Agenda: Probabilistic Relational Modeling

• Application
  • Information retrieval (IR)
  • Probabilistic Datalog

• Probabilistic relational logics
  • Overview
  • Semantics
  • Inference problems

• Scalability issues
  • Proposed solutions

Goal: Overview of central ideas

*We would like to thank all our colleagues for making their slides available (see some of the references to papers for respective credits). Slides are almost always modified.
Application

• Probabilistic Datalog for information retrieval [Fuhr 95]:

0.7 term(d1,ir).
0.8 term(d1,db).
0.5 link(d2,d1).

about(D,T) :- term(D,T).
about(D,T) :- link(D,D1), about(D1,T).

• Query/Answer

:- term(X,ir) & term(X,db).

X = 0.56 d1

Application: Probabilistic IR

• Probabilistic Datalog

0.7  term(d1,ir).
0.8  term(d1,db).
0.5  link(d2,d1).

about(D,T):- term(D,T).
about(D,T):- link(D,D1), about(D1,T).

• Query/Answer

q(X):- term(X,ir).
q(X):- term(X,db).

:-q(X)
X = 0.94 d1
Application: Probabilistic IR

• Probabilistic Datalog

0.7 \text{term(d1,ir)}.

0.8 \text{term(d1,db)}.

0.5 \text{link(d2,d1)}.

\text{about(D,T)} :- \text{term(D,T)}.

\text{about(D,T)} :- \text{link(D,D1)}, \text{about(D1,T)}.

• Query/Answer

:- \text{about(X,db)}.

\begin{align*}
X &= 0.8 \text{ d1}; \\
X &= 0.4 \text{ d2}
\end{align*}
Application: Probabilistic IR

• Probabilistic Datalog

0.7 term(d1,ir).
0.8 term(d1,db).
0.5 link(d2,d1).

about(D,T):- term(D,T).
about(D,T):- link(D,D1), about(D1,T).

• Query/Answer

:- about(X,db) & about(X,ir).

X = 0.56 d1;
X = 0.28 d2 # NOT naively 0.14 = 0.8*0.5*0.7*0.5
Solving Inference Problems

• QA requires proper probabilistic reasoning

• Scalability issues
  • Grounding and propositional reasoning?
  • In this tutorial the focus is on lifted reasoning in the sense of [Poole 2003]
    • Lifted exact reasoning
    • Lifted approximations

• Need an overview of the field:
  Consider related approaches first

Application: Probabilistic IR

• Uncertain Datalog rules: Semantics?

0.7 term(d1,ir).
0.8 term(d1,db).
0.5 link(d2,d1).
0.9 about(D,T):- term(D,T).
0.7 about(D,T):- link(D,D1), about(D1,T).
Application: Probabilistic IR

• Uncertain Datalog rules: Semantics?

0.7 \text{term}(d1,ir).
0.8 \text{term}(d1,db).
0.5 \text{link}(d2,d1).
0.9 \text{temp1}.
0.7 \text{temp2}.

about(D,T):= \text{term}(D,T), \text{temp1}.
about(D,T):= \text{link}(D,D1), about(D1,T), \text{temp2}.
Probabilistic Datalog: QA

- Derivation of lineage formula with Boolean variables corresponding to used facts
  

- Probabilistic relational algebra
  

- Ranking / top-k QA
  
Probabilistic Relational Logics: Semantics

- **Distribution semantics** (aka grounding or Herbrand semantics) [Sato 95]
  Completely define discrete joint distribution by "factorization"
  Logical atoms treated as random variables
  - Probabilistic extensions to Datalog [Schmidt et al. 90, Dantsin 91, Ng & Subramanian 93, Poole et al. 93, Fuhr 95, Rölleke & Fuhr 97 and later]
  - Primula [Jaeger 95 and later]
  - BLP, ProbLog [De Raedt, Kersting et al. 07 and later]
  - Probabilistic Relational Models (PRMs) [Poole 03 and later]
  - Markov Logic Networks (MLNs) [Domingos et al. 06]

