Scalable Inference and Learning for High-Level Probabilistic Models

Guy Van den Broeck

KU Leuven
Outline

- Motivation
  - Why high-level representations?
  - Why high-level reasoning?
- Intuition: Inference rules
- Liftability theory: Strengths and limitations
- Lifting in practice
  - Approximate symmetries
  - Lifted learning
Outline

• Motivation
  – Why high-level representations?
  – Why high-level reasoning?
• Intuition: Inference rules
• Liftability theory: Strengths and limitations
• Lifting in practice
  – Approximate symmetries
  – Lifted learning
Graphical Model Learning

Medical Records

<table>
<thead>
<tr>
<th>Name</th>
<th>Cough</th>
<th>Asthma</th>
<th>Smokes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Bob</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Charlie</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Dave</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Eve</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Graphical Model Learning

Medical Records

<table>
<thead>
<tr>
<th>Name</th>
<th>Cough</th>
<th>Asthma</th>
<th>Smokes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Bob</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Charlie</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Dave</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Eve</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Bayesian Network

Asthma

Smokes

Cough
Graphical Model Learning

Medical Records

<table>
<thead>
<tr>
<th>Name</th>
<th>Cough</th>
<th>Asthma</th>
<th>Smokes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Bob</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Charlie</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Dave</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Eve</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Bayesian Network

Big data

Asthma

Smokes

Cough

Medical Records

Bayesian Network

Big data
Graphical Model Learning

<table>
<thead>
<tr>
<th>Name</th>
<th>Cough</th>
<th>Asthma</th>
<th>Smokes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Bob</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Charlie</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Dave</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Eve</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Frank</td>
<td>1</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>
Graphical Model Learning

Medical Records

<table>
<thead>
<tr>
<th>Name</th>
<th>Cough</th>
<th>Asthma</th>
<th>Smokes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Bob</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Charlie</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Dave</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Eve</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Frank | 1 | ? | ?

Frank | 1 | 0.3 | 0.2

Bayesian Network

Asthma
Smokes
Cough

Frank
Graphical Model Learning

Medical Records

<table>
<thead>
<tr>
<th>Name</th>
<th>Cough</th>
<th>Asthma</th>
<th>Smokes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Bob</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Charlie</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Dave</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Eve</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Cough</th>
<th>Asthma</th>
<th>Smokes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frank</td>
<td>1</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>

Frank

<table>
<thead>
<tr>
<th>Name</th>
<th>Cough</th>
<th>Asthma</th>
<th>Smokes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frank</td>
<td>1</td>
<td>0.3</td>
<td>0.2</td>
</tr>
</tbody>
</table>
## Medical Records

<table>
<thead>
<tr>
<th>Name</th>
<th>Cough</th>
<th>Asthma</th>
<th>Smokes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Bob</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Charlie</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Dave</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Eve</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Cough</th>
<th>Friends</th>
<th>Brothers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frank</td>
<td>1</td>
<td>0.3</td>
<td>0.2</td>
</tr>
</tbody>
</table>

## Bayesian Network

- **Asthma**
- **Smokes**
- **Cough**
Graphical Model Learning

Medical Records

<table>
<thead>
<tr>
<th>Name</th>
<th>Cough</th>
<th>Asthma</th>
<th>Smokes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Bob</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Charlie</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Dave</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Eve</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

| Frank  | 1 | ? | ? |

Bayesian Network

- Asthma
- Smokes
- Cough

Friends

Brothers

Frank: 1, 0.3, 0.2

Frank: 1, 0.2, 0.6
Graphical Model Learning

Medical Records

<table>
<thead>
<tr>
<th>Name</th>
<th>Cough</th>
<th>Asthma</th>
<th>Smokes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Bob</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Charlie</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Dave</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Eve</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Bayesian Network

Asthma

Smokes

Cough

Rows are **independent** during learning and inference!
Statistical Relational Representations

Augment graphical model with relations between entities (rows).

**Intuition**

- Friends have similar smoking habits
- Asthma can be hereditary

**Markov Logic**
Statistical Relational Representations

Augment graphical model with relations between entities (rows).

**Intuition**

- Asthma
- Smokes
- Cough

+ Friends have similar smoking habits
+ Asthma can be hereditary

**Markov Logic**

- 2.1 Asthma \(\Rightarrow\) Cough
- 3.5 Smokes \(\Rightarrow\) Cough

+ Asthma can be hereditary
Statistical Relational Representations

Augment graphical model with relations between entities (rows).

**Intuition**

- Friends have similar smoking habits
- Asthma can be hereditary

**Markov Logic**

- 2.1 Asthma(x) ⇒ Cough(x)
- 3.5 Smokes(x) ⇒ Cough(x)

Logical variables refer to entities
Statistical Relational Representations

Augment graphical model with relations between entities (rows).

**Intuition**

+ Friends have similar smoking habits
+ Asthma can be hereditary

**Markov Logic**

2.1 \( \text{Asthma}(x) \Rightarrow \text{Cough}(x) \)

3.5 \( \text{Smokes}(x) \Rightarrow \text{Cough}(x) \)

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

1.5 \( \text{Asthma}(x) \land \text{Family}(x,y) \Rightarrow \text{Asthma}(y) \)
Equivalent Graphical Model

- Statistical relational model (e.g., MLN)

\[ 1.9 \quad \text{Smokes}(x) \land \text{Friends}(x,y) \Rightarrow \text{Smokes}(y) \]

- Ground atom/tuple = random variable in \{true, false\}
  e.g., Smokes(Alice), Friends(Alice,Bob), etc.

