Smokes(Bob) = 0
- Smokes(Charlie) = 0
- Smokes(Dave) = 1
- Smokes(Eve) = 0
- ...
FOMC Inference : Example 2

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

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

<table>
<thead>
<tr>
<th>Database:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smokes(Alice) = 1</td>
</tr>
<tr>
<td>Smokes(Bob) = 0</td>
</tr>
<tr>
<td>Smokes(Charlie) = 0</td>
</tr>
<tr>
<td>Smokes(Dave) = 1</td>
</tr>
<tr>
<td>Smokes(Eve) = 0</td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>

Domain = \{n people\}
If we know precisely who smokes, and there are $k$ smokers?

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

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

Domain = \{n people\}
If we know precisely who smokes, and there are $k$ smokers?

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

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

Domain = \{n people\}
FOMC Inference : Example 2

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

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

Domain = \{n people\}

**Database:**
- Smokes(Alice) = 1
- Smokes(Bob) = 0
- Smokes(Charlie) = 0
- Smokes(Dave) = 1
- Smokes(Eve) = 0
- ...
FOMC Inference : Example 2

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

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

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

\[ \text{Domain} = \{n \text{ people}\} \]
FOMC Inference: Example 2

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

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

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

Domain = \{n people\}

\[ 2^{n^2 - k(n-k)} \text{ models} \]
FOMC Inference : Example 2

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

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

**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?
FOMC Inference: Example 2

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

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

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

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

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

\[ \Rightarrow \binom{n}{k} 2^{n^2-k(n-k)} \] models
FOMC Inference: Example 2

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

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

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

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

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

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

- In total...
FOMC Inference : Example 2

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

Domain = \{n people\}

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

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

$\rightarrow 2^{n^2 - k(n-k)}$ 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
Overview

• Motivation and convergence of
  – The artificial intelligence story (recap)
  – The machine learning story (recap)
  – The probabilistic database story
  – The database theory story
• Main theoretical results and proof outlines
• Discussion and conclusions
• Dessert
Statistical Relational Models

An MLN = set of constraints \((w, \Gamma(x))\)

- **Weight of a world** = product of \(w\), for all rules \((w, \Gamma(x))\) and groundings \(\Gamma(a)\) that hold in the world

\[
P_{\text{MLN}}(Q) = \frac{\text{[sum of weights of models of } Q\text{]}}{Z}
\]

Applications: large KBs, e.g. DeepDive
Weighted Model Counting

- Model = solution to a propositional logic formula $\Delta$
- Model counting = \#SAT

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

<table>
<thead>
<tr>
<th>Rain</th>
<th>Cloudy</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>T</td>
<td>F</td>
<td>Yes</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>

$$\#\text{SAT} = 3$$
Weighted Model Counting

- Model = solution to a propositional logic formula $\Delta$
- Model counting = $\#SAT$
- Weighted model counting (WMC)
  - Weights for assignments to variables
  - Model weight is product of variable weights $w(.)$

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

<table>
<thead>
<tr>
<th>Rain</th>
<th>Cloudy</th>
<th>Model?</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>Yes</td>
<td>1 * 3 = 3</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>No</td>
<td>0</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>Yes</td>
<td>2 * 3 = 6</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>Yes</td>
<td>2 * 5 = 10</td>
</tr>
</tbody>
</table>

$\#SAT = 3$
Weighted Model Counting

- Model = solution to a propositional logic formula $\Delta$
- Model counting = $\#\text{SAT}$
- Weighted model counting (WMC)
  - Weights for assignments to variables
  - Model weight is product of variable weights $w(.)$

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

<table>
<thead>
<tr>
<th>Rain</th>
<th>Cloudy</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model?</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>$1 \times 3 = 3$</td>
</tr>
<tr>
<td>No</td>
<td>$0$</td>
</tr>
<tr>
<td>Yes</td>
<td>$2 \times 3 = 6$</td>
</tr>
<tr>
<td>Yes</td>
<td>$2 \times 5 = 10$</td>
</tr>
</tbody>
</table>

$\#\text{SAT} = 3$

$\text{WMC} = 19$
Assembly language for probabilistic reasoning and learning

- Bayesian networks
- Factor graphs
- Probabilistic logic programs
- Markov Logic
- Relational Bayesian networks
- Probabilistic databases
- Weighted Model Counting
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>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></th>
<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>T</td>
<td>Yes</td>
</tr>
<tr>
<td>T</td>
<td>F</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>T</td>
<td>Yes</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>Yes</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>T</td>
<td>F</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>T</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>T</td>
<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>F</td>
<td>T</td>
<td>No</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>Yes</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Weighted First-Order Model Counting

