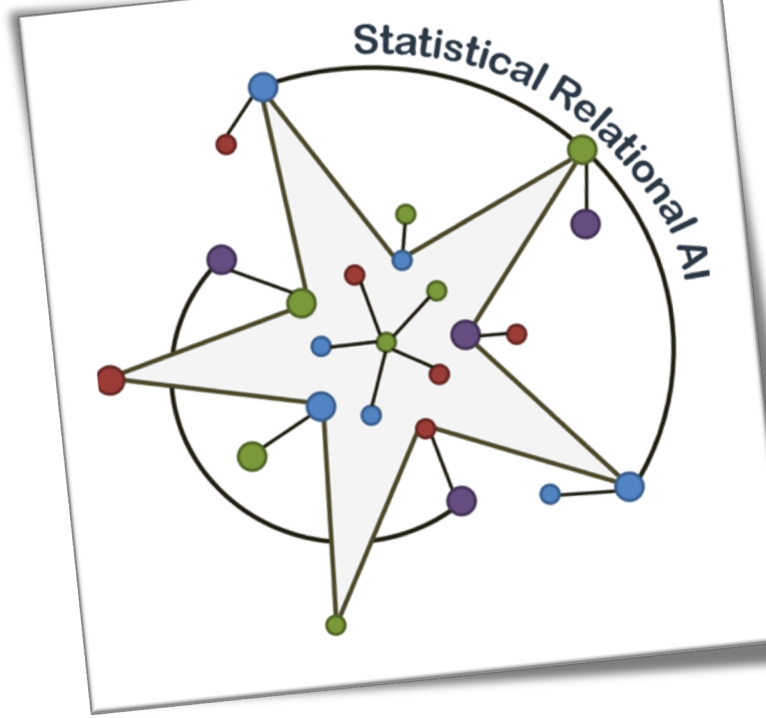
Probabilistic Relational Modeling

Statistical Relational AI

Tutorial at BTW 2019



Tanya Braun, University of Lübeck

UNIVERSITÄT ZU LÜBECK

Thanks to Ralf Möller for making his slides publicly available.

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
```

N. Fuhr, Probabilistic Datalog - A Logic For Powerful Retrieval Methods. Prod. SIGIR, pp. 282-290, 1995

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 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).

X = 0.8 d1;
X = 0.4 d2
```

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

D. Poole. "First-order Probabilistic Inference." In: IJCAI-03 Proceedings of the 18th International Joint Conference on Artificial Intelligence. 2003

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: Semnatics?

```
0.7 term(d1,ir).
0.8 term(d1,db).
0.5 link(d2,d1).
0.9 temp1.
0.7 temp2.
    about(D,T):- term(D,T), temp1.
    about(D,T):- link(D,D1), about(D1,T), temp2.
```

Probabilistic Datalog: QA

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

T. Rölleke; N. Fuhr, Information Retrieval with Probabilistic Datalog. In: Logic and Uncertainty in Information Retrieval: Advanced models for the representation and retrieval of information, 1998.

- Probabilistic relational algebra

N. Fuhr; T. Rölleke, A Probabilistic Relational Algebra for the Integration of Information Retrieval and Database Systems. ACM Transactions on Information Systems 14(1), 1997.

- Ranking / top-k QA

N. Fuhr. 2008. A probability ranking principle for interactive information retrieval. Inf. Retr. 11, 3, 251-265, 2008.

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,
In: International Conference on Logic Programming, pages 715–729, 1995.