- Ground formula = factor in propositional factor graph
Research Overview

Generality

Knowledge Representation

Graphical Models
Bayesian Networks
Research Overview

- Statistical Relational Models
- Probabilistic Databases
- Graphical Models
- Bayesian Networks

Generality vs. Knowledge Representation
Probabilistic Databases

• Tuple-independent probabilistic databases

<table>
<thead>
<tr>
<th>Name</th>
<th>Prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brando</td>
<td>0.9</td>
</tr>
<tr>
<td>Cruise</td>
<td>0.8</td>
</tr>
<tr>
<td>Coppola</td>
<td>0.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Actor</th>
<th>Director</th>
<th>Prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brando</td>
<td>Coppola</td>
<td>0.9</td>
</tr>
<tr>
<td>Coppola</td>
<td>Brando</td>
<td>0.2</td>
</tr>
<tr>
<td>Cruise</td>
<td>Coppola</td>
<td>0.1</td>
</tr>
</tbody>
</table>

• Query: SQL or First-order logic

```
SELECT Actor.name
FROM Actor, WorkedFor
WHERE Actor.name = WorkedFor.actor
```

\[ Q(x) = \exists y \text{Actor}(x) \land \text{WorkedFor}(x,y) \]

• Learned from the web, large text corpora, ontologies, etc., using **statistical** machine learning.
Google Knowledge Graph
Google Knowledge Graph

> 570 million entities
> 18 billion tuples
Research Overview

- Statistical Relational Models
- Probabilistic Databases
- Graphical Models
- Bayesian Networks
Research Overview

- Bayesian Networks
- Probabilistic Databases
- Statistical Relational Models
- Probabilistic Programming
- Graphical Models
Probabilistic Programming

- Programming language + random variables
- Reason about distribution over executions
  
  As going from hardware circuits to programming languages

- **ProbLog**: Probabilistic logic programming/datalog

- Example: Gene/protein interaction networks
  
  Edges (interactions) have probability

  “Does there exist a path connecting two proteins?”

  \[
  \text{path}(X,Y) \leftarrow \text{edge}(X,Y). \\
  \text{path}(X,Y) \leftarrow \text{edge}(X,Z), \text{path}(Z,Y). \\
  \]

  Cannot be expressed in first-order logic

  Need a full-fledged programming language!
Research Overview

- Probabilistic Programming
- Statistical Relational Models
- Probabilistic Databases
- Graphical Models
- Bayesian Networks
Research Overview

Knowledge Representation
- Probabilistic Programming
- Statistical Relational Models
- Probabilistic Databases
- Graphical Models
- Bayesian Networks

Reasoning
- Program Induction
- Statistical Relational Learning
- Graphical Model Learning
- Lifted Inference
- Lifted Learning

Machine Learning
- Program Sampling
- Graphical Model Inference
- Lifted Learning
Research Overview

Knowledge Representation

- Probabilistic Programming
- Statistical Relational Models
- Probabilistic Databases
- Graphical Models
- Bayesian Networks

Reasoning

- Program Sampling
- Graphical Model Inference
- Lifted Inference

Machine Learning

- Program Induction
- Statistical Relational Learning
- Graphical Model Learning
- Lifted Learning
Outline

• Motivation
  – Why high-level representations?
  – Why high-level reasoning?
• Intuition: Inference rules
• Liftability theory: Strengths and limitations
• Lifting in practice
  – Approximate symmetries
  – Lifted learning
A Simple Reasoning Problem

- 52 playing cards
- Let us ask some simple questions

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

Probability that Card1 is Q?
A Simple Reasoning Problem

Probability that Card1 is Q?

1/13

[Van den Broeck; AAAI-KRR’15]
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 Card1 is Hearts given that Card1 is red?

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

Probability that Card52 is Spades given that Card1 is QH? 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)
Classical Reasoning

- Higher treewidth
- Fewer conditional independencies
- Slower inference
Is There Conditional Independence?

\[ P(\text{Card52} \mid \text{Card1}) \not\equiv P(\text{Card52} \mid \text{Card1, Card2}) \]
Is There Conditional Independence?

\[ P(\text{Card52} \mid \text{Card1}) \overset{?}{=} P(\text{Card52} \mid \text{Card1, Card2}) \]

\[ ? \overset{?}{=} ? \]
Is There Conditional Independence?

\[ P(\text{Card52} \mid \text{Card1}) \overset{?}{=} P(\text{Card52} \mid \text{Card1, Card2}) \]

\[ 13/51 \overset{?}{=} ? \]
Is There Conditional Independence?

\[ P(\text{Card52} \mid \text{Card1}) \overset{?}{=} P(\text{Card52} \mid \text{Card1, Card2}) \]

\[ 13/51 \overset{?}{=} ? \]
Is There Conditional Independence?

\[ P(\text{Card52} \mid \text{Card1}) \neq P(\text{Card52} \mid \text{Card1, Card2}) \]

\[ \frac{13}{51} \neq \frac{12}{50} \]
Is There Conditional Independence?

\[ P(\text{Card52} \mid \text{Card1}) \neq P(\text{Card52} \mid \text{Card1}, \text{Card2}) \]

\[ 13/51 \neq 12/50 \]
Is There Conditional Independence?

\[ P(\text{Card52} \mid \text{Card1}) \neq P(\text{Card52} \mid \text{Card1, Card2}) \]

\[ \frac{13}{51} \neq \frac{12}{50} \]

\[ P(\text{Card52} \mid \text{Card1, Card2}) \neq P(\text{Card52} \mid \text{Card1, Card2, Card3}) \]
Is There Conditional Independence?

\[ P(\text{Card52} \mid \text{Card1}) \neq P(\text{Card52} \mid \text{Card1, Card2}) \]

\[ \frac{13}{51} \neq \frac{12}{50} \]

\[ P(\text{Card52} \mid \text{Card1, Card2}) \neq P(\text{Card52} \mid \text{Card1, Card2, Card3}) \]

\[ \frac{12}{50} \neq \frac{12}{49} \]
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?