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

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

Days = {Monday, Tuesday}

<table>
<thead>
<tr>
<th>$\text{Rain}(\text{M})$</th>
<th>$\text{Cloudy}(\text{M})$</th>
<th>$\text{Rain}(\text{T})$</th>
<th>$\text{Cloudy}(\text{T})$</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>$1 * 1 * 3 * 3 = 9$</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>No</td>
<td>$2 * 1 * 3 * 3 = 18$</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>Yes</td>
<td>$2 * 1 * 5 * 3 = 30$</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>Yes</td>
<td>$2 * 2 * 3 * 3 = 36$</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>Yes</td>
<td>$2 * 2 * 3 * 5 = 60$</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>No</td>
<td>$2 * 2 * 3 * 5 = 60$</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>Yes</td>
<td>$2 * 2 * 5 * 5 = 100$</td>
</tr>
</tbody>
</table>

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

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

### Formula

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

### Days

Days = \{Monday, Tuesday\}

### Weights

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

### Model Checking

<table>
<thead>
<tr>
<th>Days</th>
<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>T</td>
<td>Yes</td>
<td>$1 \times 1 \times 3 \times 3 = 9$</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>No</td>
<td>$2 \times 1 \times 3 \times 3 = 0$</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>Yes</td>
<td>$2 \times 1 \times 3 \times 3 = 18$</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>Yes</td>
<td>$2 \times 1 \times 5 \times 3 = 30$</td>
</tr>
<tr>
<td>T</td>
<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>T</td>
<td>F</td>
<td>No</td>
<td>$0$</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>No</td>
<td>$0$</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>No</td>
<td>$0$</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>Yes</td>
<td>$1 \times 2 \times 3 \times 3 = 18$</td>
</tr>
<tr>
<td>T</td>
<td>F</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>F</td>
<td>T</td>
<td>Yes</td>
<td>$2 \times 2 \times 3 \times 3 = 36$</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>Yes</td>
<td>$2 \times 2 \times 5 \times 3 = 60$</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>Yes</td>
<td>$1 \times 2 \times 3 \times 5 = 30$</td>
</tr>
<tr>
<td>T</td>
<td>F</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>F</td>
<td>Yes</td>
<td>$2 \times 2 \times 3 \times 5 = 60$</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>Yes</td>
<td>$2 \times 2 \times 5 \times 5 = 100$</td>
</tr>
</tbody>
</table>

$\#\text{SAT} = 9$

WFOMC = 361
Assembly language for high-level probabilistic reasoning and learning

- Parfactor graphs
- Probabilistic logic programs
- Markov Logic
- Relational Bayesian networks
- Probabilistic databases
- Weighted First-Order Model Counting

[VdB et al.; IJCAI’11, PhD’13, KR’14, UAI’14]
Symmetric WFOMC

Def. A weighted vocabulary is \((R, w)\), where

- \(R = (R_1, R_2, ..., R_k) = \) relational vocabulary
- \(w = (w_1, w_2, ..., w_k) = \) weights

• Fix an FO formula \(Q\), domain of size \(n\)
• The weight of a ground tuple \(t\) in \(R_i\) is \(w_i\)

This talk: complexity of FOMC / WFOMC(\(Q, n\))
• Data complexity: fixed \(Q\), input \(n\) / and \(w\)
• Combined complexity: input (\(Q, n\)) / and \(w\)
Example

\[ Q = \forall x \exists y \ R(x,y) \]
\[ \text{FOMC}(Q, n) = (2^n - 1)^n \quad \text{WOMC}(Q, n, w_R) = ((1 + w_R)^n - 1)^n \]
Example

\[ Q = \forall x \exists y \ R(x,y) \]

\[ \text{FOMC}(Q, n) = (2^n-1)^n \quad \text{WOMC}(Q, n, w_R) = ((1+w_R)^n-1)^n \]

\[ Q = \exists x \exists y \ [R(x) \land S(x,y) \land T(y)] \]

\[ \text{FOMC}(Q, n) = \sum_{i=0}^{n} \sum_{j=0}^{n} \binom{n}{i} \binom{n}{j} 2^{(n-i)(n-j)} (2^{ij} - 1) \]