- **Probabilistic Soft Logic (PSL)** [Kimmig, Bach, Getoor et al. 12]
  Define density function using log-linear model

- **Maximum entropy semantics** [Kern-Isberner, Beierle, Finthammer, Thimm 10, 12]
  Partial specification of discrete joint with “uniform completion”

Sato, T., A statistical learning method for logic programs with distribution semantics,
Inference Problems w/ and w/o Evidence

• **Static case**
  • Projection (margins),
  • Most-probable explanation (MPE)
  • Maximum a posteriori (MAP)
  • Query answering (QA): compute bindings

• **Dynamic case**
  • Filtering (current state)
  • Prediction (future states)
  • Hindsight (previous states)
  • MPE, MAP (temporal sequence)
ProbLog

% Intensional probabilistic facts:
0.6::heads(C) :- coin(C).

% Background information:
coin(c1).
coin(c2).
coin(c3).
coin(c4).

% Rules:
someHeads :- heads(_).

% Queries:
query(someHeads).
0.9744

https://dtai.cs.kuleuven.be/problog/
ProbLog

• **Compute marginal probabilities** of any number of ground atoms in the presence of evidence

• **Learn the parameters** of a ProbLog program from partial interpretations

• **Sample** from a ProbLog program
  • Generate random structures (use case: [Goodman & Tenenbaum 16])

• **Solve decision theoretic problems:**
  • Decision facts and utility statements

**References**


Retrieved 2018-9-23 from https://probmods.org/
Markov Logic Networks (MLNs)

• Weighted formulas for modelling constraints [Richardson & Domingos 06]

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

- \(w = \text{weight}\)
- \(\Gamma(x) = \text{FO formula}\)

**weight** of a world = product of \(\exp(w)\)

- for all MLN rules \((w, \Gamma(x))\) and groundings \(\Gamma(a)\) that hold in that world

Probability of a world = \(\frac{\text{weight}}{Z}\)

- \(Z = \text{sum of weights of all worlds}\) (no longer a simple expression!)

Why exp?

- Log-linear models
- Let $D$ be a set of constants and $\omega \in \{0,1\}^m$ a world with $m$ atoms w.r.t. $D$

$$\text{weight}(\omega) = \prod_{\{(w,\Gamma(x))\in MLN \mid \exists a \in D^n : \omega \models \Gamma(a)\}} \exp(w)$$

$$\ln(\text{weight}(\omega)) = \sum_{\{(w,\Gamma(x))\in MLN \mid \exists a \in D^n : \omega \models \Gamma(a)\}} w$$

- Sum allows for component-wise optimization during weight learning

- $Z = \sum_{\omega \in \{0,1\}^m} \ln(\text{weight}(\omega))$
- $P(\omega) = \frac{\ln(\text{weight}(\omega))}{Z}$
Maximum Entropy Principle

• Given:
  • States $s = s_1, s_2, ..., s_n$
  • Density $p(s) = p_s$

• Maximum Entropy Principle:
  • W/o further information, select $p_s$
    s.t. entropy is maximized
    $\sum_{j=1}^{n} p_s(s_j) \log p_s(s_j) = -p_s \log p_s$
  • w.r.t. constraints (expected values)
    $\sum_{j=1}^{n} p_s(s_j)f_i(s_j) = D_i, \forall i$
Maximum Entropy Principle

- Consider Lagrange functional for determining $p_s$

\[ L = -p_s \log p_s - \sum_i \lambda_i \left( \sum_{j=1}^n p_s(s_j) f_i(s_j) - D_i \right) - \mu \left( \sum_{j=1}^n p_s(s_j) - 1 \right) \]

- Partial derivatives of $L$ w.r.t. $p_s$ → roots:

\[ p_s(s) = \frac{\exp[-\sum_i \lambda_i f_i(s)]}{Z} \]

where $Z$ is for normalization (Boltzmann-Gibbs distribution)

- "Global" modeling: additions/changes to constraints/rules influence the whole joint probability distribution
Maximum-Entropy Semantics for PRMs