[Image of playing cards]

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

[Van den Broeck; AAAI-KRR’15]
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; 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 reasoning
- Symmetry
- Exchangeability

⇒ Lifted Inference

[Niepert, Van den Broeck; AAAI’14], [Van den Broeck; AAAI-KRR’15]
Other Examples of Lifted Inference

- Syllogisms & First-order resolution
- Reasoning about populations

We are investigating a rare disease. The disease is more rare in women, presenting only in one in every two billion women and one in every billion men. Then, assuming there are 3.4 billion men and 3.6 billion women in the world, the probability that more than five people have the disease is

\[
1 - \sum_{n=0}^{5} \sum_{f=0}^{n} \binom{3.6 \cdot 10^9}{f} (1 - 0.5 \cdot 10^{-9})^{3.6 \cdot 10^9 - f} (0.5 \cdot 10^{-9})^f \\
\times \binom{3.4 \cdot 10^9}{n-f} (1 - 10^{-9})^{3.4 \cdot 10^9 - (n-f)} (10^{-9})^{n-f}
\]

[Van den Broeck; AAAI-KRR’15], [Van den Broeck; PhD’13]
Equivalent Graphical Model

- Statistical relational model (e.g., MLN)
  \[3.14 \text{ FacultyPage}(x) \land \text{Linked}(x,y) \Rightarrow \text{CoursePage}(y)\]

- As a probabilistic graphical model:
  - 26 pages; 728 variables; 676 factors
  - 1000 pages; 1,002,000 variables; 1,000,000 factors

- Highly intractable?
  - **Lifted inference** in milliseconds!
Outline

• Motivation
  – Why high-level representations?
  – Why high-level reasoning?

• Intuition: Inference rules

• Liftability theory: Strengths and limitations

• Lifting in practice
  – Approximate symmetries
  – Lifted learning
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>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>

$\#SAT = 3$
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 $w(.)$

#### Example
- **Formula**: $\Delta = (\text{Rain} \Rightarrow \text{Cloudy})$
- **Weights**:
  - $w(\text{Rain}) = 1$
  - $w(\neg\text{Rain}) = 2$
  - $w(\text{Cloudy}) = 3$
  - $w(\neg\text{Cloudy}) = 5$

<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>$1 \times 3 = 3$</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>$2 \times 3 = 6$</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>Yes</td>
<td>$2 \times 5 = 10$</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 $w(.)$

### Example

<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>

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

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

$\text{#SAT} = 3$
$\text{WMC} = 19$
Assembly language for probabilistic reasoning

- 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\}
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}

<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>

$\#\text{SAT} = 3$
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 $\Delta$

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

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>Yes</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>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>
</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}

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

<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>$1 \times 1 \times 3 \times 3 = 9$</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>No</td>
<td>$2 \times 1 \times 3 \times 3 = 18$</td>
</tr>
<tr>
<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>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>Yes</td>
<td>$2 \times 1 \times 5 \times 5 = 100$</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</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>No</td>
<td>$2 \times 2 \times 3 \times 3 = 36$</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>Yes</td>
<td>$2 \times 2 \times 5 \times 3 = 60$</td>
</tr>
<tr>
<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>
<tr>
<td>T</td>
<td>T</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>No</td>
<td>$2 \times 2 \times 3 \times 5 = 60$</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>Yes</td>
<td>$2 \times 2 \times 5 \times 5 = 100$</td>
</tr>
<tr>
<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>

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

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

**Formula:**
\[
\Delta = \forall d \ (\text{Rain}(d) \Rightarrow \text{Cloudy}(d))
\]

**Days:** {Monday, Tuesday}

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

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

+$\text{SAT}=9$ $+$ $\text{WFOMC}=361$
Assembly language for high-level probabilistic reasoning

- 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]
WFOMC Inference: Example

- FO-Model Counting: $w(R) = w(\neg R) = 1$
- Apply inference rules backwards (step 4-3-2-1)
WFOMC Inference: Example

• FO-Model Counting: \( w(R) = w(\neg R) = 1 \)
• Apply inference rules backwards (step 4-3-2-1)

4. \[ \Delta = (\text{Stress}(\text{Alice}) \Rightarrow \text{Smokes}(\text{Alice})) \]

Domain = \{Alice\}
WFOMC Inference: Example

• FO-Model Counting: $w(R) = w(\neg R) = 1$
• Apply inference rules backwards (step 4-3-2-1)

4. $\Delta = (\text{Stress(Alice)} \Rightarrow \text{Smokes(Alice)})$

→ 3 models

Domain = \{Alice\}
WFOMC Inference: Example

• FO-Model Counting: $w(R) = w(\neg R) = 1$
• Apply inference rules backwards (step 4-3-2-1)

4. $\Delta = (\text{Stress(\text{Alice})} \Rightarrow \text{Smokes(\text{Alice})})$
   Domain = \{\text{Alice}\}
   $\rightarrow$ 3 models

3. $\Delta = \forall x, (\text{Stress}(x) \Rightarrow \text{Smokes}(x))$
   Domain = \{n \text{ people}\}
WFOMC Inference: Example

