Outline

- Part 1: Motivation
- Part 2: Probabilistic Databases
- Part 3: Weighted Model Counting
- Part 4: Lifted Inference for WFOMC
- Part 5: Completeness of Lifted Inference
- Part 6: Query Compilation
- Part 7: Symmetric Lifted Inference Complexity
- Part 8: Open-World Probabilistic Databases
- Part 9: Discussion & Conclusions

WMC Probabilistic Inference

- Model = solution to a propositional logic formula Δ
- Model counting = #SAT

Δ = (Rain ⇒ Cloudy)

Rain	Cloudy	Model?
T	T	Yes
T	F	No
F	T	Yes
F	F	Yes

+

#SAT = 3

WMC Probabilistic Inference

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

Δ = (Rain ⇒ Cloudy)

Rain

w(R)	w(¬R)
1	2

Cloudy

w(C)	w(¬C)
3	5

Rain	Cloudy	Model?	Weight
T	T	Yes	1 * 3 = 3
T	F	No	0
F	T	Yes	2 * 3 = 6
F	F	Yes	2 * 5 = 10

+

#SAT = 3

WMC Probabilistic Inference

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

Δ = (Rain ⇒ Cloudy)

Rain

w(R)	w(¬R)
1	2

Cloudy

w(C)	w(¬C)
3	5

Rain	Cloudy	Model?	Weight
T	T	Yes	1 * 3 = 3
T	F	No	0
F	T	Yes	2 * 3 = 6
F	F	Yes	2 * 5 = 10

#SAT = 3

WMC = 19

Weighted Model Counting

- Assembly language for non-lifted inference
- Reductions to WMC for inference in
  - Bayesian networks [Chavira’05, Sang’05 , Chavira’08]
  - Factor graphs [Choi’13]
  - Relational Bayesian networks [Chavira’06]
  - Probabilistic logic programs [Fierens’11, Fierens’15]
  - Probabilistic databases [Olteanu’08, Jha’11]
- State-of-the-art exact solvers
  - Knowledge compilation (WMC → d-DNNF → AC)

    Winner of the UAI’08 exact inference competition!
  - DPLL counters

Weighted First-Order Model Counting

Model = solution to first-order logic formula Δ

Δ = ∀d (Rain(d)
⇒ Cloudy(d))

Days = {Monday}

[VdB'11, Gogate'11]

Weighted First-Order Model Counting

Model = solution to first-order logic formula Δ

Δ = ∀d (Rain(d) ⇒ Cloudy(d))

Days = {Monday}

Rain(M)	Cloudy(M)	Model?
T	T	Yes
T	F	No
F	T	Yes
F	F	Yes

+

#SAT = 3

[VdB'11, Gogate'11]

Weighted First-Order Model Counting

Model = solution to first-order logic formula Δ

Δ = ∀d (Rain(d) ⇒ Cloudy(d))

Days = {Monday
Tuesday}

Rain(M)	Cloudy(M)	Rain(T)	Cloudy(T)	Model?
T	T	T	T	Yes
T	F	T	T	No
F	T	T	T	Yes
F	F	T	T	Yes
T	T	T	F	No
T	F	T	F	No
F	T	T	F	No
F	F	T	F	No
T	T	F	T	Yes
T	F	F	T	No
F	T	F	T	Yes
F	F	F	T	Yes
T	T	F	F	Yes
T	F	F	F	No
F	T	F	F	Yes
F	F	F	F	Yes

[VdB'11, Gogate'11]

Weighted First-Order Model Counting

Model = solution to first-order logic formula Δ

Δ = ∀d (Rain(d) ⇒ Cloudy(d))

Days = {Monday **Tuesday**}

Rain(M)	Cloudy(M)	Rain(T)	Cloudy(T)	Model?
T	T	T	T	Yes
T	F	T	T	No
F	T	T	T	Yes
F	F	T	T	Yes
T	T	T	F	No
T	F	T	F	No
F	T	T	F	No
F	F	T	F	No
T	T	F	T	Yes
T	F	F	T	No
F	T	F	T	Yes
F	F	F	T	Yes
T	T	F	F	Yes
T	F	F	F	No
F	T	F	F	Yes
F	F	F	F	Yes

+ ———

#SAT = 9

[VdB'11, Gogate'11]