- **Probabilistic Conditionals** [Kern-Isberner et al 10, 12]

  
r1 : \( \text{likes}(X, Y) \mid \text{el}(X) \land \text{ke}(Y) \)[0.6]
  
r2 : \( \text{likes}(X, \text{fred}) \mid \text{el}(X) \land \text{ke}(\text{fred}) \)[0.4]
  
r1 : \( \text{likes}(\text{clyde}, \text{fred}) \mid \text{el}(\text{clyde}) \land \text{ke}(\text{fred}) \)[0.7]

  \(el = \text{elephant}, ke = \text{keeper}\)

- **Averaging semantics**
- **Aggregating semantics**
- **Allows for "local modeling" → transfer learning made easier**

---


Factor graphs

• Unifying representation for specifying discrete distributions with a factored representation
  • Potentials (weights) rather than probabilities

• Also used in engineering community for defining densities w.r.t. continuous domains
  [Loeliger et al. 07]

Agenda for the remaining part

Scalability: Proposed solutions

• Limited expressivity
  • Probabilistic databases

• Knowledge Compilation
  • Linear programming
  • Weighted first-order model counting

• Approximation (if time permits)
  • Grounding + belief propagation (TensorLog)

Goal: Give overview of the field
(all parts fit together)
## Probabilistic Databases

### P(Joe) = 1.0

### P(Jim) = 0.4

### Q(z) = Owner(z,x), Location(x,t,’Office’)

### Table 1: Probabilistic Database

<table>
<thead>
<tr>
<th>Name</th>
<th>Object</th>
<th>W_1</th>
<th>W_2</th>
<th>W_3</th>
<th>W_4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owner</td>
<td></td>
<td>0.3</td>
<td>0.4</td>
<td>0.2</td>
<td>0.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joe</td>
<td>Book302</td>
</tr>
<tr>
<td>Joe</td>
<td>Laptop77</td>
</tr>
<tr>
<td>Jim</td>
<td>Laptop77</td>
</tr>
<tr>
<td>Fred</td>
<td>GgleGlass</td>
</tr>
</tbody>
</table>

### Table 2: Probabilistic Database

<table>
<thead>
<tr>
<th>Name</th>
<th>Object</th>
<th>W_1</th>
<th>W_2</th>
<th>W_3</th>
<th>W_4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owner</td>
<td></td>
<td>0.3</td>
<td>0.4</td>
<td>0.2</td>
<td>0.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joe</td>
<td>Book302</td>
</tr>
<tr>
<td>Jim</td>
<td>Laptop77</td>
</tr>
<tr>
<td>Fred</td>
<td>GgleGlass</td>
</tr>
</tbody>
</table>

### Table 3: Probabilistic Database

<table>
<thead>
<tr>
<th>Name</th>
<th>Object</th>
<th>W_1</th>
<th>W_2</th>
<th>W_3</th>
<th>W_4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owner</td>
<td></td>
<td>0.3</td>
<td>0.4</td>
<td>0.2</td>
<td>0.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joe</td>
<td>Book302</td>
</tr>
<tr>
<td>Jim</td>
<td>Laptop77</td>
</tr>
<tr>
<td>Fred</td>
<td>GgleGlass</td>
</tr>
</tbody>
</table>

### Table 4: Probabilistic Database

<table>
<thead>
<tr>
<th>Name</th>
<th>Object</th>
<th>W_1</th>
<th>W_2</th>
<th>W_3</th>
<th>W_4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owner</td>
<td></td>
<td>0.3</td>
<td>0.4</td>
<td>0.2</td>
<td>0.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joe</td>
<td>Book302</td>
</tr>
<tr>
<td>Jim</td>
<td>Laptop77</td>
</tr>
<tr>
<td>Fred</td>
<td>GgleGlass</td>
</tr>
</tbody>
</table>

### Location

<table>
<thead>
<tr>
<th>Name</th>
<th>Owner (x, t)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joe</td>
<td>Book302, Hall</td>
</tr>
<tr>
<td>Joe</td>
<td>Laptop77, Office</td>
</tr>
<tr>
<td>Jim</td>
<td>Laptop77, Office</td>
</tr>
<tr>
<td>Fred</td>
<td>GgleGlass, Office</td>
</tr>
</tbody>
</table>

