Brief Tutorial on Probabilistic Databases

Dan Suciu
University of Washington
About This Talk

- Probabilistic databases
  - Tuple-independent
  - Query evaluation
- Statistical relational models
  - Representation, learning, inference in FO
  - Reasoning/learning = lifted inference
- Sources:
  - Book 2011 [S., Olteanu, Re, Koch]
  - Upcoming F&T survey [van Den Broek, S]
Background: Relational databases

Database \( D = \)

<table>
<thead>
<tr>
<th>Smoker</th>
<th>Friend</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Alice</td>
<td>Alice</td>
</tr>
<tr>
<td>2009</td>
<td>Bob</td>
</tr>
<tr>
<td>Alice</td>
<td>Alice</td>
</tr>
<tr>
<td>2010</td>
<td>Carol</td>
</tr>
<tr>
<td>Bob</td>
<td>Bob</td>
</tr>
<tr>
<td>Carol</td>
<td>Carol</td>
</tr>
<tr>
<td>2009</td>
<td>2010</td>
</tr>
</tbody>
</table>

Query: \( Q(z) = \exists x \ (\text{Smoker}(x,'2009') \land \text{Friend}(x,z)) \)

Constraint: \( Q = \forall x \ (\text{Smoker}(x,'2010') \Rightarrow \text{Friend}(x,'Bob')) \)

\( Q(D) = \text{true} \)
**Probabilistic Database**

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></td>
<td>$p_1$</td>
</tr>
<tr>
<td>a1</td>
<td>b2</td>
<td></td>
<td>$p_2$</td>
</tr>
<tr>
<td>a2</td>
<td>b2</td>
<td></td>
<td>$p_3$</td>
</tr>
</tbody>
</table>

Possible worlds semantics:

$$
\sum_{\text{world}} P_D(\text{world}) = 1
$$

$$
P_D(Q) = \sum_{\text{world} \models Q} P_D(\text{world})
$$

$$
(1-p_1)p_2p_3
$$

$$
(1-p_1)(1-p_2)(1-p_3)
$$
Outline

• Model Counting
• Small Dichotomy Theorem
• Dichotomy Theorem
• Query Compilation
• Conclusions, Open Problems
Model Counting

• Given propositional Boolean formula $F$, compute the number of models $\#F$

Example:
$F = (X_1 \lor X_2) \land (X_2 \lor X_3) \land (X_3 \lor X_1)$

$\#F = 4$

[valiant'79] #P-hard, even for 2CNF
Probability of a Formula

- Each variable $X$ has a probability $p(X)$;
- $P(F) =$ probability that $F =$true, when each $X$ is set to true independently

Example:
$F = (X_1 \lor X_2) \land (X_2 \lor X_3) \land (X_3 \lor X_1)$

$P(F) = (1-p_1)p_2p_3 + p_1(1-p_2)p_3 + p_1p_2(1-p_3) + p_1p_2p_3$

If $p(X) = \frac{1}{2}$ for all $X$, then $P(F) = \#F / 2^n$
Algorithms for Model Counting

[Gomes, Sabharwal, Selman’2009]
Based on full search DPLL:

- **Shannon expansion.**
  \[ \#F = \#F[X=0] + \#F[X=1] \]

- **Caching.**
  Store \#F, look it up later

- **Components.** If \( \text{Vars}(F_1) \cap \text{Vars}(F_2) = \emptyset \):
  \[ \#(F_1 \land F_2) = \#F_1 \ast \#F_2 \]
Relational Representation (1/2)

• Fix an FO sentence $Q$ and a domain $\Delta$
• Ground atom $\rightarrow$ Boolean variable

**Definition** The lineage $F_{Q,\Delta}$ is:

- $F_{Q,\Delta} = Q$ if $Q$ = ground atom
- $F_{Q_1 \land Q_2,\Delta} = F_{Q_1,\Delta} \land F_{Q_2,\Delta}$ same for $\lor$, $\rightarrow$, $\neg$
- $F_{\forall x.Q,\Delta} = \bigwedge_{a \in \Delta} F_{Q[a/x],\Delta}$
- $F_{\exists x.Q,\Delta} = \bigvee_{a \in \Delta} F_{Q[a/x],\Delta}$

$Q = \forall x \ (\text{Student}(x) \Rightarrow \text{Person}(x))$

$F_{Q,[n]} = (\text{Student}(1) \Rightarrow \text{Person}(1)) \land \ldots \land (\text{Student}(n) \Rightarrow \text{Person}(n))$
Relational Representation (2/2)

- For a database $D$, denote

$$F_{Q,D} = F_{Q,\text{domain}(D)}$$

where all tuples not in $D$ are set to $false$

- $F_{Q,\Delta}$ or $F_{Q,D}$ is called the \textit{lineage} or the \textit{provenance} or the \textit{grounding} of $Q$
Weighted FO Model Counting

• Probabilities of ground atoms in $D = \text{probabilities of Boolean variables } p(X)$

• Fix $Q$. Given $D$, compute $P(F_{Q,D})$

• Simple fact: $P_D(Q) = P(F_{Q,D})$

<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>
This Talk

Fix a query $Q$:

- What is the complexity of $P_D(Q)$ in the size of $D$?