Weighted First-Order Model Counting

Model = solution to first-order logic 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?
T	T	T	T	Yes
T	F	T	T	No
F	T	T	T	Yes
F	F	T	T	Yes
T	T	T	F	No
T	F	T	F	No
F	T	T	F	No
F	F	T	F	No
T	T	F	T	Yes
T	F	F	T	No
F	T	F	T	Yes
F	F	F	T	Yes
T	T	F	F	Yes
T	F	F	F	No
F	T	F	F	Yes
F	F	F	F	Yes

+ ———

#SAT = 9

[VdB'11, Gogate'11]

Weighted First-Order Model Counting

Model = solution to first-order logic 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

[VdB'11, Gogate'11]

Weighted First-Order Model Counting

Model = solution to first-order logic 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

+ ———— + ————

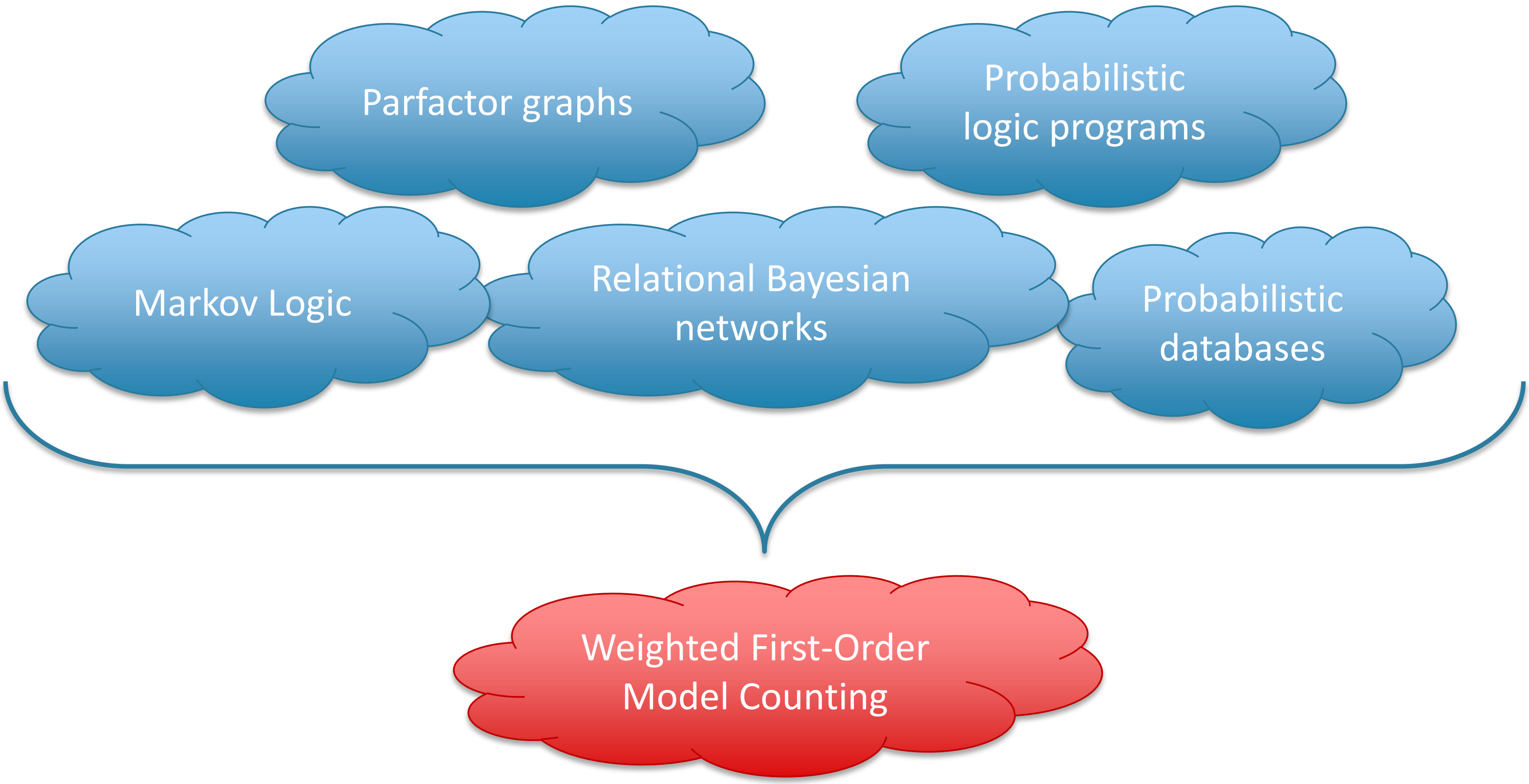
#SAT = 9 **WFOMC = 608**

[VdB'11, Gogate'11]