Computable in PTIME in \( n \)
Example

\( Q = \forall x \exists y \ R(x,y) \)

\[ \text{FOMC}(Q, n) = (2^n - 1)^n \quad \text{WOMC}(Q, n, w_R) = ((1 + w_R)^n - 1)^n \]

\( Q = \exists x \exists y \ [R(x) \land S(x,y) \land T(y)] \)

\[ \text{FOMC}(Q, n) = \sum_{i=0}^{n} \sum_{j=0}^{n} \binom{n}{i} \binom{n}{j} 2^{(n-i)(n-j)} (2^{ij} - 1) \]

\[ \text{WFOMC}(Q, n, w_R, w_S, w_T) = \sum_{i=0}^{n} \sum_{j=0}^{n} \binom{n}{i} \binom{n}{j} w_R^i w_T^j (1 + w_S)^{(n-i)(n-j)} ((1 + w_S)^{ij} - 1) \]

Computable in PTIME in \( n \)
Example

\[ Q = \exists x \exists y \exists z \ [R(x, y) \land S(y, z) \land T(z, x)] \]

Can we compute \( FOMC(Q, n) \) in PTIME?

Open problem…

Conjecture \( FOMC(Q, n) \) not computable in PTIME in \( n \)
From MLN to WFOMC

MLN:  
∞ Smoker(x) ⇒ Person(x)

w ~Smoker(x) ∨ ~Friend(x,y) ∨ Smoker(y)

MLN’:  
∞ Smoker(x) ⇒ Person(x)

∞ R(x,y) ⇐ ~Smoker(x) ∨ ~Friend(x,y) ∨ Smoker(y)

w R(x,y)

Theorem  
\[ P_{MLN}(Q) = P(Q \mid \text{hard constraints in } \text{MLN’}) = WFOMC(Q \land \text{MLN’}) / WFOMC(\text{MLN’}) \]

R is a symmetric relation
Overview

• Motivation and convergence of
  – The artificial intelligence story (*recap*)
  – The machine learning story (*recap*)
  – The probabilistic database story
  – The database theory story
• Main theoretical results and proof outlines
• Discussion and conclusions
• Dessert
Probabilistic Databases

- Weights or probabilities given explicitly, for each tuple

- Examples: Knowledge Vault, Nell, Yago

- Dichotomy theorem:
  for any query in UCQ/FO(∃,∧,∨) (or FO(∀,∧,∨), asymmetric WFOMC is in PTIME or #P-hard.)
Motivation 2: Probabilistic Databases

Probabilistic database $D$:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$x$</td>
<td>$y$</td>
<td>$P$</td>
</tr>
<tr>
<td>$a_1$</td>
<td>$b_1$</td>
<td>$p_1$</td>
</tr>
<tr>
<td>$a_1$</td>
<td>$b_2$</td>
<td>$p_2$</td>
</tr>
<tr>
<td>$a_2$</td>
<td>$b_2$</td>
<td>$p_3$</td>
</tr>
</tbody>
</table>
Motivation 2: Probabilistic Databases

Probabilistic database $D$:

<table>
<thead>
<tr>
<th>$x$</th>
<th>$y$</th>
<th>$P$</th>
</tr>
</thead>
<tbody>
<tr>
<td>a1</td>
<td>b1</td>
<td>$p_1$</td>
</tr>
<tr>
<td>a1</td>
<td>b2</td>
<td>$p_2$</td>
</tr>
<tr>
<td>a2</td>
<td>b2</td>
<td>$p_3$</td>
</tr>
</tbody>
</table>

Possible worlds semantics:

<table>
<thead>
<tr>
<th>$x$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>a1</td>
<td>b1</td>
</tr>
<tr>
<td>a1</td>
<td>b2</td>
</tr>
<tr>
<td>a2</td>
<td>b2</td>
</tr>
</tbody>
</table>

$p_1 p_2 p_3$
Motivation 2: Probabilistic Databases

Probabilistic database $D$:

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>a1</td>
<td>b1</td>
<td>$p_1$</td>
</tr>
<tr>
<td>a1</td>
<td>b2</td>
<td>$p_2$</td>
</tr>
<tr>
<td>a2</td>
<td>b2</td>
<td>$p_3$</td>
</tr>
</tbody>
</table>

Possible worlds semantics:

- $p_1 p_2 p_3$
- $(1-p_1)p_2 p_3$
- $p_1 p_2 p_3$
Motivation 2: Probabilistic Databases

Probabilistic database $D$:

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>a1</td>
<td>b1</td>
<td>$p_1$</td>
<td></td>
</tr>
<tr>
<td>a1</td>
<td>b2</td>
<td>$p_2$</td>
<td></td>
</tr>
<tr>
<td>a2</td>
<td>b2</td>
<td>$p_3$</td>
<td></td>
</tr>
</tbody>
</table>

Possible worlds semantics:

$p_1p_2p_3$

$(1-p_1)p_2p_3$

$((1-p_1)(1-p_2)(1-p_3))$
An Example

\[ Q = \exists x \exists y \ R(x) \land S(x, y) \]

\[ P(Q) = \]

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>a_1</td>
<td>b_1</td>
<td>q_1</td>
</tr>
<tr>
<td>a_1</td>
<td>b_2</td>
<td>q_2</td>
</tr>
<tr>
<td>a_2</td>
<td>b_3</td>
<td>q_3</td>
</tr>
<tr>
<td>a_2</td>
<td>b_4</td>
<td>q_4</td>
</tr>
<tr>
<td>a_2</td>
<td>b_5</td>
<td>q_5</td>
</tr>
</tbody>
</table>
An Example

\[Q = \exists x \exists y \ R(x) \land S(x, y)\]

\[P(Q) = 1 - (1 - q_1)(1 - q_2)\]

\[
\begin{array}{c|c|c}
    x & y & P \\
    \hline
    a_1 & b_1 & q_1 \\
    a_1 & b_2 & q_2 \\
    a_2 & b_3 & q_3 \\
    a_2 & b_4 & q_4 \\
    a_2 & b_5 & q_5 \\
\end{array}
\]
An Example

\[ P(Q) = p_1 \times [ 1 - (1-q_1)(1-q_2) ] \]

\[
Q = \exists x \exists y \, R(x) \land S(x,y)
\]
An Example

\[
P(Q) = p_1 \left[ 1 - (1-q_1)(1-q_2) \right] 
1 - (1-q_3)(1-q_4)(1-q_5) 
\]

\[
Q = \exists x \exists y \ R(x) \land S(x, y)
\]
An Example

\[ Q = \exists x \exists y \ R(x) \land S(x,y) \]

\[ P(Q) = p_1 \left[ 1-(1-q_1)(1-q_2) \right] \]
\[ p_2 \left[ 1-(1-q_3)(1-q_4)(1-q_5) \right] \]

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>a_1</td>
<td>b_1</td>
<td>q_1</td>
</tr>
<tr>
<td>a_1</td>
<td>b_2</td>
<td>q_2</td>
</tr>
<tr>
<td>a_2</td>
<td>b_3</td>
<td>q_3</td>
</tr>
<tr>
<td>a_2</td>
<td>b_4</td>
<td>q_4</td>
</tr>
<tr>
<td>a_2</td>
<td>b_5</td>
<td>q_5</td>
</tr>
</tbody>
</table>
\[ Q = \exists x \exists y \ R(x) \land S(x, y) \]

**An Example**

\[
P(Q) = 1 - \left\{ 1 - p_1 \left[ 1 - (1 - q_1)(1 - q_2) \right] \right\} \times \left\{ 1 - p_2 \left[ 1 - (1 - q_3)(1 - q_4)(1 - q_5) \right] \right\}
\]

**Table:**

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>a_1</td>
<td>b_1</td>
<td>q_1</td>
</tr>
<tr>
<td>a_1</td>
<td>b_2</td>
<td>q_2</td>
</tr>
<tr>
<td>a_2</td>
<td>b_3</td>
<td>q_3</td>
</tr>
<tr>
<td>a_2</td>
<td>b_4</td>
<td>q_4</td>
</tr>
<tr>
<td>a_2</td>
<td>b_5</td>
<td>q_5</td>
</tr>
</tbody>
</table>
\[ Q = \exists x \exists y \; R(x) \land S(x,y) \]

**An Example**

\[
P(Q) = 1 - \{1 - p_1 \left[ 1 - (1 - q_1) \left(1 - q_2\right) \right]\} \times \{1 - p_2 \left[ 1 - (1 - q_3) \left(1 - q_4\right) \left(1 - q_5\right) \right]\}
\]

One can compute \( P(Q) \) in \( \text{PTIME} \) in the size of the database \( D \)