- What is the best runtime of a DPLL-based algorithm on $F_{Q,D}$ in the size of $D$?
Discussion: Correlations

[ Domingos & Richardson ’06 ] MLN = popular FO framework for Machine Learning tasks

\[ \text{Smoker}(x) \land \text{Friends}(x, y) \rightarrow \text{Smoker}(y), \text{ weight } = 2.3 \]

**Theorem** [ Jha, S’11 ] One can construct effectively \( D \) s.t.

\[
P_{\text{MLN}, \Delta}(Q) = P_D(Q | \Gamma) = \frac{P_D(Q \land \Gamma)}{P_D(\Gamma)}
\]
Outline

• Model Counting

• Small Dichotomy Theorem

• Dichotomy Theorem

• Query Compilation

• Conclusions, Open Problems
Background: Query Plans

\[ Q(z) = R(z,x), S(x,y), T(y,u), u=123 \]

Query plan = expressions over the input relation

Operators = selection, projection, join, union, difference

\[ \Pi_z \]

\[ \bowtie_x \]

\[ \sigma_{u=123} \]

\[ R(z,x) \]

\[ S(x,y) \]

\[ T(y,u) \]
An Example

Boolean query

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

\[ P_D(Q) = 1 - \left\{ 1 - p_1^* \left[ 1 - (1-q_1)^* (1-q_2) \right] \right\} * \]

\[ \left\{ 1 - p_2^* \left[ 1 - (1-q_3)^* (1-q_4)^* (1-q_5) \right] \right\} \]

One can compute \( P_D(Q) \) in \( \text{PTIME} \) in the size of the 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>q1</td>
</tr>
<tr>
<td>a1</td>
<td>b2</td>
<td>q2</td>
</tr>
<tr>
<td>a2</td>
<td>b3</td>
<td>q3</td>
</tr>
<tr>
<td>a2</td>
<td>b4</td>
<td>q4</td>
</tr>
<tr>
<td>a2</td>
<td>b5</td>
<td>q5</td>
</tr>
</tbody>
</table>
Extensional Plans

• Modify each operator to compute output probabilities, assuming independent events
\[ Q() = R(x), \ S(x,y) \]

\[ 1-(1-p_1 q_1)(1-p_1 q_2)(1-p_2 q_3)(1-p_2 q_4)(1-p_2 q_5) \]

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

\[ 1-{1-p_1[1-(1-q_1)(1-q_2)]}* \]
\[ {1-p_2[1-(1-q_4)(1-q_5) (1-q_6)]} \]
Safe Queries

**Definition** A plan for $Q$ is *safe* if it computes the probabilities correctly.

$Q$ is *safe* if it has a safe plan.

- In AI, computing $Q$ using a safe plan is called *lifted inference*
- **Safe query** = **Liftable query**

- If $Q$ is safe then $P_D(Q)$ is in PTIME
Unsafe Queries

\[ H_0() = R(x), S(x,y), T(y) \]

**Theorem.** [Dalvi&S.2004] \( P_D(H_0) \) is \#P-hard

However:
1. This plan computes an upper bound [VLDB’15]
2. Use samples on T [VLDB’16]

Wolfgang Gatterbauer’s talk today
Hierarchical Queries

Fix $Q$; $at(x)$ = set of atoms (=literals) containing the variable $x$

**Definition** $Q$ is hierarchical if for all variables $x, y$:

$at(x) \subseteq at(y)$ or $at(x) \supseteq at(y)$ or $at(x) \cap at(y) = \emptyset$

Hierarchical

$Q() = R(x,y), S(x,z)$

Non-hierarchical

$H_0() = R(x), S(x,y), T(y)$
The Small Dichotomy Theorem

Non-repeating Conjunctive Query =
  = Conjunctive Query “without self-joins”
  = “Simple” conjunctive query

[Dalvi&S.04]

**Theorem** Let $Q$ be a non-repeating CQ
- If $Q$ is hierarchical, then $P_D(Q)$ is in PTIME.
- If $Q$ is not hierarchical then $P_D(Q)$ is $\#P$-hard.