- FO-Model Counting: $w(R) = w(\neg R) = 1$
- Apply inference rules backwards (step 4-3-2-1)

4. \[ \Delta = (\text{Stress(Alice)} \Rightarrow \text{Smokes(Alice)}) \]
   \[ \rightarrow 3 \text{ models} \]

Domain = \{Alice\}

3. \[ \Delta = \forall x, (\text{Stress(x)} \Rightarrow \text{Smokes(x)}) \]
   \[ \rightarrow 3^n \text{ models} \]

Domain = \{n \text{ people}\}
WFOMC Inference: Example

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

\[ \rightarrow 3^n \text{ models} \]

Domain = \{n people\}
### WFOMC Inference: Example

<table>
<thead>
<tr>
<th>Step</th>
<th>Condition</th>
<th>Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.</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>3.</td>
<td>$\Delta = \forall x, (\text{Stress}(x) \Rightarrow \text{Smokes}(x))$</td>
<td>$\text{Domain} = {n \text{ people}}$</td>
</tr>
</tbody>
</table>

$\rightarrow 3^n$ models
WFOMC Inference: Example

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
</table>
| 2. | \[ \Delta = \forall y, (\text{ParentOf}(y) \land \text{Female} \Rightarrow \text{MotherOf}(y)) \]  
   | \[ \text{D} = \{n \text{ people}\} \]  
   |  
   | If Female = true? \[ \Delta = \forall y, (\text{ParentOf}(y) \Rightarrow \text{MotherOf}(y)) \]  
   | \[ \rightarrow 3^n \text{ models} \]  
| 3. | \[ \Delta = \forall x, (\text{Stress}(x) \Rightarrow \text{Smokes}(x)) \]  
   | \[ \text{Domain} = \{n \text{ people}\} \]  
   |  
   | \[ \rightarrow 3^n \text{ models} \]  


### WFOMC Inference: Example

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

- **Domain** = \{n people\}

\( \rightarrow 3^n \) models

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

- **D** = \{n people\}

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

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

3. \( \Delta = \forall x, (\text{Stress}(x) \Rightarrow \text{Smokes}(x)) \)  
   \( \rightarrow 3^n \) models
   Domain = \{n people\}

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

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

   \( \rightarrow 3^n + 4^n \) models
WFOMC Inference: Example

| 1.  | $\Delta = \forall x, y, (\text{ParentOf}(x,y) \land \text{Female}(x) \Rightarrow \text{MotherOf}(x,y))$ | $D = \{n \text{ people}\}$ |
| 2.  | $\Delta = \forall y, (\text{ParentOf}(y) \land \text{Female} \Rightarrow \text{MotherOf}(y))$ | $D = \{n \text{ people}\}$ |
|     | If Female = true? $\Delta = \forall y, (\text{ParentOf}(y) \Rightarrow \text{MotherOf}(y))$ | $\rightarrow 3^n$ models |
|     | If Female = false? $\Delta = \text{true}$ | $\rightarrow 4^n$ models |
|     | $\rightarrow 3^n + 4^n$ models |
| 3.  | $\Delta = \forall x, (\text{Stress}(x) \Rightarrow \text{Smokes}(x))$ | Domain = $\{n$ people$\}$ |
|     | $\rightarrow 3^n$ models |
### WFOMC Inference: Example

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

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

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

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.
\[ \Delta = \forall x, (\text{Stress}(x) \Rightarrow \text{Smokes}(x)) \]
\[ \rightarrow 3^n \text{ models} \]
Atom Counting: Example

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

Domain = \{n people\}
Atom Counting: Example

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

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
- ...

- Smokes
- Friends
- Smokes

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

Domain = \{n people\}
Atom Counting: Example

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

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
- ...

\[
\begin{align*}
\text{Smokes} & : k \\
\text{Friends} & : k \\
\text{Smokes} & : n-k \\
\end{align*}
\]
Atom Counting: Example

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

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
  ...

Diagram:

- Smokes (\( k \))
- Friends (\( k \))
- Smokes (\( n-k \))

(... graphical representation of relationships between Smokes, Friends, and Smokes)
Atom Counting: Example

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

- 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
- ...

Domain = \{n people\}
Atom Counting: Example

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
- ...
Atom Counting: Example

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

- 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
- ...

*Domain = \{n people\}
Atom Counting: Example

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

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
- ...

\[ \text{Smokes} \leftrightarrow \text{Friends} \leftrightarrow \text{Smokes} \]

\(k\) \quad \text{Smokes}
\n\(n-k\) \quad \text{Smokes}

\(k\) \quad \text{Friends}
\n\(n-k\) \quad \text{Friends}
Atom Counting: Example

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

- 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
- ...

Domain = \{n people\}
Atom Counting: Example

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

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
- ...

![Diagram showing the relationship between Smokes, Friends, and Smokes with numbers k and n-k]
Atom Counting: Example

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

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

Domain = \( \{n \text{ people}\} \)
**Atom Counting: Example**

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

- 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
- ...