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

ProbLog: A probabilistic Prolog and its application in link discovery, L. De Raedt, A. Kimmig, and H. Toivonen, Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07), Hyderabad, India, pages 2462-2467, 2007

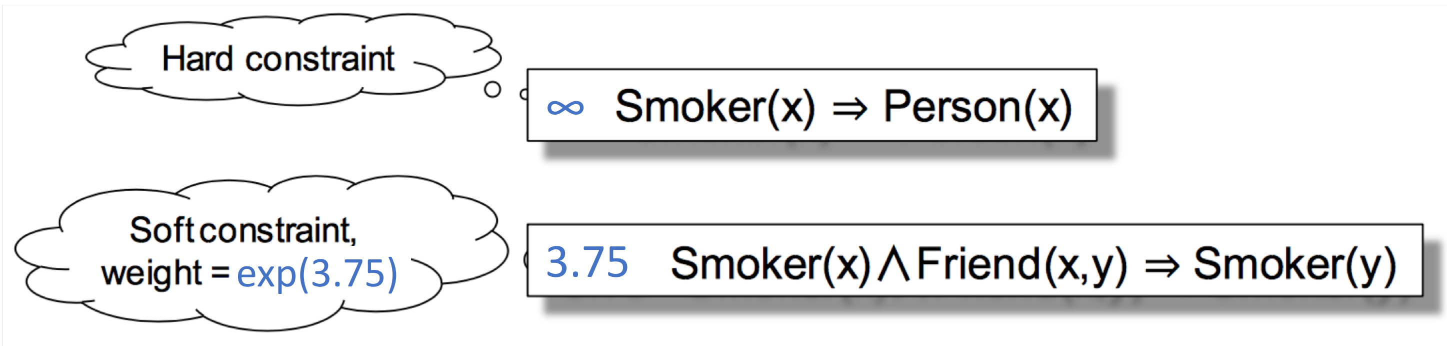
K. Kersting and L. De Raedt, Bayesian logic programming: Theory and Tool. In L. Getoor and B. Taskar, editors, An Introduction to Statistical Relational Learning. MIT Press, 2007

Daan Fierens, Guy Van den Broeck, Joris Renkens, Dimitar Shterionov, Bernd Gutmann, Ingo Thon, Gerda Janssens, and Luc De Raedt. Inference and learning in probabilistic logic programs using weighted Boolean formulas, In: Theory and Practice of Logic Programming, 2015

N. D. Goodman, J. B. Tenenbaum, and The ProbMods Contributors. Probabilistic Models of Cognition (2nd ed.), 2016. 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 $(w, \Gamma(\boldsymbol{x}))$
  - w = weight
  - $\Gamma(\boldsymbol{x})$ = FO formula
- *weight* of a world = product of $\exp(w)$
  - for all MLN rules $(w, \Gamma(\boldsymbol{x}))$ and groundings $\Gamma(\boldsymbol{a})$ that hold in that world
- Probability of a world $= \frac{weight}{Z}$
  - Z = sum of weights of all worlds (no longer a simple expression!)

Richardson, M., & Domingos, P. Markov logic networks. Machine learning, 62(1-2), 107-136. 2006

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

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

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

  - Sum allows for component-wise optimization during weight learning

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

Maximum Entropy Principle

- Given:
  - States $s = s_1, s_2, \dots, s_n$
  - 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

$$L = \underbrace{-p_s \log p_s}_{\text{Entropy}} - \sum_i \underbrace{\lambda_i}_{\text{weighted}} \underbrace{\left(\sum_{j=1}^{n} p_s(s_j) f_i(s_j) - D_i \right)}_{\text{Constraints}} - \underbrace{\mu}_{\text{weighted}} \underbrace{\left(\left(\sum_{j=1}^{n} p_s(s_j) \right) - 1 \right)}_{\text{Regularization}}$$