By duality, the same holds for a non-repeating clause
## Summary so Far

| Complexity of $P_D(Q)$ | Non-repeating CQ  
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Non-repeating clause</td>
</tr>
<tr>
<td><strong>PTIME</strong></td>
<td>Hierarchical</td>
</tr>
<tr>
<td><strong>#P - hard</strong></td>
<td>Non-hierarchical</td>
</tr>
</tbody>
</table>
Outline

• Model Counting
• Small Dichotomy Theorem
  • Dichotomy Theorem
• Query Compilation
• Conclusions, Open Problems
The Rules for Lifted Inference

Preprocess $Q$ (omitted from this talk; see book), then apply these rules (some have preconditions)

\[
P(\neg Q) = 1 - P(Q)\quad \text{negation}
\]

\[
P(Q_1 \land Q_2) = P(Q_1)P(Q_2)\]
\[
P(Q_1 \lor Q_2) = 1 - (1 - P(Q_1))(1 - P(Q_2))\quad \text{Independent join / union}
\]

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

\[
P(Q_1 \land Q_2) = P(Q_1) + P(Q_2) - P(Q_1 \lor Q_2)\]
\[
P(Q_1 \lor Q_2) = P(Q_1) + P(Q_2) - P(Q_1 \land Q_2)\quad \text{Inclusion/exclusion}
\]
$\text{FO}^{\text{un}} = \text{Unate FO}$

An FO sentence is \textit{unate} if:

- Negations occur only on atoms
- Every relational symbol $R$ either occurs only positively, or only negatively

$\text{FO}^{\text{un}} = \text{FO restricted to unate sentences}$
Dichotomy Theorem

[Dalvi&S’12]

**Theorem** For any $Q$ in $\forall^*\text{FO}^{\text{un}}$ (or $\exists^*\text{FO}^{\text{un}}$)
- If rules succeed, then $P_D(Q)$ in PTIME in $|D|$
- If rules fail, then $P_D(Q)$ is $\#P$ hard in $|D|$

Note: Unions of Conjunctive queries (UCQ) is essentially $\exists^*\text{FO}^{\text{un}}$
Example: Liftable Query

\[ Q_J() = S(x_1,y_1), R(y_1), S(x_2,y_2), T(y_2) \]

\[ = [S(x_1,y_1),R(y_1)] \land [S(x_2,y_2),T(y_2)] \]

\[ P(Q_J) = P(Q_1) + P(Q_2) - P(Q_1 \lor Q_2) \]

\[ Q_1 \lor Q_2 = \exists y [S(x_1,y),R(y) \lor S(x_2,y)),T(y)] \]

\[ P(Q_1 \lor Q_2) = \]

\[ = 1 - \prod_{b \in \text{Domain}} (1 - P[S(x_1,b), R(b) \lor S(x_2,b)), T(b)]) \]

\[ = 1 - \prod_{b \in \text{Domain}} (1 - P[S(x_1,b)] \ast P[R(b) \lor T(b)]) = \ldots \text{ etc} \]

Runtime = \( O(n^2) \).

Simons 2016
**Example: Liftable Query**

\[ Q_J = \forall x_1 \forall y_1 \forall x_2 \forall y_2 (S(x_1,y_1) \lor R(y_1) \lor S(x_2,y_2) \lor T(y_2)) \]

\[ = [\forall x_1 \forall y_1 S(x_1,y_1) \lor R(y_1)] \lor [\forall x_2 \forall y_2 S(x_2,y_2) \lor T(y_2)] \]

\[ P(Q_J) = P(Q_1) + P(Q_2) - P(Q_1 \land Q_2) \]

PTIME (have seen before)

\[ y = y_1 = y_2 \]

\[ Q_1 \land Q_2 = \forall y [(\forall x_1 S(x_1,y) \lor R(y)) \land (\forall x_2 S(x_2,y)) \lor T(y)] \]

\[ = \forall y [\forall x S(x,y) \lor (R(y) \land T(y))] \]

\[ P(Q_1 \land Q_2) = \prod_{b \in \text{Domain}} P[\forall x . S(x,b) \lor (R(b) \land T(b))] = \ldots \text{etc} \]

Runtime = \(O(n^2)\).
Unliftable Queries $H_k$

$H_0 = R(x) \vee S(x,y) \vee T(y)$

$H_1 = [R(x_0) \vee S(x_0,y_0)] \land [S(x_1,y_1) \vee T(y_1)]$

$H_2 = [R(x_0) \vee S_1(x_0,y_0)] \land [S_1(x_1,y_1) \vee S_2(x_1,y_1)] \lor [S_2(x_2,y_2) \vee T(y_2)]$