### Q(z) = Owner(z, x), Location(x, t, ’Office’)
**BID Tables**

<table>
<thead>
<tr>
<th>Object</th>
<th>Time</th>
<th>Loc</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Laptop77</td>
<td>9:07</td>
<td>Rm444</td>
<td>$p_1$</td>
</tr>
<tr>
<td>Laptop77</td>
<td>9:07</td>
<td>Hall</td>
<td>$p_2$</td>
</tr>
<tr>
<td>Book302</td>
<td>9:18</td>
<td>Office</td>
<td>$p_3$</td>
</tr>
<tr>
<td>Book302</td>
<td>9:18</td>
<td>Rm444</td>
<td>$p_4$</td>
</tr>
<tr>
<td>Book302</td>
<td>9:18</td>
<td>Lift</td>
<td>$p_5$</td>
</tr>
</tbody>
</table>

**W** = {$p_1 p_3 \cdot p_4 \cdot p_5$}

**BID Table**

**Disjoint**

**Independent**

**Worlds**
QA: Example

Transformation to SQL is possible

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

\[
P(Q) = 1 - \left\{ 1 - p1 \left[ 1 - (1-q1)(1-q2) \right] \right\} \times \left\{ 1 - p2 \left[ 1 - (1-q3)(1-q4)(1-q5) \right] \right\}
\]

Determine \( P(Q) \) in PTIME w.r.t. size of \( D \)

\[
\begin{array}{c|c|c|c}
R & x & P & S \\
\hline
& a1 & p1 & \text{q1} \\
& a2 & p2 & \text{q2} \\
& a3 & p3 & \text{q3} \\
& a1 & b1 & \text{q1} \\
& a1 & b2 & \text{q2} \\
& a2 & b3 & \text{q3} \\
& a2 & b4 & \text{q4} \\
& a2 & b5 & \text{q5} \\
\end{array}
\]
Problem: Some Queries don't Scale

• Dichotomy P vs. #P [Suciu et al. 11]

• Important research area:
  • Transformation of queries to SQL
  • Lifting of queries [Kazemi et al. 17]

• With probabilistic databases, queries tend to be large and complex
  • Difficult to meet constraints for P-fragment (or to avoid the #P-fragment)


Probabilistic Relational Logic

• First-order logic formulas for expressivity

• Knowledge compilation for scalability
  • Compilation to linear programming
    • Probabilistic Soft Logic [Kimmig, Bach, Getoor et al. 12]
    • Probabilistic Doxastic Temporal Logic [Martiny & Möller 16]
  • Weighted first-order model counting (WFOMC) [Van den Broeck, Taghipour, Meert, Davis, & De Raedt 11]


Probabilistic Soft Logic: Example

• First-order logic weighted rules
  0.3 : friend(B,A) ∧ votesFor(A,P) → votesFor(B,P)
  0.8 : spouse(B,A) ∧ votesFor(A,P) → votesFor(B,P)

• Evidence
  friend(John,Alex) = 1  votesFor(Alex,Romney) = 1
  spouse(John,Mary) = 1  votesFor(Mary,Obama) = 1

• Inference
  votesFor(John, Romney)
  votesFor(John, Obama)

PSL’s Interpretation of Logical Connectives

• Continuous truth assignments
  Łukasiewicz relaxation of AND, OR, NOT
  • $I(\ell_1 \land \ell_2) = \max\{0, I(\ell_1) + I(\ell_2) - 1\}$
  • $I(\ell_1 \lor \ell_2) = \min\{I(\ell_1) + I(\ell_2), 1\}$
  • $I(\neg \ell_1) = 1 - I(\ell_1)$