WFOMC Probabilistic Inference

- Assembly language for lifted inference
- Reduction to WFOMC for lifted inference in
  - Markov logic networks [VdB’11,Gogate’11]
  - Parfactor graphs [VdB’13]
  - Probabilistic logic programs [VdB’14]
  - Probabilistic databases [Gribkoff’14]

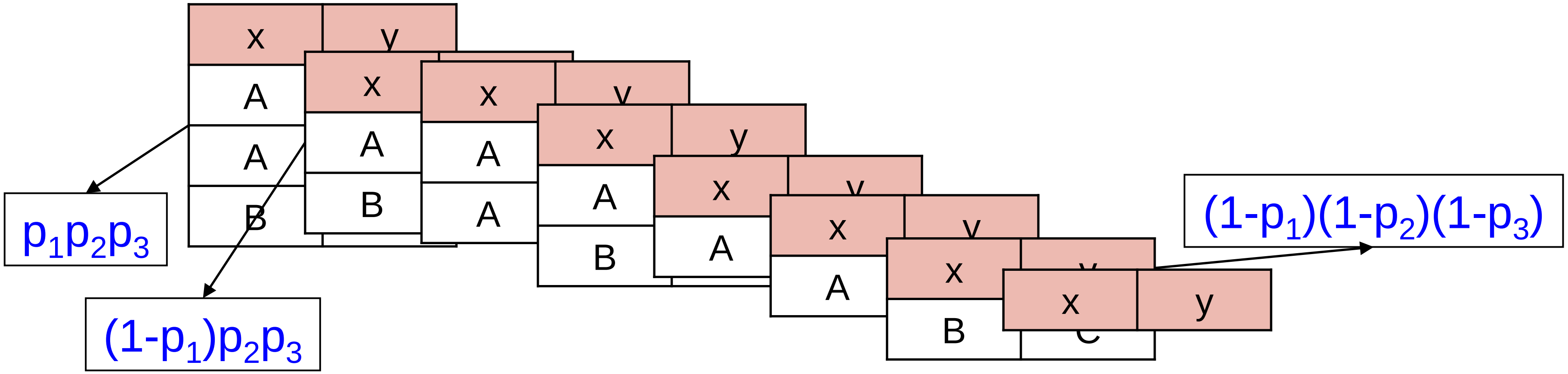
Assembly language for high-level probabilistic reasoning