$H_3 = [R(x_0) \vee S_1(x_0,y_0)] \land [S_1(x_1,y_1) \vee S_2(x_1,y_1)] \land [S_2(x_2,y_2) \vee S_3(x_2,y_2)] \land [S_3(x_3,y_3) \vee T(y_3)]$

$\ldots$

Every $H_k$, $k \geq 1$ is hierarchical

Theorem. [Dalvi&S’12] Every query $H_k$ is $\#P$-hard
A Closer Look at $H_k$

If we drop any one clause $\rightarrow$ in PTIME

$$H_3 = [R(x_0) \lor S_1(x_0, y_0)] \land [S_1(x_1, y_1) \lor S_2(x_1, y_1)] \land [S_2(x_2, y_2) \lor S_3(x_2, y_2)] \land [S_3(x_3, y_3) \lor T(y_3)]$$

Independent join
## Summary so Far

<table>
<thead>
<tr>
<th>Complexity of $P_D(Q)$</th>
<th>Non-repeating CQ</th>
<th>Non-repeating clauses</th>
<th>$\exists^* \text{FO}^{un}$</th>
<th>$\forall^* \text{FO}^{un}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>PTIME</td>
<td>Hierarchical</td>
<td></td>
<td>Rules succeed</td>
<td></td>
</tr>
<tr>
<td>#P - hard</td>
<td>Non-hierarchical</td>
<td></td>
<td>Rules fail</td>
<td></td>
</tr>
</tbody>
</table>
Outline

• Model Counting
• Small Dichotomy Theorem
• Dichotomy Theorem
• Query Compilation
• Conclusions, Open Problems
Lifted v.s. Grounded Inference

• To compute $P_D(Q)$:
  compute the lineage $F_{Q,D}$
  use DPLL-based algorithm for $P(F_{Q,D})$

• For which queries $Q$ can this be in PTIME?

• [Huang&Darwiche’2005] The trace of a DPLL-based algorithm is “decision-DNNF”
**Def** [Darwiche] A Decision-DNNF is a rooted DAG where:
- Internal nodes are decision or $\land$
- Sink nodes are 0 or 1

Children of $\land$ have disjoint sets of variables

Every root-to-sink path reads each variable at most once
Notations

\( H_{k0} = \forall x \forall y \, R(x) \lor S_1(x,y) \)
\( H_{k1} = \forall x \forall y \, S_1(x,y) \lor S_2(x,y) \)
\( H_{k2} = \forall x \forall y \, S_2(x,y) \lor S_3(x,y) \)
...
...
\( H_{kk} = \forall x \forall y \, S_k(x,y) \lor T(y) \)

f(\( Z_0, Z_1, \ldots, Z_k \)) = \) a Boolean function

Example: \( f = Z_0 \land Z_1 \land \ldots \land Z_k \) then \( f(H_{k0}, H_{k1}, \ldots, H_{kk}) = H_k \)
Easy/Hard Queries

[Beame’14]

**Theorem** Let $Q = f(H_{k0}, H_{k1}, \ldots, H_{kk})$ where $f(Z_0, Z_1, \ldots, Z_k)$ is a monotone Boolean function.

- Any Decision-DNNF for $F_{Q,[n]}$ has size $2^\Omega(\sqrt{n})$.
- $P_D(Q)$ is in PTIME iff $\mu_Q(0, 1) = 0$

$\mu = \text{Möbius function of the implicates lattice of } Q$

Consequence: Any DPLL-based algorithm takes time $2^\Omega(\sqrt{n})$, even if the query is in PTIME!
Cancellations

\[ Q_W = (H_{30} \land H_{32}) \lor (H_{30} \land H_{33}) \lor (H_{31} \land H_{33}) \]

\[ H_{30} = \forall x \forall y \ R(x) \lor S_1(x, y) \]
\[ H_{31} = \forall x \forall y \ S_1(x, y) \lor S_2(x, y) \]
\[ H_{32} = \forall x \forall y \ S_2(x, y) \lor S_3(x, y) \]
\[ H_{33} = \forall x \forall y \ S_3(x, y) \lor T(y) \]

\[ P(Q_W) = P(H_{30} \land H_{32}) + P(H_{30} \land H_{33}) + P(H_{31} \land H_{33}) + \]
\[ - P(H_{30} \land H_{32} \land H_{33}) - P(H_{30} \land H_{31} \land H_{33}) \]
\[ - P(H_{30} \land H_{31} \land H_{32} \land H_{33}) \]
\[ + P(H_{30} \land H_{31} \land H_{32} \land H_{33}) \]