- Partial derivatives of L w.r.t. $p_s \longrightarrow$ 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 : (likes(X, Y) | el(X) ∧ ke(Y))[0.6]
r2 : (likes(X, fred) | el(X) ∧ ke(fred))[0.4]
r1 : (likes(clyde, fred) | el(clyde) ∧ ke(fred))[0.7]
```

el = elephant, ke = keeper

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

G. Kern-Isberner and M. Thimm. “Novel Semantical Approaches to Relational Probabilistic Conditionals.” In: Proc. KR’10, pp. 382–392, 2010.

G. Kern-Isberner, C. Beierle, M. Finthammer, and M. Thimm. “Comparing and Evaluating Approaches to Probabilistic Reasoning: Theory, Implementation, and Applications.” In: Transactions on Large-Scale Data- and Knowledge-Centered Systems VI. LNCS 7600. Springer , pp. 31–75, 2012.

M. Finthammer, “Concepts and Algorithms for Computing Maximum Entropy Distributions for Knowledge Bases with Relational Probabilistic Conditionals.” IOS Press, 2017.

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]

H.-A. Loeliger, J. Dauwels, J. Hu, S. Korl, L. Ping, and F. R. Kschischang. "The Factor Graph Approach to Model-Based Signal Processing." In: Proc. IEEE 95.6, pp. 1295–1322, 2007.

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')

W_1 — 0.3

Owner

Name	Object
Joe	Book302
Joe	Laptop77
Jim	Laptop77
Fred	GgleGlass

Location

Object	Time	Loc
Laptop77	5:07	Hall
Laptop77	9:05	Office
Book302	8:18	Office

Q=

Joe
Jim

W_2 — 0.4

Owner

Name	Object
Joe	Book302
Jim	Laptop77
Fred	GgleGlass

Location

Object	Time	Loc
Book302	8:18	Office

Q=

Joe

W_3 — 0.2

Owner

Name	Object
Joe	Laptop77

Location

Object	Time	Loc
Laptop77	5:07	Hall
Laptop77	9:05	Office

Q=

Joe

W_4 — 0.1

Owner

Name	Object
Joe	Book302
Jim	Laptop77
Fred	GgleGlass

Location

Object	Time	Loc
Laptop77	5:07	Hall
Laptop77	9:05	Office
Book302	8:18	Office

Q=

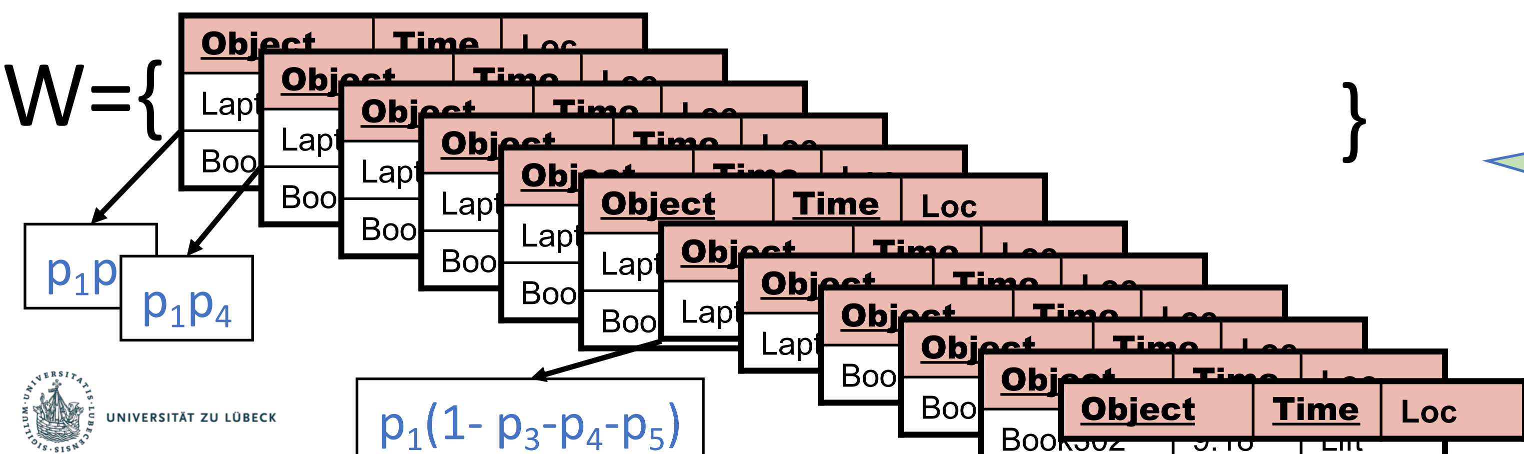
Joe
Jim

BID Tables

BID Table

Object	**Time**	**Loc**	**P**
Laptop77	9:07	Rm444	p_1
Laptop77	9:07	Hall	p_2
Book302	9:18	Office	p_3
Book302	9:18	Rm444	p_4
Book302	9:18	Lift	p_5

disjoint (rows 1–2); disjoint (rows 3–5); independent



QA: Example

Transformation to SQL is possible