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

- If we know that there are \( k \) smokers?
Atom Counting: Example

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

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

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

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

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

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

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

Domain = \{n people\}

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

**Database:**

<table>
<thead>
<tr>
<th></th>
<th>Smokes</th>
<th>Friends</th>
<th>Smokes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bob</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Charlie</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dave</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Eve</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

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

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

- In total...
Atom Counting: Example

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

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

- 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  
  ...

  \[ \rightarrow 2^{n^2 - k(n-k)} \] 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
First-Order Knowledge Compilation

Markov Logic

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

[Van den Broeck et al.; IJCAI’11, NIPS’11, PhD’13, KR’14]
First-Order Knowledge Compilation

Markov Logic

### FOL Sentence

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

### Weight Function

- \( w(\text{Smokes}) = 1 \)
- \( w(\neg \text{Smokes}) = 1 \)
- \( w(\text{Friends}) = 1 \)
- \( w(\neg \text{Friends}) = 1 \)
- \( w(F) = 3.14 \)
- \( w(\neg F) = 1 \)

3.14 \( \text{Smokes}(x) \land \text{Friends}(x,y) \Rightarrow \text{Smokes}(y) \)
First-Order Knowledge Compilation

Markov Logic

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

Weight Function

- \( w(\text{Smokes}) = 1 \)
- \( w(\neg \text{Smokes}) = 1 \)
- \( w(\text{Friends}) = 1 \)
- \( w(\neg \text{Friends}) = 1 \)
- \( w(F) = 3.14 \)
- \( w(\neg F) = 1 \)

FOL Sentence

\[ \forall x,y, F(x,y) \leftrightarrow \left[ \text{Smokes}(x) \land \text{Friends}(x,y) \Rightarrow \text{Smokes}(y) \right] \]

Compile

First-Order d-DNNF Circuit

[Van den Broeck et al.; IJCAI’11, NIPS’11, PhD’13, KR’14]
First-Order Knowledge Compilation

Markov Logic

\[ 3.14 \text{ Smokes}(x) \land \text{Friends}(x,y) \Rightarrow \text{Smokes}(y) \]

Weight Function

\[
\begin{align*}
  w(\text{Smokes}) &= 1 \\
  w(\neg \text{Smokes}) &= 1 \\
  w(\text{Friends}) &= 1 \\
  w(\neg \text{Friends}) &= 1 \\
  w(F) &= 3.14 \\
  w(\neg F) &= 1
\end{align*}
\]

FOL Sentence

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

Compile

First-Order d-DNNF Circuit

Domain

Alice
Bob
Charlie

[Van den Broeck et al.; IJCAI’11, NIPS’11, PhD’13, KR’14]
First-Order Knowledge Compilation

Markov Logic

3.14 Smokes(x) ∧ Friends(x,y) ⇒ Smokes(y)

Weight Function

\[
\begin{align*}
  w(\text{Smokes}) &= 1 \\
  w(\neg \text{Smokes}) &= 1 \\
  w(\text{Friends}) &= 1 \\
  w(\neg \text{Friends}) &= 1 \\
  w(F) &= 3.14 \\
  w(\neg F) &= 1
\end{align*}
\]

FOL Sentence

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

Compile

First-Order d-DNNF Circuit

Domain

Alice
Bob
Charlie

\[ Z = \text{WFOMC} = 1479.85 \]

[Van den Broeck et al.; IJCAI’11, NIPS’11, PhD’13, KR’14]
Let us automate this:

- **Relational** model

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

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

Let us automate this:

∀p, ∃c, Card(p,c)
∀c, ∃p, Card(p,c)
∀p, ∀c, ∀c', Card(p,c) ∧ Card(p,c') ⇒ c = c'

Playing Cards Revisited

Let us automate this:

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

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

[Van den Broeck.; AAAI-KR'15]
Playing Cards Revisited

Let us automate this:

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

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

Computed in time polynomial in \(n\)

Outline

• Motivation
  – Why high-level representations?
  – Why high-level reasoning?

• Intuition: Inference rules

• Liftability theory: Strengths and limitations

• Lifting in practice
  – Approximate symmetries
  – Lifted learning
Theory of Inference

Goal:
Understand complexity of probabilistic reasoning

• Low-level graph-based concepts (treewidth)
  ⇒ inadequate to describe high-level reasoning
• Need to develop “liftability theory”
• Deep connections to
  – database theory, finite model theory, 0-1 laws,
  – counting complexity

[Van den Broeck.; NIPS’11], [Van den Broeck, Jaeger.; StarAI’12]
Lifted Inference: Definition

• **Informal** [Poole’03, etc.]

  Exploit symmetries, Reason at first-order level, Reason about groups of objects, Scalable inference, High-level probabilistic reasoning, etc.

• **A formal definition:** Domain-lifted inference

  Inference runs in time polynomial in the number of entities in the domain.

[Van den Broeck.; NIPS’11]
Lifted Inference: Definition

- **Informal** [Poole’03, etc.]
  Exploit symmetries, Reason at first-order level, Reason about groups of objects, Scalable inference, High-level probabilistic reasoning, etc.

- **A formal definition:** **Domain-lifted inference**
  
  Inference runs in time polynomial in the number of entities in the domain.
  - Polynomial in #rows, #entities, #people, #webpages, #cards
  - ~ data complexity in databases

[Van den Broeck.; NIPS'11]
Lifted Inference: Definition

- **Informal [Poole’03, etc.]**
  
  Exploit symmetries, Reason at first-order level, Reason about groups of objects, Scalable inference, High-level probabilistic reasoning, etc.

- **A formal definition: Domain-lifted inference**
  
  Inference runs in time **polynomial in the number of entities in the domain**.

  - Polynomial in \#rows, \#entities, \#people, \#webpages, \#cards
  - \~ data complexity in databases

[Van den Broeck.; NIPS’11]
Lifted Inference: Definition

- **Informal** [Poole’03, etc.]
  Exploit symmetries, Reason at first-order level, Reason about groups of objects, Scalable inference, High-level probabilistic reasoning, etc.

- **A formal definition: Domain-lifted inference**

  Inference runs in time **polynomial** in the number of entities in the **domain**.