<table>
<thead>
<tr>
<th>R</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>a₁</td>
<td>p₁</td>
</tr>
<tr>
<td>a₂</td>
<td>p₂</td>
</tr>
<tr>
<td>a₃</td>
<td>p₃</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>S</th>
<th>x</th>
<th>y</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>a₁</td>
<td>b₁</td>
<td>q₁</td>
<td></td>
</tr>
<tr>
<td>a₁</td>
<td>b₂</td>
<td>q₂</td>
<td></td>
</tr>
<tr>
<td>a₂</td>
<td>b₃</td>
<td>q₃</td>
<td></td>
</tr>
<tr>
<td>a₂</td>
<td>b₄</td>
<td>q₄</td>
<td></td>
</tr>
<tr>
<td>a₂</td>
<td>b₅</td>
<td>q₅</td>
<td></td>
</tr>
</tbody>
</table>
An Example

\[ Q = \exists x \exists y \ R(x) \land S(x, y) \]

\[ P(Q) = 1 - \left\{ 1 - p_1 \left[ 1 - (1-q_1)(1-q_2) \right] \right\} \times \left\{ 1 - p_2 \left[ 1 - (1-q_3)(1-q_4)(1-q_5) \right] \right\} \]

One can compute \( P(Q) \) in PTIME in the size of the database \( D \)
Probabilistic Database Inference

Preprocess $Q$ (omitted from this talk; see book [S.’2011])

- $P(Q_1 \land Q_2) = P(Q_1)P(Q_2)$
  $P(Q_1 \lor Q_2) = 1 - (1 - P(Q_1))(1 - P(Q_2))$

- $P(\exists z \ Q) = 1 - \prod_{a \in \text{Domain}} (1 - P(Q[a/z]))$
  $P(\forall z \ Q) = \prod_{a \in \text{Domain}} P(Q[a/z])$

- $P(Q_1 \land Q_2) = P(Q_1) + P(Q_2) - P(Q_1 \lor Q_2)$
  $P(Q_1 \lor Q_2) = P(Q_1) + P(Q_2) - P(Q_1 \land Q_2)$

If rules succeed, WFOMC($Q,n$) in PTIME; else, $\#P$-hard

$\#P$-hardness no longer holds for symmetric WFOMC
Overview

• Motivation and convergence of
  – The artificial intelligence story (recap)
  – The machine learning story (recap)
  – The probabilistic database story
  – The database theory story
• Main theoretical results and proof outlines
• Discussion and conclusions
• Dessert
Motivation: 0/1 Laws

Definition. \( \mu_n(Q) \) = fraction of all structures over a domain of size \( n \) that are models of \( Q \)

\[ \mu_n(Q) = \frac{\text{FOMC}(Q, n)}{\text{FOMC}(\text{TRUE}, n)} \]

Theorem.
For every \( Q \) in FO, \( \lim_{n \rightarrow \infty} \mu_n(Q) = 0 \) or 1

Example: \( Q = \forall x \exists y \ R(x, y) \);
\[ \text{FOMC}(Q, n) = (2^n - 1)^n \]
\[ \mu_n(Q) = \frac{(2^n - 1)^n}{2^{n^2}} \rightarrow 1 \]
Motivation: 0/1 Laws

In 1976 Fagin proved the 0/1 law for FO using a transfer theorem.

But is there an elementary proof? Find explicit formula for \( \mu_n(Q) \), then compute the limit. [Fagin communicated to us that he tried this first]
Overview

- Motivation and convergence of
  - The artificial intelligence story (recap)
  - The machine learning story (recap)
  - The probabilistic database story
  - The database theory story
- Main theoretical results and proof outlines
- Discussion and conclusions
- Dessert
Class $\text{FO}^2$

- $\text{FO}^2 = \text{FO}$ restricted to two variables

- Intuition: SQL queries that have a plan where all temp tables have arity $\leq 2$

- “The graph has a path of length 10”:

$$\exists x \exists y (R(x,y) \land \exists x \ (R(y,x) \land \exists y \ (R(x,y) \land \ldots )))$$
Main Positive Results

Data complexity:

- for any formula $Q$ in $\text{FO}^2$, $\text{WFOMC}(Q, n)$ is in PTIME [see NIPS’11, KR’13]
- for any $\gamma$-acyclic conjunctive query w/o self-joins $Q$, $\text{WFOMC}(Q, n)$ is in PTIME
Main Negative Results