```
SELECT DISTINCT 'true'
FROM R, S
WHERE R.x = S.x
```

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

$$P(Q) = 1- \{1- p1*[\ 1-(1-q1)*(1-q2)\ ]\} * \{1- p2*[\ 1-(1-q3)*(1-q4)*(1-q5)]\}$$

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

R

x	P
a1	p1
a2	p2
a3	p3

S

x	y	P
a1	b1	q1
a1	b2	q2
a2	b3	q3
a2	b4	q4
a2	b5	q5

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)

D. Suciu, D. Olteanu, R. Christopher, and C. Koch. Probabilistic Databases. 1st. Morgan & Claypool Publishers, 2011.

S. M. Kazemi, A. Kimmig, G. Van den Broeck, and D. Poole. "Domain Recursion for Lifted Inference with Existential Quantifiers." In: Seventh International Workshop on Statistical Relational AI (StaRAI). Aug. 2017

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]

Kimmig, A., Bach, S. H., Broecheler, M., Huang, B. & Getoor, L. A Short Introduction to Probabilistic Soft Logic. NIPS Workshop on Probabilistic Programming: Foundations and Applications, 2012.

Karsten Martiny, Ralf Möller: PDT Logic: A Probabilistic Doxastic Temporal Logic for Reasoning about Beliefs in Multi-agent Systems In: J. Artif. Intell. Res. (JAIR), Vol.57, p.39-112, 2016.

Van den Broeck, G., Taghipour, N., Meert, W., Davis, J., & De Raedt, L., Lifted probabilistic inference by first-order knowledge compilation. In Proc.IJCAI-11, pp. 2178-2185, 2011.

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)
```

Kimmig, A., Bach, S. H., Broecheler, M., Huang, B. & Getoor, L. A Short Introduction to Probabilistic Soft Logic. NIPS Workshop on Probabilistic Programming: Foundations and Applications, 2012.

PSL’s Interpretation of Logical Connectives

- Continuous truth assignments
  Łukasiewicz relaxation of AND, OR, NOT
  - $I(\ell_1 \wedge \ell_2) = \max\{0, I(\ell_1) + I(\ell_2) - 1\}$
  - $I(\ell_1 \vee \ell_2) = \min\{I(\ell_1) + I(\ell_2), 1\}$
  - $I(\neg \ell_1) = 1 - I(\ell_1)$
- Distance to satisfaction d
  - Implication: $\ell_1 \rightarrow \ell_2$ is satisfied iff $\mathrm{I}(\ell_1) \leq I(\ell_2)$
  - $d = \max\{0, I(\ell_1) - I(\ell_2)\}$
  - Example
    - $\mathrm{I}(\ell_1) = 0.3, \mathrm{I}(\ell_2) = 0.9 \Rightarrow d = 0$
    - $\mathrm{I}(\ell_1) = 0.9, \mathrm{I}(\ell_2) = 0.3 \Rightarrow d = 0.6$

PSL Probability Distribution

- Density function

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

- $f(I)$: a possible continuous truth assignment
- $\frac{1}{Z}$: Normalization constant
- $\sum_{r \in R}$: For all rules
- λ_r: Weight of formula *r*
- $d_r(I)$: Distance to satisfaction of rule r

Weighted First-order Model Counting

- Model = Satisfying assignment of a propositional formula Δ

Δ = ∀d (Rain(d) ⇒ Cloudy(d))