• Distance to satisfaction $d$
  • Implication: $\ell_1 \rightarrow \ell_2$ is satisfied iff $I(\ell_1) \leq I(\ell_2)$
  • $d = \max\{0, I(\ell_1) - I(\ell_2)\}$
  • Example
    • $I(\ell_1) = 0.3, I(\ell_2) = 0.9 \Rightarrow d = 0$
    • $I(\ell_1) = 0.9, I(\ell_2) = 0.3 \Rightarrow d = 0.6$
PSL Probability Distribution

• Density function

\[ f(I) = \frac{1}{Z} \exp \left[ - \sum_{r \in R} \lambda_r \cdot d_r(I) \right] \]

- A possible continuous truth assignment
- Normalization constant
- For all rules
- Weight of formula \( r \)
- Distance to satisfaction of rule \( r \)
Weighted First-order Model Counting

- **Model** = Satisfying assignment of a propositional formula \( \Delta \)

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

Days = \{Monday, Tuesday\}

**Rain**

<table>
<thead>
<tr>
<th>d</th>
<th>w(R(d))</th>
<th>w(\neg R(d))</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>T</td>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>

**Cloudy**

<table>
<thead>
<tr>
<th>d</th>
<th>w(C(d))</th>
<th>w(\neg C(d))</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>T</td>
<td>6</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rain(M)</th>
<th>Cloudy(M)</th>
<th>Rain(T)</th>
<th>Cloudy(T)</th>
<th>Model?</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>Yes</td>
<td>1 * 3 * 4 * 6 = 72</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>No</td>
<td>0</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>Yes</td>
<td>2 * 3 * 4 * 6 = 144</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>Yes</td>
<td>2 * 5 * 4 * 6 = 240</td>
</tr>
</tbody>
</table>


\#SAT = 9  \quad \text{WFOMC} = 608
From Probabilities to Weights

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>P</th>
<th>w(Friend(x,y))</th>
<th>w(¬Friend(x,y))</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>( p_1 )</td>
<td>( w_1 = p_1 )</td>
<td>( w_1 = 1 - p_1 )</td>
</tr>
<tr>
<td>A</td>
<td>C</td>
<td>( p_2 )</td>
<td>( w_2 = p_2 )</td>
<td>( w_2 = 1 - p_2 )</td>
</tr>
<tr>
<td>B</td>
<td>C</td>
<td>( p_3 )</td>
<td>( w_3 = p_3 )</td>
<td>( w_3 = 1 - p_3 )</td>
</tr>
<tr>
<td>A</td>
<td>A</td>
<td>( 0 )</td>
<td>( w_4 = 0 )</td>
<td>( w_4 = 1 )</td>
</tr>
<tr>
<td>A</td>
<td>C</td>
<td>( 0 )</td>
<td>( w_5 = 0 )</td>
<td>( w_5 = 1 )</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Discussion

• Simple idea: replace $p, 1-p$ with $w, w$
  • Weights, not necessarily probabilities

• Query answering by WFOMC
  • For obtaining probabilities:
    Divide world weight by $Z = \text{sum of all world weights}$
Z Computation

• Formula $\Delta$
  • All MLN constraints are hard: $\Delta = \land_{(\infty, \Gamma(x)) \in MLN} (\forall x \Gamma(x))$
  • If $(\omega_i, \Gamma_i(x))$ is a soft MLN constraint, then:
    • Remove $(\omega_i, \Gamma_i(x))$ from the MLN
    • Add new probabilistic relation $F_i(x)$
    • Add hard constraint $(\infty, \forall x (F_i(x) \iff \Gamma_i(x)))$

• Weight function $\omega(.)$
  • For all constants $A$, relations $F_i$, set $\omega(F_i(A)) = \exp(\omega_i)$, $\omega(\neg F_i(A)) = 1$

• Theorem: $Z = WFOMC(\Delta)$
Example

• Formula $\Delta$

$\Delta = \forall x \ (\text{Smoker}(x) \Rightarrow \text{Person}(x))$
$\quad \land \ \forall x \forall y \ (F(x,y) \Leftrightarrow [\text{Smoker}(x) \land \text{Friend}(x,y) \Rightarrow \text{Smoker}(y)])$

$\infty$ Smoker(x) ⇒ Person(x)