Data complexity:
- There exists an FO formula $Q$ s.t. symmetric $\text{FOMC}(Q, n)$ is #P$_1$ hard
- There exists $Q$ in $\text{FO}^3$ s.t. $\text{FOMC}(Q, n)$ is #P$_1$ hard
- There exists a conjunctive query $Q$ s.t. symmetric $\text{WFOMC}(Q, n)$ is #P$_1$ hard
- There exists a positive clause $Q$ w.o. ‘$=$’ s.t. symmetric $\text{WFOMC}(Q, n)$ is #P$_1$ hard

Combined complexity:
- $\text{FOMC}(Q, n)$ is #P-hard
Review: $\#P_1$

- $\#P_1 = \text{class of functions in } \#P \text{ over a unary input alphabet}$

- Valiant 1979: there exists $\#P_1$ complete problems

- Bertoni, Goldwurm, Sabatini 1988: counting strings of a given length in some CFG is $\#P_1$ complete

- Goldberg: “no natural combinatorial problems known to be $\#P_1$ complete”
Main Result 1

Theorem 1. There exists an FO$^3$ sentence $Q$ s.t. $\text{FOMC}(Q,n)$ is $\#P_1$-hard

Proof

• Step 1. Construct a Turing Machine $U$ s.t.
  – $U$ is in $\#P_1$ and runs in linear time in $n$
  – $U$ computes a $\#P_1$–hard function

• Step 2. Construct an FO$^3$ sentence $Q$ s.t. $\text{FOMC}(Q,n) / n! = U(n)$
Main Result 2

**Theorem 2** There exists a Conjunctive Query \( Q \) s.t. \( \text{WFOMC}(Q,n) \) is \( \#P_1 \)-hard

- Note: the decision problem is trivial \((Q \text{ has a model iff } n > 0)\)
- *Unweighted* Model Counting for CQ: open

**Proof** Start with a formula \( Q \) that is \( \#P_1 \)-hard for FOMC, and transform it to a CQ in five steps (next)
Step 1: Remove $\exists$

Rewrite

\[ Q = \forall x \exists y \psi(x,y) \]

to

\[ Q' = \forall x \forall y (\neg \psi(x,y) \lor \neg A(x)) \]

where $A$ = new symbol with weight $w = -1$

Claim: $\text{WFOMC}(Q, n) = \text{WFOMC}(Q', n)$

Proof Consider a model for $Q'$, and a constant $x = a$

- If $\exists b \psi(a,b)$, then $A(a) = \text{false}$; contributes $w = 1$

- Otherwise, $A(a)$ can be either true or false, contributing either $w = 1$ or $w = -1$, and $1 - 1 = 0$.

\[ Q = \forall^* \ldots, \quad \text{WFOMC}(Q, n) \text{ is } \#P_1\text{-hard} \]
Step 2: Remove Negation

• Transform Q to Q’ w/o negation s.t.
\[\text{WFOMC}(Q, n) = \text{WFOMC}(Q', n)\]

• Similarly to step 1 and omitted

\[Q = \forall^*[\text{positive}], \quad \text{WFOMC}(Q, n) \text{ is } \#P_1\text{-hard}\]
Step 3: Remove “=”

Rewrite $Q$ to $Q'$ as follows:

- Add new binary symbol $E$ with weight $w$
- Define: $Q' = Q[ E / "=" ] \land (\forall x \ E(x, x))$

Claim: $WFOMC(Q, n)$ computable using oracle for $WFOMC(Q', n)$
(coefficient of $w^n$ in polynomial $WFOMC(Q', n)$)

Start: $Q$ s.t. $FOMC(Q, n)$ is $\#P_1$-hard
Step 4: To UCQ

- Write $Q = \forall^* (C_1 \land C_2 \land \ldots)$ where each $C_i$ is a positive clause.