Days = {Monday, **Tuesday**}

Rain

d	w(R(d))	w(¬R(d))
M	1	2
T	4	1

Cloudy

d	w(C(d))	w(¬C(d))
M	3	5
T	6	2

Rain(M)	Cloudy(M)	Rain(T)	Cloudy(T)	Model?	Weight
T	T	T	T	Yes	1 * 3 * 4 * 6 = 72
T	F	T	T	No	0
F	T	T	T	Yes	2 * 3 * 4 * 6 = 144
F	F	T	T	Yes	2 * 5 * 4 * 6 = 240
T	T	T	F	No	0
T	F	T	F	No	0
F	T	T	F	No	0
F	F	T	F	No	0
T	T	F	T	Yes	1 * 3 * 1 * 6 = 18
T	F	F	T	No	0
F	T	F	T	Yes	2 * 3 * 1 * 6 = 36
F	F	F	T	Yes	2 * 5 * 1 * 6 = 60
T	T	F	F	Yes	1 * 3 * 1 * 2 = 6
T	F	F	F	No	0
F	T	F	F	Yes	2 * 3 * 1 * 2 = 12
F	F	F	F	Yes	2 * 5 * 1 * 2 = 20

+ **#SAT = 9**   + **WFOMC = 608**

Gogate, V., & Domingos, P., Probabilistic Theorem Proving. Proc. UAI, 2012.

Van den Broeck, G., Taghipour, N., Meert, W., Davis, J., & De Raedt, L., Lifted probabilistic inference by first-order knowledge compilation. In Proc.IJCAI-11, pp. 2178-2185, 2011.

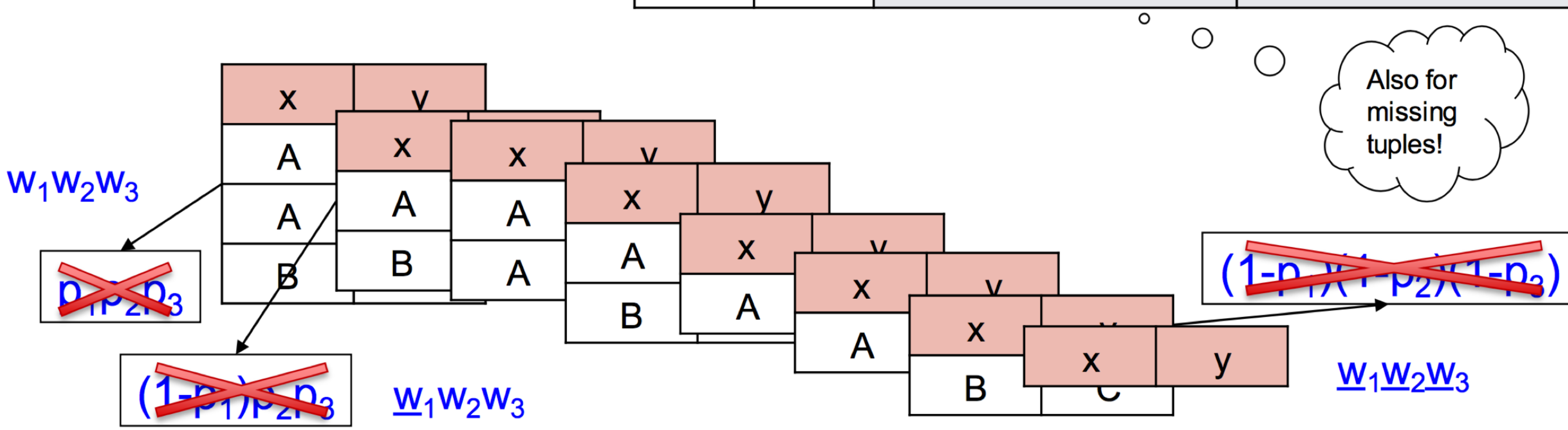
From Probabilities to Weights

Friend

x	y	P
A	B	~~p_1~~
A	C	~~p_2~~
B	C	~~p_3~~

→

x	y	w(Friend(x,y))	w(¬Friend(x,y))
A	B	$w_1 = p_1$	$\underline{w}_1 = 1\text{-}p_1$
A	C	$w_2 = p_2$	$\underline{w}_2 = 1\text{-}p_2$
B	C	$w_3 = p_3$	$\underline{w}_3 = 1\text{-}p_3$
A	A	$w_4 = 0$	$\underline{w}_4 = 1$
A	C	$w_5 = 0$	$\underline{w}_5 = 1$
	…	…	