  - Polynomial in \#rows, \#entities, \#people, \#webpages, \#cards
  - \sim \ data complexity in databases

<table>
<thead>
<tr>
<th>Name</th>
<th>Cough</th>
<th>Asthma</th>
<th>Smokes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Bob</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Charlie</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
First-Order Knowledge Compilation

Markov Logic

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

Weight Function

- \( w(\text{Smokes}) = 1 \)
- \( w(\neg \text{Smokes}) = 1 \)
- \( w(\text{Friends}) = 1 \)
- \( w(\neg \text{Friends}) = 1 \)
- \( w(F) = 3.14 \)
- \( w(\neg F) = 1 \)

FOL Sentence

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

Compile?

First-Order \( d \)-DNNF Circuit

Domain

- Alice
- Bob
- Charlie

\[ Z = \text{WFOMC} = 1479.85 \]

[Van den Broeck.; NIPS'11]
First-Order Knowledge Compilation

Markov Logic

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

Weight Function

\[
\begin{align*}
\text{w(\text{Smokes})} &= 1 \\
\text{w(\neg \text{Smokes})} &= 1 \\
\text{w(\text{Friends})} &= 1 \\
\text{w(\neg \text{Friends})} &= 1 \\
\text{w(F)} &= 3.14 \\
\text{w(\neg F)} &= 1
\end{align*}
\]

FOL Sentence

\(\forall x,y, F(x,y) \iff [ \text{Smokes}(x) \land \text{Friends}(x,y) \implies \text{Smokes}(y) ]\)

Compile?

First-Order d-DNNF Circuit

Domain

Alice
Bob
Charlie

\(Z = \text{WFOMC} = 1479.85\)

Evaluation in time polynomial in domain size

[Van den Broeck.; NIPS'11]
First-Order Knowledge Compilation

**Markov Logic**

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

**Weight Function**

\[
\begin{align*}
    w(\text{Smokes}) &= 1 \\
    w(\neg \text{Smokes}) &= 1 \\
    w(\text{Friends}) &= 1 \\
    w(\neg \text{Friends}) &= 1 \\
    w(F) &= 3.14 \\
    w(\neg F) &= 1
\end{align*}
\]

**FOL Sentence**

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

**Compile?**

**First-Order d-DNNF Circuit**

**Domain**

Alice  Bob  Charlie

\( Z = \text{WFOMC} = 1479.85 \)

Evaluation in time polynomial in domain size

Domain-lifted!

[Van den Broeck.; NIPS'11]
First-Order Knowledge Compilation

Markov Logic

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

Weight Function

- \( w(\text{Smokes}) = 1 \)
- \( w(\neg \text{Smokes}) = 1 \)
- \( w(\text{Friends}) = 1 \)
- \( w(\neg \text{Friends}) = 1 \)
- \( w(F) = 3.14 \)
- \( w(\neg F) = 1 \)

FOL Sentence

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

Compile?

First-Order d-DNNF Circuit

Domain

Alice
Bob
Charlie

Evaluation in time polynomial in domain size
What Can Be Lifted?

**Theorem:** WFOMC for $FO^2$ is liftable

[Van den Broeck.; NIPS’11], [Van den Broeck et al.; KR’14]
What Can Be Lifted?

**Theorem:** WFOMC for $\text{FO}^2$ is liftable

**Corollary:** Markov logic with two logical variables per formula is liftable.

[Van den Broeck.; NIPS’11], [Van den Broeck et al.; KR’14]
What Can Be Lifted?

**Theorem:** WFOMC for $FO^2$ is liftable

**Corollary:** Markov logic with two logical variables per formula is liftable.

**Corollary:** Tight probabilistic logic programs with two logical variables are liftable.

... 

[Van den Broeck.; NIPS’11], [Van den Broeck et al.; KR’14]
$\text{FO}^2$ is liftable!

Properties
- Smokes(x)
- Gender(x)
- Young(x)
- Tall(x)

Properties
- Smokes(y)
- Gender(y)
- Young(y)
- Tall(y)
FO² is liftable!
"Smokers are more likely to be friends with other smokers."
"Colleagues of the same age are more likely to be friends."
"People are either family or friends, but never both."
"If X is family of Y, then Y is also family of X."
"If X is a parent of Y, then Y cannot be a parent of X."
**FO² is liftable!**

<table>
<thead>
<tr>
<th>Name</th>
<th>Cough</th>
<th>Asthma</th>
<th>Smokes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Bob</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Charlie</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Dave</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Eve</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Frank</td>
<td>1</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>

Medical Records

Statistical Relational Model in FO²

- 2.1 Asthma(x) ⇒ Cough(x)
- 3.5 Smokes(x) ⇒ Cough(x)
- 1.9 Smokes(x) ∧ Friends(x,y) ⇒ Smokes(y)
- 1.5 Asthma (x) ∧ Family(x,y) ⇒ Asthma (y)

[Van den Broeck.; NIPS’11], [Van den Broeck et al.; KR’14]
**FO² is liftable!**

Medical Records

<table>
<thead>
<tr>
<th>Name</th>
<th>Cough</th>
<th>Asthma</th>
<th>Smokes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Bob</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Charlie</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Dave</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Eve</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

| Frank  | 1     | ?      | ?      |

Statistical Relational Model in FO²

- 2.1 Asthma(x) $\Rightarrow$ Cough(x)
- 3.5 Smokes(x) $\Rightarrow$ Cough(x)
- 1.9 Smokes(x) $\land$ Friends(x,y) $\Rightarrow$ Smokes(y)
- 1.5 Asthma (x) $\land$ Family(x,y) $\Rightarrow$ Asthma (y)

Big data

[Van den Broeck.; NIPS’11], [Van den Broeck et al.; KR’14]
Can Everything Be Lifted?