- The dual $Q' = \exists^* (C_1' \lor C_2' \lor \ldots)$ is a UCQ.
Step 5: from UCQ to CQ

- **UCQ**: \( Q = C_1 \lor C_2 \lor \ldots \lor C_k \)

- \( P(Q) = \ldots + (-1)^S P(\land_{i \in S} C_i) + \ldots \)

- \( 2^{k-1} \) CQs \( P(Q_1), P(Q_2), \ldots P(Q_{2^{k-1}}) \)

- 1 CQ (using fresh copies of symbols):
  \( P(Q'_1 Q'_2 \ldots Q'_{2^{k-1}}) = P(Q'_1) P(Q'_2) \ldots P(Q'_{2^{k-1}}) \)

**Start:** \( Q \) s.t. FOMC\((Q, n)\) is \#P\(_1\)-hard
Overview

• Motivation and convergence of
  – The artificial intelligence story (recap)
  – The machine learning story (recap)
  – The probabilistic database story
  – The database theory story
• Main theoretical results and proof outlines
• Discussion and conclusions
• Dessert
Motivation: 0/1 Laws

In 1976 Fagin proved the 0/1 law for FO using a transfer theorem.

But is there an elementary proof? Find explicit formula for $\mu_n(Q)$, then compute the limit. [Fagin communicated to us that he tried this first]
Motivation: 0/1 Laws

In 1976 Fagin proved the 0/1 law for FO using a transfer theorem.

But is there an elementary proof? Find explicit formula for $\mu_n(Q)$, then compute the limit. [Fagin communicated to us that he tried this first]

A: unlikely when $\text{FOMC}(Q,n)$ is $\#P_1$-hard
Fagin (1974) restated:
1. $\text{NP} = \exists \text{SO}$
   (Fagin’s classical characterization of NP)
2. $\text{NP}_1 = \{ \text{Spec}(\Phi) \mid \Phi \in \text{FO} \}$ in tally notation
   (less well known!)

We show: $\text{#P}_1$ corresponds to $\{ \text{FOMC}(Q,n) \mid Q \text{ in FO} \}$
Discussion

- Convergence of AI/ML/DB/theory
- First-order model counting is a basic problem that touches all these areas
- Under-investigated
- Hardness proofs are more difficult than for \#P

Open problems:
- New algorithm for symmetric model counting
- New hardness reduction techniques
Fertile Ground

[VD; NIPS’11], [VD et al.; KR’14], [Gribkoff, VdB, Suciu; UAI’15], [Beame, VdB, Gribkoff, Suciu; PODS’15], etc.
Fertile Ground

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

[VD; NIPS’11], [VD et al.; KR’14], [Gribkoff, VdB, Suciu; UAI’15], [Beame, VdB, Gribkoff, Suciu; PODS’15], etc.
Overview

• Motivation and convergence of
  – The artificial intelligence story (*recap*)
  – The machine learning story (*recap*)
  – The probabilistic database story
  – The database theory story
• Main theoretical results and proof outlines
• Discussion and conclusions
• Dessert
The Decision Problem

• Counting problem
  “count the number of XXX s.t...”

• Decision problem
  “does there exists an XXX s.t. ...?”

• #3SAT and 3SAT:
  – counting is #P-complete, decision is NP-hard

• #2SAT and 2SAT:
  – counting is #P-hard, decision is in PTIME
Counting/Decision Problems for FO

- **Counting**: given $Q, n$, count the number of models of $Q$ over a domain of size $n$

- **Decision**: given $Q, n$, does there exists a model of $Q$ over a domain of size $n$?

- **Data complexity**: fix $Q$, input = $n$

- **Combined complexity**: input = $Q, n$
The Spectrum

**Definition.** [Scholz 1952]

$\text{Spec}(\mathbb{Q}) = \{n \mid \mathbb{Q} \text{ has a model over domain } [n]\}$

**Example:** $\mathbb{Q}$ says “$(D, +, *, 0, 1)$ is a field”:

$\text{Spec}(\mathbb{Q}) = \{p^k \mid p \text{ prime, } k \geq 1\}$

Spectra studied intensively for over 50 years

The FO decision problem is precisely spectrum membership
The Data Complexity

Suppose $n$ is given in binary representation:

- Jones&Selman’72: $\text{spectra} = \text{NETIME}$

$$\text{NETIME} = \bigcup_{c \geq 0} \text{NTIME}(2^{cn}) \quad \text{NEXPTIME} = \bigcup_{c \geq 0} \text{NTIME}(2^{cn})$$

Suppose $n$ is given in unary representation:

- Fagin’74: $\text{spectra} = \text{NP}_1$
Combined Complexity

Consider the combined complexity for \( \text{FO}^2 \)
“given \( Q, n \), check if \( n \in \text{Spec}(Q) \)”

We prove its complexity:
• NP-complete for \( \text{FO}^2 \),
• PSPACE-complete complete for FO
Thanks!