Discussion

- Simple idea: replace p, 1-p with w, $\underline{w}$
  - Weights, not necessarily probabilities
- Query answering by WFOMC
  - For obtaining probabilities:
    Divide world weight by Z = sum of all world weights

Z Computation

- Formula Δ
  - All MLN constraints are hard: $\Delta = \bigwedge_{(\infty,\Gamma(\boldsymbol{x}))\in MLN}(\forall \boldsymbol{x}\ \Gamma(\boldsymbol{x}))$
  - If $\left(w_i, \Gamma_\mathrm{i}(\boldsymbol{x})\right)$ is a soft MLN constraint, then:
    - Remove $\left(w_i, \Gamma_\mathrm{i}(\boldsymbol{x})\right)$ from the MLN
    - Add new probabilistic relation $\mathrm{F_i}(\boldsymbol{x})$
    - Add hard constraint $\left(\infty, \forall \boldsymbol{x}\ \left(\mathrm{F_i}(\boldsymbol{x}) \Leftrightarrow \Gamma_\mathrm{i}(\boldsymbol{x})\right)\right)$
- Weight function $w(.)$
  - For all constants $\boldsymbol{A}$, relations F_i,
    set $w\left(F_i(\boldsymbol{A})\right) = \exp(w_i), w\left(\neg F_i(\boldsymbol{A})\right) = 1$
- **Theorem:** $Z = WFOMC(\Delta)$

Van den Broeck, G., Meert, W., & Darwiche, A.,. Skolemization for weighted first-order model counting. In Proc. KR-13, **2013**.

Jha, A., & Suciu, D.,.Probabilistic databases with MarkoViews. Proceedings of the VLDB Endowment, 5(11), 1160-1171, **2012**.

Example

- Formula Δ

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

Δ = ∀x (Smoker(x) ⇒ Person(x))
∧ ∀x∀y (F(x,y) ⇔ [Smoker(x) ∧ Friend(x,y) ⇒ Smoker(y)])

- Weight function $w(.)$

F

x	y	w(F(x,y))	w(¬F(x,y))
A	A	exp(3.75)	1
A	B	exp(3.75)	1
A	C	exp(3.75)	1
B	A	exp(3.75)	1
	. . .	. . .	

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

$$Z = WFOMC(\Delta)$$

Knowledge Compilation for Counting

- Main idea: convert Δ 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

can multiply the counts

can add the counts

- 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

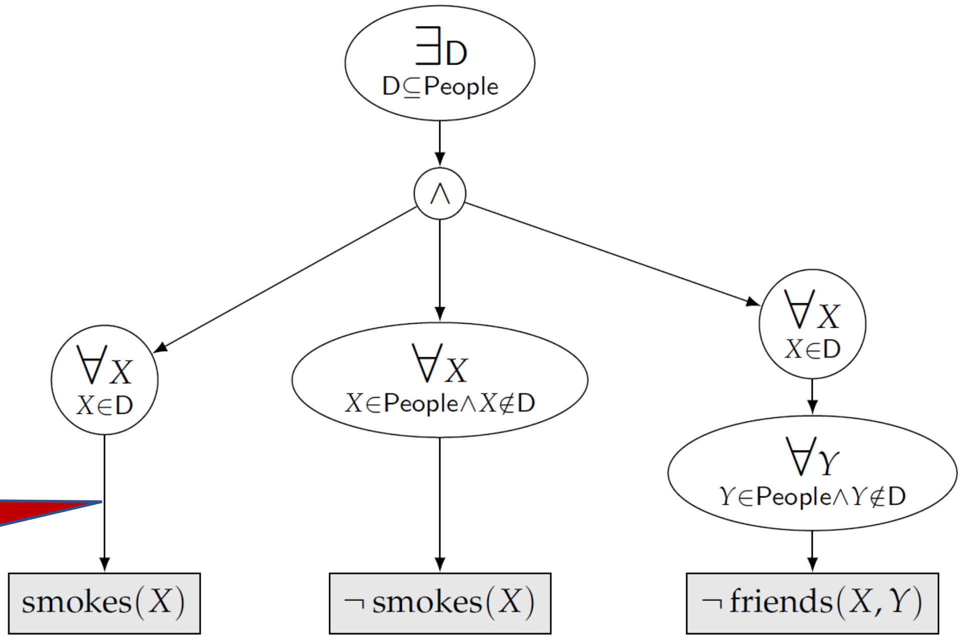
Adnan Darwiche, Pierre Marquis, A Knowledge Compilation Map, In: JAIR 17 (2002) 229-264, **2002**.