[Beame, Van den Broeck, Gribkoff, Suciu; PODS’15]
Can Everything Be Lifted?

**Theorem:** There exists an FO$^3$ sentence $\Theta_1$ for which first-order model counting is $\#P_1$-complete in the domain size.

[Beame, Van den Broeck, Gribkoff, Suciu; PODS’15]
Can Everything Be Lifted?

**Theorem:** There exists an FO$^3$ sentence $\Theta_1$ for which first-order model counting is $\#P_1$-complete in the domain size.

A counting Turing machine is a nondeterministic TM that prints the number of its accepting computations.

The class $\#P_1$ consists of all functions computed by a polynomial-time counting TM with unary input alphabet.

**Proof:** Encode a universal $\#P_1$-TM in FO$^3$

[Beame, Van den Broeck, Gribkoff, Suciu; PODS’15]
Fertile Ground

Monadic

Y-acyclic CQs

Safe monotone CNF

Safe type-1 CNF

FO^3

FO^2 CNF

FO^2

FO CNF

Θ₁

Y₁

CQs

[VD; NIPS’11], [VD et al.; KR’14], [Gribkoff, VdB, Suciu; UAI’15], [Beame, VdB, Gribkoff, Suciu; PODS’15], etc.
\[ \Delta = \forall x, y, z, \text{Friends}(x, y) \land \text{Friends}(y, z) \Rightarrow \text{Friends}(x, z) \]

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

1. Independence

\[ P( ) = \] 

\[
\begin{array}{ccc}
\text{Name} & \text{Cough} & \text{Asthma} & \text{Smokes} \\
\hline
\text{Alice} & 1 & 1 & 0 \\
\text{Bob} & 0 & 0 & 0 \\
\text{Charlie} & 0 & 1 & 0 \\
\end{array}
\]

\[ = \] 

\[
\begin{array}{ccc}
\text{Name} & \text{Cough} & \text{Asthma} & \text{Smokes} \\
\hline
\text{Alice} & 1 & 1 & 0 \\
\text{Bob} & 0 & 0 & 0 \\
\text{Charlie} & 0 & 1 & 0 \\
\end{array}
\]

2. Partial Exchangeability

\[ P( ) = \] 

\[
\begin{array}{ccc}
\text{Name} & \text{Cough} & \text{Asthma} & \text{Smokes} \\
\hline
\text{Alice} & 1 & 1 & 0 \\
\text{Bob} & 0 & 0 & 0 \\
\text{Charlie} & 0 & 1 & 0 \\
\end{array}
\]

\[ = \] 

\[
\begin{array}{ccc}
\text{Name} & \text{Cough} & \text{Asthma} & \text{Smokes} \\
\hline
\text{Charlie} & 1 & 1 & 0 \\
\text{Alice} & 0 & 0 & 0 \\
\text{Bob} & 0 & 1 & 0 \\
\end{array}
\]

3. Independent and identically distributed (i.i.d.)

\[ = \] 

Independence + Partial Exchangeability
Statistical Properties for Tractability

- Tractable classes independent of representation
- Traditionally:
  - Tractable learning from \text{i.i.d. data}
  - Tractable inference when \text{cond. independence}
- New understanding:
  - Tractable learning from \text{exchangeable data}
  - Tractable inference when
    - \text{Conditional independence}
    - \text{Conditional exchangeability}
    - \text{A combination}

[Niepert, Van den Broeck; AAAI’14]
Outline

• Motivation
  – Why high-level representations?
  – Why high-level reasoning?
• Intuition: Inference rules
• Liftability theory: Strengths and limitations
• Lifting in practice
  – Approximate symmetries
  – Lifted learning
Approximate Symmetries

• What if not liftable? Asymmetric graph?

• Exploit approximate symmetries:
  – Exact symmetry $g$: $\Pr(x) = \Pr(x^g)$
    E.g. Ising model
    without external field
  – Approximate symmetry $g$: $\Pr(x) \approx \Pr(x^g)$
    E.g. Ising model with external field

$P \approx P$

[Van den Broeck, Darwiche; NIPS’13], [Van den Broeck, Niepert; AAAI’15]
Example: Statistical Relational Model

- WebKB: Classify pages given links and words
- Very large Markov logic network

1.3 $\text{Page}(x, \text{Faculty}) \Rightarrow \text{HasWord}(x, \text{Hours})$
1.5 $\text{Page}(x, \text{Faculty}) \land \text{Link}(x, y) \Rightarrow \text{Page}(y, \text{Course})$

and 5000 more ...

- No symmetries with evidence on Link or Word
- Where do approx. symmetries come from?