Also = \( H_3 \)

\( P(Q_W) \) is in PTIME
The CNF Lattice

**Definition.** The DNF lattice of $Q = Q_1 \lor Q_2 \lor \ldots$ is:
- Elements are prime implicants
- Order is implication

$$Q_W = (H_{30} \land H_{32}) \lor (H_{30} \land H_{33}) \lor (H_{31} \land H_{33})$$
The Möbius’ Function

**Def.** The Möbius function:

\[ \mu(1,1) = 1 \quad \mu(u,1) = -\sum_{u < v \leq 1} \mu(v,1) \]

**Möbius’ Inversion Formula:**

\[ P(Q) = -\sum_{Q_i < 1} \mu(Q_i,1) P(Q_i) \]
The Möbius’ Function

**Def.** The Möbius function:
\[ \mu(1,1) = 1 \quad \mu(u,1) = -\sum_{u<v\leq 1} \mu(v,1) \]

**Möbius’ Inversion Formula:**
\[ P(Q) = -\sum_{Q_i<1} \mu(Q_i,1) P(Q_i) \]
The Möbius’ Function

**Def. The Möbius function:**

\[ \mu(1, 1) = 1 \quad \mu(u, 1) = -\sum_{u < v \leq 1} \mu(v, 1) \]

**Möbius’ Inversion Formula:**

\[ P(Q) = -\sum Qi < 1 \mu(Qi, 1) P(Qi) \]
The Möbius’ Function

Def. The Möbius function:

\[ \mu(1, 1) = 1 \]
\[ \mu(u, 1) = -\sum_{u < v \leq 1} \mu(v, 1) \]

Möbius’ Inversion Formula:

\[ P(Q) = -\sum_{Q_i < 1} \mu(Q_i, 1) P(Q_i) \]
The Möbius’ Function

Def. The Möbius function:
\[
\begin{align*}
\mu(1, 1) &= 1 \\
\mu(u, 1) &= -\sum_{u < v \leq 1} \mu(v, 1)
\end{align*}
\]

Möbius’ Inversion Formula:
\[
P(Q) = -\sum_{Q_i < 1} \mu(Q_i, 1) P(Q_i)
\]
The Möbius’ Function

**Def. The Möbius function:**

\[
\begin{align*}
\mu(1,1) &= 1 \\
\mu(u,1) &= -\sum_{u < v \leq 1} \mu(v,1)
\end{align*}
\]

**Möbius’ Inversion Formula:**

\[
P(Q) = \sum_{Q_i < 1} \mu(Q_i,1) P(Q_i)
\]
## Summary

<table>
<thead>
<tr>
<th></th>
<th>nr CQ nr clause</th>
<th>$\exists^* \text{FO}^\text{un}$ $\forall^* \text{FO}^\text{un}$</th>
<th>$f(H_{k0}, \ldots, H_{kk})$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PTIME</strong></td>
<td>Hierarchical</td>
<td>Rules succeed</td>
<td>$\mu(0,1) = 0$</td>
</tr>
<tr>
<td><strong>#P-hard</strong></td>
<td>Non-hierarchical</td>
<td>Rules fail</td>
<td>$\mu(0,1) \neq 0$</td>
</tr>
<tr>
<td><strong>DPLL</strong></td>
<td></td>
<td></td>
<td>$2^{\Omega(\sqrt{n})}$</td>
</tr>
</tbody>
</table>

Simons 2016
Möbius Über Alles

∀\*^{FO}_{un} \text{ or } \exists\*^{FO}_{un} (\approx UCQ)

PTIME

Poly-size FBDD, dec-DNNF

Poly-size OBDD, SDD = inversion-free

Read Once

Open
Outline

• Model Counting
• Small Dichotomy Theorem
• Dichotomy Theorem
• Query Compilation

• Conclusions, Open Problems
Summary

• Query evaluation on probabilistic databases = weighted model counting
• Each query $Q$ defines a different WMC problem
• Dichotomy: depending on $Q$, WMC is in PTIME or $\#P$-hard
• Using a DPLL-based algorithm on the grounded $Q$ is suboptimal
Discussion: Extensions

Open problems: extend the dichotomy theorem to:

• Mixed probabilistic/deterministic relations
• Functional dependencies
• Interpreted predicates: $<$, $\neq$

Open problem: complexity of MAP
Discussion: Symmetric Relations

• A relation $R$ is symmetric if all ground tuples have the same probability

• [van den Broeck’14] For every $Q$ in $\text{FO}^2$, $P(Q)$ is in PTIME on symmetric databases.