Compilation to d-DNNF

- "Domain-liftable" FO formula

$$\forall X, Y \in People,$$
$$smokes(X) \wedge friends(X,Y) \Rightarrow smokes(Y)$$

- 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.

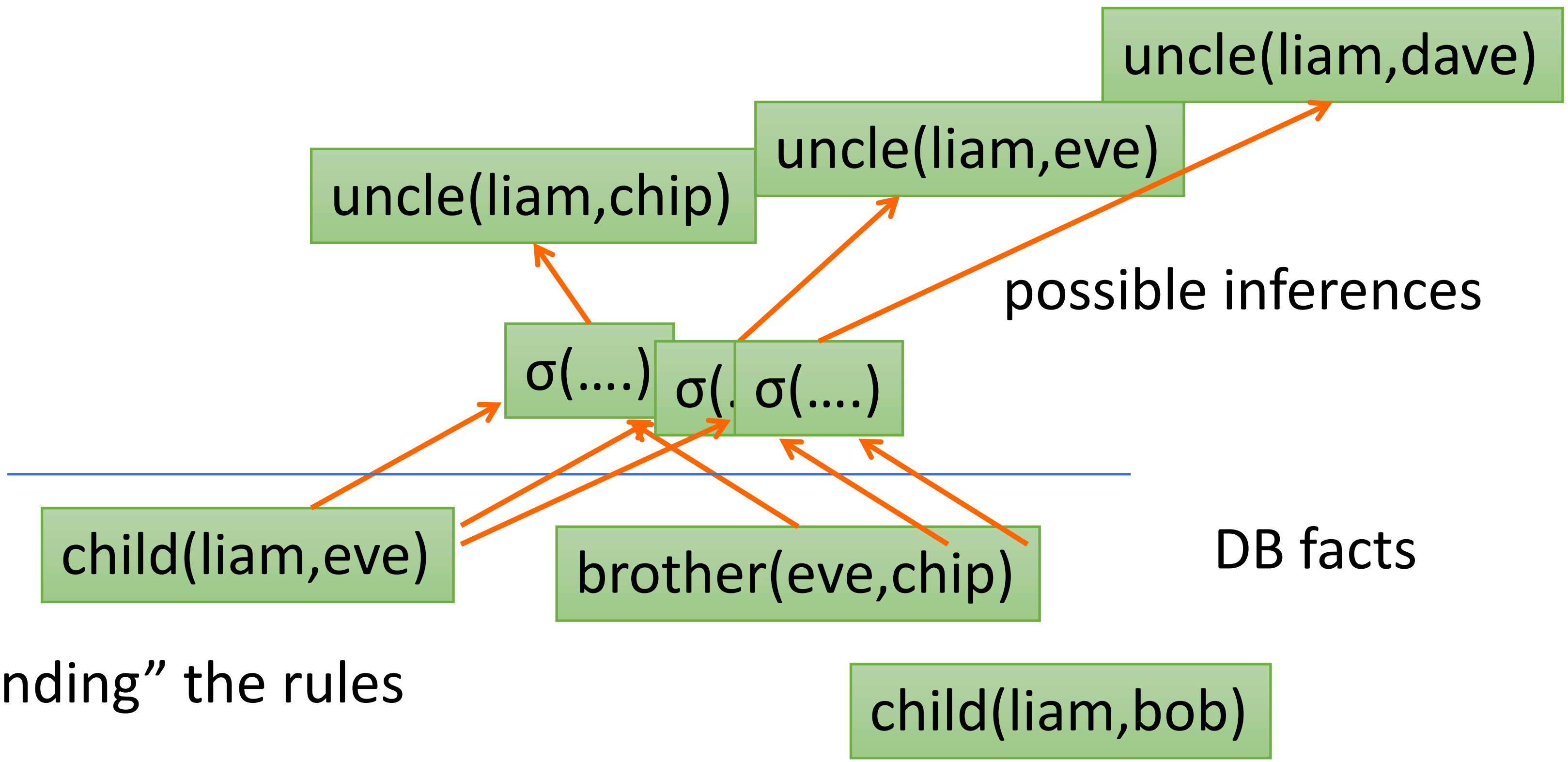


d-DNNF form of Δ can grow large

Guy Van den Broeck, Nima Taghipour, Wannes Meert, Jesse Davis, Luc De Raedt: Lifted Probabilistic Inference by First-Order Knowledge Compilation. In: Proc. IJCAI 2011, pp. 2178-2185, 2011.

TensorLog [Cohen & Yang 17]

```
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

William W. Cohen and Fan Yang, TensorLog: Deep Learning Meets Probabilistic Databases in arxiv.org 1707.05390, **2017**

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 → N^2 = 4 quintillion
- N = 1 million → N^2 = 1 trillion

A KB with 1M entities is *small*

Key Question: How to reason?