[Van den Broeck, Darwiche; NIPS’13], [Van den Broeck, Niepert; AAAI’15]
Over-Symmetric Approximations

- OSA makes model more symmetric
- E.g., low-rank Boolean matrix factorization

\[ \text{Link ("aaai.org", "google.com")} \quad \text{Link ("aaai.org", "google.com")} \]
\[ \text{Link ("google.com", "aaai.org")} \quad \text{Link ("google.com", "aaai.org")} \]
\[ \text{Link ("google.com", "gmail.com")} \quad \text{Link ("google.com", "gmail.com")} \]
\[ \text{Link ("ibm.com", "aaai.org")} \quad \text{Link ("ibm.com", "aaai.org")} \]

\[ \text{Link ("aaai.org", "ibm.com")} \quad \text{Link ("aaai.org", "ibm.com")} \]
\[ \text{Link ("ibm.com", "aaai.org")} \quad \text{Link ("ibm.com", "aaai.org")} \]

\[ \text{google.com and ibm.com become symmetric!} \]

[Van den Broeck, Darwiche; NIPS’13]
Experiments: WebKB

[Van den Broeck, Niepert; AAAI’15]
Outline

• Motivation
  – Why high-level representations?
  – Why high-level reasoning?
• Intuition: Inference rules
• Liftability theory: Strengths and limitations
• Lifting in practice
  – Approximate symmetries
  – Lifted learning
Lifted Weight Learning

- **Given:** A set of first-order logic formulas

\[ w \text{ FacultyPage}(x) \land \text{Linked}(x,y) \Rightarrow \text{CoursePage}(y) \]

A set of training databases

- **Learn:** The associated maximum-likelihood weights

\[ \frac{\partial}{\partial w_j} \log \Pr_w(db) = n_j(db) - \mathbb{E}_w[n_j] \]

Count in databases Efficient

Expected counts Requires inference

\[ \mathbb{E}_w[n_F] = \Pr(F\theta_1) + \cdots + \Pr(F\theta_m) \]

- **Idea:** Lift the computation of \( \mathbb{E}_w[n_j] \)

[Van den Broeck et al.; StarAI’13+]
Learning Time

Learns a model over 900,030,000 random variables

\[ w \text{ Smokes}(x) \land \text{Friends}(x,y) \implies \text{Smokes}(y) \]

Big data

Learns a model over 900,030,000 random variables

[Van den Broeck et al.; StarAI’13]
Learning Time

\[ w \text{ Smokes}(x) \land \text{Friends}(x,y) \implies \text{Smokes}(y) \]

Learns a model over 900,030,000 random variables

[Van den Broeck et al.; StarAI’13]
## Lifted Structure Learning

- **Given:** A set of training databases
- **Learn:** A set of first-order logic formulas
  The associated maximum likelihood weights
- **Idea:** Learn liftable models (regularize with symmetry)

<table>
<thead>
<tr>
<th></th>
<th>IMDb</th>
<th>UWCSE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Baseline</td>
<td>Lifted Weight Learning</td>
</tr>
<tr>
<td>Fold 1</td>
<td>-548</td>
<td>-378</td>
</tr>
<tr>
<td>Fold 2</td>
<td>-689</td>
<td>-390</td>
</tr>
<tr>
<td>Fold 3</td>
<td>-1,157</td>
<td>-851</td>
</tr>
<tr>
<td>Fold 4</td>
<td>-415</td>
<td>-285</td>
</tr>
<tr>
<td>Fold 5</td>
<td>-413</td>
<td>-267</td>
</tr>
</tbody>
</table>

[VHaaren, Van den Broeck, et al.;’15]
Outline

• Motivation
  – Why high-level representations?
  – Why high-level reasoning?
• Intuition: Inference rules
• Liftability theory: Strengths and limitations
• Lifting in practice
  – Lifted learning
  – Approximate symmetries
Conclusions

- A radically new reasoning paradigm
- Lifted inference is **frontier** and **integration** of AI, KR, ML, DBs, theory, etc.
- We need
  - relational databases and logic
  - probabilistic models and statistical learning
  - algorithms that scale
- Many theoretical open problems
- It works in practice
Long-Term Outlook

Probabilistic inference and learning exploit

~ 1988: conditional independence

~ 2000: contextual independence (local structure)
Long-Term Outlook

Probabilistic inference and learning exploit

~ 1988: conditional independence
~ 2000: contextual independence (local structure)
~ 201?: symmetry & exchangeability
### Collaborators

<table>
<thead>
<tr>
<th>KU Leuven</th>
<th>UCLA</th>
<th>Indiana Univ.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Luc De Raedt</td>
<td>Adnan Darwiche</td>
<td>Sriraam Natarajan</td>
</tr>
<tr>
<td>Wannes Meert</td>
<td>Arthur Choi</td>
<td></td>
</tr>
<tr>
<td>Jesse Davis</td>
<td>Doga Kisa</td>
<td></td>
</tr>
<tr>
<td>Hendrik Blockeel</td>
<td>Karthika Mohan</td>
<td></td>
</tr>
<tr>
<td>Daan Fierens</td>
<td>Judea Pearl</td>
<td></td>
</tr>
<tr>
<td>Angelika Kimmig</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nima Taghipour</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kurt Driessens</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jan Ramon</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maurice Bruynooghe</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Siegfried Nijssen</td>
<td>Mathias Niepert</td>
<td></td>
</tr>
<tr>
<td>Jessa Bekker</td>
<td>Dan Suciu</td>
<td></td>
</tr>
<tr>
<td>Ingo Thon</td>
<td>Eric Gribkoff</td>
<td></td>
</tr>
<tr>
<td>Bernd Gutmann</td>
<td>Paul Beame</td>
<td></td>
</tr>
<tr>
<td>Vaishak Belle</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Joris Renkens</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Davide Nitti</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bart Bogaerts</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jonas Vlasselaer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jan Van Haaren</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Prototype Implementation

1. Read .MLN, WMC, FOPI
2. Compile WFOMC
3. Infer FO-d-DNNF
4. Learn weights
5. Export .d-DNNF, Facts, Dimacs

Pr(Friends(Guy, Wannes)) = 0.70
Pr(Friends(Jesse, Guy)) = 0.75
...

2.3 Smokes(x) ∧ Friends(x, y) → Smokes(y)

http://dtai.cs.kuleuven.be/wfomc
Thanks