• [Beame’15] Hardness results.

• In general the complexity is open
Discussion: Negation

[Fink&Olteanu’14] Restrict FO to non-repeating expressions
• Theorem Hierarchical expressions are in PTIME, non-hierarchical are #P-hard.

[Gribkoff,S.,v.d.Broeck’14] ∀*FO or ∃*FO
• Need resolution compute some queries with negation

Open problem: completeness/dichotomy?
Thank You!
BACKUP
Weighted Model Counting

- Each variable \( X \) has a weight \( w(X) \);
- Weight of a model = \( \prod_{X=\text{true}} w(X) \);
- \( \text{WMC}(F) = \text{sum of weights of models of } F \)

Example:
\( F = (X_1 \lor X_2) \land (X_2 \lor X_3) \land (X_3 \lor X_1) \)

\[
\text{WMC}(F) = w_2 \cdot w_3 + w_1 \cdot w_3 + w_1 \cdot w_2 + w_1 \cdot w_2 \cdot w_3
\]

Set \( w(X) = 1 \): then \( \text{WMC}(F) = \#F \)
Probability of a Formula

• Each variable \(X\) has a probability \(p(X)\);
• \(P(F)\) = probability that \(F=\text{true}\), when each \(X\) is set to \(\text{true}\) independently.

Example:
\(F = (X_1 \lor X_2) \land (X_2 \lor X_3) \land (X_3 \lor X_1)\)

\[
P(F) = (1-p_1)p_2p_3 + p_1(1-p_2)p_3 + p_1p_2(1-p_3) + p_1p_2p_3
\]

Set \(w(X) = p(X)/(1-p(X))\)
Then \(P(F) = WMC(F) / Z\), where \(Z = \prod_X (1+w(X))\)
Discussion: Dichotomy for #SAT

• [Creignou&Hemann’96] consider model counting #F where the formula F is given by general\textit{ized clauses}

• Dichotomy Theorem: #F is in PTIME when all clauses are affine, #P-hard otherwise

• Not helpful in our context:
  – If Q is a UCQ, then the clauses of F_{Q,D} are of the form X \lor Y \lor Z \ldots and are not affine;
  – But P(F_{Q,D}) is not always #P-hard, because Q restricts the structure of the clauses
Warm-up: Weights

Replace probabilities with weights:

<table>
<thead>
<tr>
<th>R:</th>
<th></th>
<th>S:</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>y</td>
<td>w</td>
</tr>
<tr>
<td>a1</td>
<td>b1</td>
<td>w₁</td>
</tr>
<tr>
<td>a2</td>
<td>b1</td>
<td>w₂</td>
</tr>
<tr>
<td>a3</td>
<td>b2</td>
<td>w₃</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>y</th>
<th>w</th>
</tr>
</thead>
<tbody>
<tr>
<td>b1</td>
<td>v₁</td>
</tr>
<tr>
<td>b2</td>
<td>v₂</td>
</tr>
<tr>
<td>b3</td>
<td>v₃</td>
</tr>
</tbody>
</table>

Weight of a possible world:

Weight( ) = w₁w₂v₂

P_D(world) = Weight(world)/Z

Z = Σ_{world'} Weight(world')

Z = (1+v₁)(1+v₂)(1+v₃)(1+w₁)(1+w₂)(1+w₃)
Markov Logic Networks

Replace probabilities with weights:

\[
R(x, y) \Rightarrow S(y) \quad w_4
\]

Add soft constraints:

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>w</th>
</tr>
</thead>
<tbody>
<tr>
<td>a1</td>
<td>b1</td>
<td>( w_1 )</td>
</tr>
<tr>
<td>a2</td>
<td>b1</td>
<td>( w_2 )</td>
</tr>
<tr>
<td>a3</td>
<td>b2</td>
<td>( w_3 )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>y</th>
<th>w</th>
</tr>
</thead>
<tbody>
<tr>
<td>b1</td>
<td>( v_1 )</td>
</tr>
<tr>
<td>b2</td>
<td>( v_2 )</td>
</tr>
<tr>
<td>b3</td>
<td>( v_3 )</td>
</tr>
</tbody>
</table>
Markov Logic Networks

Replace probabilities with weights:

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>w</th>
</tr>
</thead>
<tbody>
<tr>
<td>a1</td>
<td>b1</td>
<td>w₁</td>
</tr>
<tr>
<td>a2</td>
<td>b1</td>
<td>w₂</td>
</tr>
<tr>
<td>a3</td>
<td>b2</td>
<td>w₃</td>
</tr>
</tbody>
</table>

Add soft constraints:

\[ R(x,y) \Rightarrow S(y) \quad w_4 \]

Weight of a possible world:

\[
\text{Weight}(\begin{array}{|c|c|}
\hline
x & y \\
\hline
a1 & b1 \\
\hline
a3 & b2 \\
\hline
\end{array}) = w₁w₃v₁w₄w₄w₄w₄w₄w₄ \]
Markov Logic Networks

Replace probabilities with weights:

Add soft constraints:

\[ P_{\text{MLN}}(\text{world}) = \frac{\text{Weight}(\text{world})}{Z} \]

\[ Z = \sum_{\text{world'}} \text{Weight}(\text{world'}) \]

Weight of a possible world:

Weight( ) = \( w_1w_3v_1w_4w_4w_4w_4w_4w_4 \)
Markov Logic Networks

Replace probabilities with weights:

Weight( ) = \w_1 \w_3 \v_1 \w_4 \w_4 \w_4 \w_4 \w_4

Add soft constraints:

\text{R}(x,y) \implies \text{S}(y) \quad \w_4

\text{P}_\text{MLN}(\text{world}) = \frac{\text{Weight(world)}}{Z}

Z = \sum_{\text{world'}} \text{Weight(} \text{world'})

Z = (1+\v_1) (1+\v_2) (1+\v_3) (1+\w_1) (1+\w_2) (1+\w_3)

Z is \#P-hard to compute

Weight of a possible world:

\text{Weight( )} = \w_1 \w_3 \v_1 \w_4 \w_4 \w_4 \w_4 \w_4

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>w</th>
</tr>
</thead>
<tbody>
<tr>
<td>a1</td>
<td>b1</td>
<td>\w_1</td>
</tr>
<tr>
<td>a2</td>
<td>b1</td>
<td>\w_2</td>
</tr>
<tr>
<td>a3</td>
<td>b2</td>
<td>\w_3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>w</th>
</tr>
</thead>
<tbody>
<tr>
<td>b1</td>
<td></td>
<td>\v_1</td>
</tr>
<tr>
<td>b2</td>
<td></td>
<td>\v_2</td>
</tr>
<tr>
<td>b3</td>
<td></td>
<td>\v_3</td>
</tr>
</tbody>
</table>

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>y</td>
<td></td>
</tr>
<tr>
<td>a1</td>
<td>b1</td>
<td></td>
</tr>
<tr>
<td>a3</td>
<td>b2</td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>y</td>
<td>w</td>
<td></td>
</tr>
<tr>
<td>b1</td>
<td></td>
<td>\v_1</td>
</tr>
<tr>
<td>b2</td>
<td></td>
<td>\v_2</td>
</tr>
<tr>
<td>b3</td>
<td></td>
<td>\v_3</td>
</tr>
</tbody>
</table>
Discussion

Weights v.s. probabilities

• Soft constraints *with probabilities* may be inconsistent

• Soft constraints *with weights* (\( \neq 0, \infty \)) always consistent

Weight values have no semantics!

• Learned from training data

Inconsistent:
S(x): \( p=0.5 \)
S(x) \( \land R(x) \): \( p=0.9 \)

Consistent:
S(x): \( w=5 \)
S(x) \( \land R(x) \): \( w=9 \)
MLN’s to Tuple-Independent PDB

Replace probabilities with weights:

R:

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>w</th>
</tr>
</thead>
<tbody>
<tr>
<td>a1</td>
<td>b1</td>
<td>w1</td>
</tr>
<tr>
<td>a2</td>
<td>b1</td>
<td>w2</td>
</tr>
<tr>
<td>a3</td>
<td>b2</td>
<td>w3</td>
</tr>
</tbody>
</table>

S:

<table>
<thead>
<tr>
<th>y</th>
<th>w</th>
</tr>
</thead>
<tbody>
<tr>
<td>b1</td>
<td>v1</td>
</tr>
<tr>
<td>b2</td>
<td>v2</td>
</tr>
<tr>
<td>b3</td>
<td>v3</td>
</tr>
</tbody>
</table>

Soft constraint:

\[ R(x,y) \Rightarrow S(y) \quad w_4 \]

Replace with hard constraint:
MLN’s to Tuple-Independent PDB

Replace probabilities with weights:

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>w</th>
</tr>
</thead>
<tbody>
<tr>
<td>a1</td>
<td>b1</td>
<td>w₁</td>
</tr>
<tr>
<td>a2</td>
<td>b1</td>
<td>w₂</td>
</tr>
<tr>
<td>a3</td>
<td>b2</td>
<td>w₃</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>y</th>
<th>w</th>
</tr>
</thead>
<tbody>
<tr>
<td>b1</td>
<td>v₁</td>
</tr>
<tr>
<td>b2</td>
<td>v₂</td>
</tr>
<tr>
<td>b3</td>
<td>v₃</td>
</tr>
</tbody>
</table>

Soft constraint:

\[ R(x,y) \Rightarrow S(y) \quad w₄ \]

Replace with hard constraint:

\[ \Gamma \equiv \forall x \forall y \quad A(x,y) \iff (R(x,y) \Rightarrow S(y)) \]

New relation A:

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>w</th>
</tr>
</thead>
<tbody>
<tr>
<td>a1</td>
<td>b1</td>
<td>w₄</td>
</tr>
<tr>
<td>a1</td>
<td>b2</td>
<td>w₄</td>
</tr>
<tr>
<td>a1</td>
<td>b3</td>
<td>w₄</td>
</tr>
<tr>
<td>a1</td>
<td>b1</td>
<td>w₄</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
MLN’s to Tuple-Independent PDB

Replace probabilities with weights:

Soft constraint:

Replace with hard constraint:

New relation A:
MLN’s to Tuple-Independent PDB

Replace probabilities with weights:

\[ R(x, y) \implies S(y) \]

Soft constraint:

New relation \( A \):

\[ A(x, y) \iff (R(x, y) \implies S(y)) \]

Weight:

\[ \text{Weight}(a1, b1) = w_1w_3v_1w_4w_4w_4w_4w_4w_4 \]

Theorem:

\[ P_{\text{MLN}}(Q) = P_{\text{D}}(Q | \Gamma) \]

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>y</td>
<td>w</td>
</tr>
<tr>
<td>a1</td>
<td>b1</td>
<td>w_1</td>
</tr>
<tr>
<td>a2</td>
<td>b1</td>
<td>w_2</td>
</tr>
<tr>
<td>a3</td>
<td>b2</td>
<td>w_3</td>
</tr>
</tbody>
</table>

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>y</td>
<td>w</td>
<td></td>
</tr>
<tr>
<td>b1</td>
<td>v_1</td>
<td></td>
</tr>
<tr>
<td>b2</td>
<td>v_2</td>
<td></td>
</tr>
<tr>
<td>b3</td>
<td>v_3</td>
<td></td>
</tr>
</tbody>
</table>

Replace with hard constraint:

\[ \Gamma \equiv \forall x \forall y \]

\[ A(x, y) \iff (R(x, y) \implies S(y)) \]
Improved Translation

Replace $\iff$ with $\implies$

A clause remains a clause!

New weight is $w_4-1$

Probability may be $< 0$ !!! That’s OK

Theorem: $P_{MLN}(Q) = P_D(Q | \Gamma)$
SlimShot = SafePlans + Sample

Accuracy = f(Number of Samples)
Lower is better

SlimShot needs $N \approx 200$ for Accuracy=10%
Runtime

Runtime = f(N), where N=10000

Runtime = f(precision)

Dataset: Smokers
Scalability

Dataset: Smokers

Dataset: Drinkers

SlimShot
Duality

- The dual of a query $Q$ is the formula obtained by the following transformations:
  $\land / \lor \rightarrow \lor / \land \quad \forall / \exists \rightarrow \exists / \forall$

- $Q$ and its dual have the same complexity

Query:
$H_0() = \exists x \exists y (R(x) \land S(x,y) \land T(y))$

Dual query:
$H_0 = \forall x \forall y (R(x) \lor S(x,y) \lor T(y))$
Example: Liftable Clause

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

\[
P(Q) = \prod_{b \in \text{Domain}} P(\exists x \ S(x,b) \Rightarrow R(b))
\]

Indep. \ \forall

\[
P(Q) = \prod_{b \in \text{Domain}} [1 - P(\exists x \ S(x,b)) \times (1 - P(R(b)))]
\]

Indep. or:

\[
P(X \Rightarrow Y) = P(\neg X \lor Y) = P(X) (1 - P(Y))
\]

\[
P(Q) = \prod_{b \in \text{Domain}} [1 - (1 - \prod_{a \in \text{Domain}} (1 - P(S(a,b)))) \times (1 - P(R(b)))]
\]

Indep. \ \exists

Lookup the probabilities in \(D\)

Runtime = \(O(n^2)\).