```
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" facts
- ~~N = 1 million → N^2 = 1 trillion~~

x is the nephew

$f_1(x) = Y$

x is the uncle

$f_2(\mathbf{x}) = \mathbf{Y}$

one-hot vectors

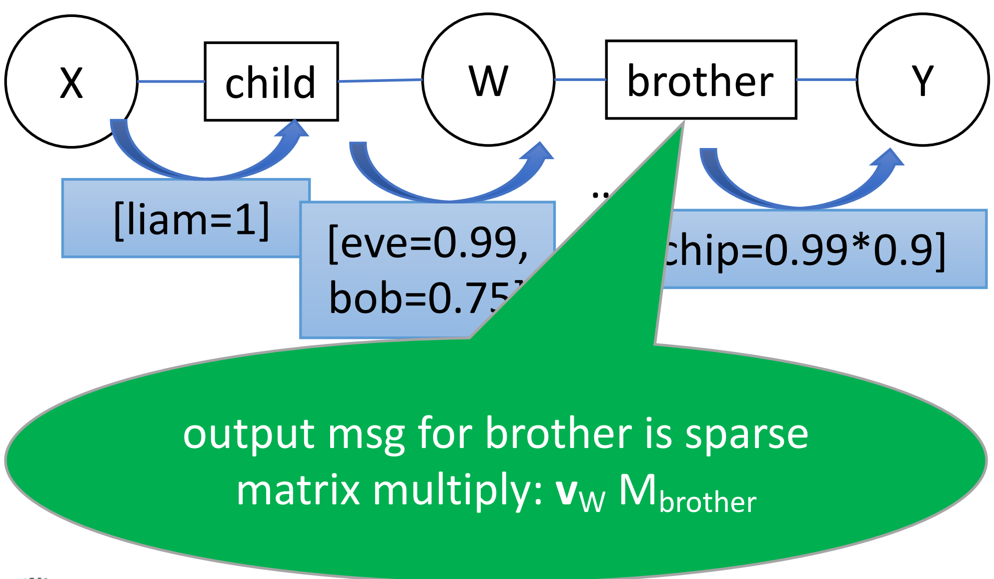
vectors encoding weighted set of DB instances

TensorLog: Approximation by Belief Propagation

```
child(liam,eve),0.99     infant(liam),0.7
child(dave,eve),0.99     infant(dave),0.1
child(liam,bob),0.75     aunt(joe,eve),0.9
husband(eve,bob),0.9     brother(eve,chip),0.9
```

Query: uncle(liam, Y) ?

uncle(X,Y):-child(X,W),brother(W,Y)



General 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

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
  - Exact lifted inference 40+50 min
  - Approximate lifted inference 30 min
- 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