3.75 Smoker(x) ∧ Friend(x,y) ⇒ Smoker(y)

• Weight function $w(.)$

<table>
<thead>
<tr>
<th>$F$</th>
<th>$x$</th>
<th>$y$</th>
<th>$w(F(x,y))$</th>
<th>$w(\neg F(x,y))$</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A</td>
<td>exp(3.75)</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>exp(3.75)</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>C</td>
<td>exp(3.75)</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>A</td>
<td>exp(3.75)</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

Note: if no tables given for Smoker, Person, etc. (i.e., no evidence), then set their $w = w_0 = 1$

$Z = WFOMC(\Delta)$
Knowledge Compilation for Counting

• Main idea: convert $\Delta$ into a different “form” from which one can easily read off the solution count (and many other quantities of interest) [Darwiche & Marquis 2002]

• d-DNNF: deterministic, decomposable negation normal form
  • Think of the formula as a directed acyclic graph (DAG)
  • Negations allowed only at the leaves (NNF)
  • Children of AND node don’t share any variables (different “components”)
  • Children of OR node don’t share any solutions

• Once converted to d-DNNF, can answer many queries in linear time
  • Satisfiability, tautology, logical equivalence, solution counts, ...
  • Any query that a BDD could answer
Compilation to d-DNNF

• “Domain-liftable” FO formula

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

• Probability of a query depends only on the size(s) of the domain(s), a weight function for the first-order predicates, and the weighted model count over the FO d-DNNF.

uncle(X,Y):-child(X,W),brother(W,Y).
uncle(X,Y):-aunt(X,W),husband(W,Y).
status(X,tired):-child(W,X),infant(W).

“Grounding” the rules

DB facts

possible inferences

Explicit grounding not scalable

uncle(X,Y):-child(X,W),brother(W,Y).
uncle(X,Y):-aunt(X,W),husband(W,Y).
status(X,tired):-child(W,X),infant(W).

Example: inferring family relations like “uncle”
• N people
• N^2 possible “uncle” inferences

• N = 2 billion \Rightarrow N^2 = 4 \text{ quintillion}
• N = 1 million \Rightarrow N^2 = 1 \text{ trillion}

A KB with 1M entities is small
Key Question: How to reason?

Example: inferring family relations like “uncle”

- N people
- \( N^2 \) possible “uncle” facts

\[ N = 1 \text{ million} \quad \Rightarrow \quad N^2 = 1 \text{ trillion} \]

\( x \) is the nephew \quad \( x \) is the uncle

\( f_1(x) = Y \quad f_2(x) = Y \)

one-hot vectors

vectors encoding weighted set of DB instances
Query: uncle(liam, Y) ?

Generic case for \( p(c,Y) \):
- initialize the evidence variable \( X \) to a one-hot vector for \( c \)
- wait for BP to converge
- read off the message \( y \) that would be sent from the output variable \( Y \).
- un-normalized prob
- \( y[d] \) is the weighted number of proofs supporting \( p(c,d) \) using this clause

Output msg for brother is sparse matrix multiply: \( v_W M_{\text{brother}} \)
Wrap-up Statistical Relational AI

• Probabilistic relational logics
  • Overview
  • Semantics
  • Inference problems

• Dealing with scalability issues (avoiding grounding)
  • Reduce expressivity (liftable queries)
  • Knowledge compilation (WFOMC)
  • Approximation (BP)

Next: Exact Lifted Inference
Mission and Schedule of the Tutorial*

Providing an overview and a synthesis of StaR AI

- Introduction 10 min
  - StaR AI ✓
- Overview: Probabilistic relational modeling 40 min
  - Semantics (grounded-distributional, maximum entropy) ✓
  - Inference problems and their applications
  - Algorithms and systems
  - Scalability (limited expressivity, knowledge compilation, approximation)
- Scalability by lifting 40+50 min
  - Exact lifted inference 30 min
  - Approximate lifted inference
- Summary 10 min

*Thank you to the SRL/StaRAI crowd for all their exciting contributions! The tutorial is necessarily incomplete. Apologies to anyone whose work is not cited