From Probabilities to Weights

Friend

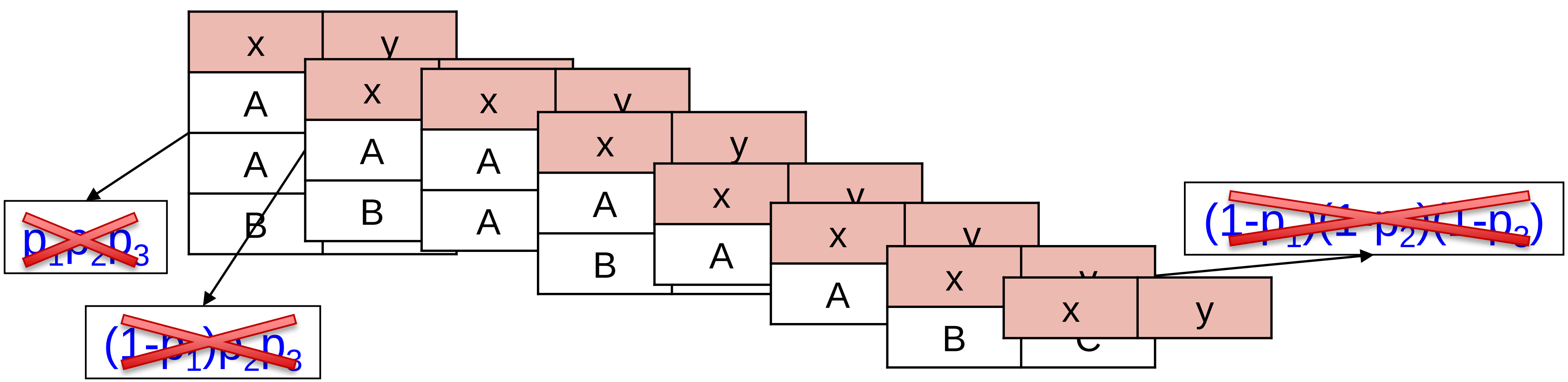
x	y	P
A	B	p_1
A	C	p_2
B	C	p_3



From Probabilities to Weights

Friend

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



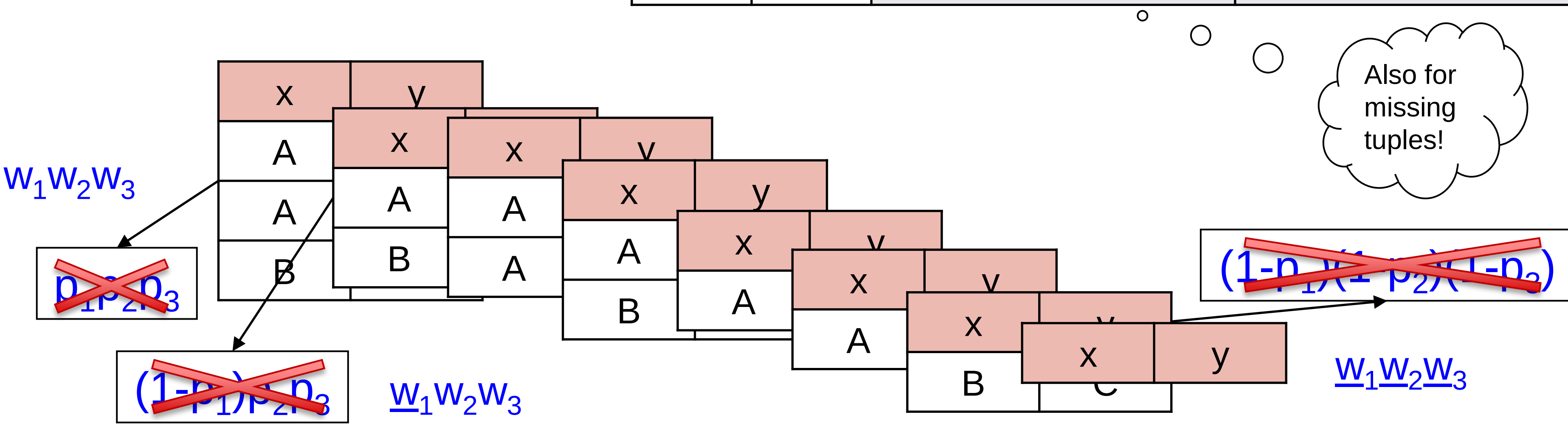
From Probabilities to Weights

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	$w_1 = p_1$	$\underline{w}_1 = 1-p_1$
A	C	$w_2 = p_2$	$\underline{w}_2 = 1-p_2$
B	C	$w_3 = p_3$	$\underline{w}_3 = 1-p_3$
A	A	$w_4 = 0$	$\underline{w}_4 = 1$
A	C	$w_5 = 0$	$\underline{w}_5 = 1$
	…	…	



Discussion

- Simple idea: replace p, $1\text{-}p$ by w, $\underline{w}$
- Query computation becomes WFOMC

- To obtain a probability space, divide the weight of each world by Z = sum of weights of all worlds:

  $Z = (w_1+\underline{w}_1)\ (w_2+\underline{w}_2)\ (w_3+\underline{w}_3)\ \ldots$

- Why weights instead of probabilities?
  They can describe complex correlations (next)

Markov Logic

Capture knowledge through soft constraints (a.k.a. “features”):

Hard constraint

∞ Smoker(x) $\Rightarrow$ Person(x)

Soft constraint,
weight = exp(3.73)

3.75 Smoker(x) $\wedge$ Friend(x,y) $\Rightarrow$ Smoker(y)

[Richardson’06]

Markov Logic

Capture knowledge through soft constraints (a.k.a. “features”):

Hard constraint

∞ Smoker(x) $\Rightarrow$ Person(x)

Soft constraint,
weight = exp(3.73)

3.75 Smoker(x) $\wedge$ Friend(x,y) $\Rightarrow$ Smoker(y)

An MLN is a set of constraints (w, $\Gamma(\mathbf{x})$), where w=weight, $\Gamma(\mathbf{x})$=FO formula

[Richardson’06]

Markov Logic

Capture knowledge through soft constraints (a.k.a. “features”):

Hard constraint

∞ Smoker(x) $\Rightarrow$ Person(x)

Soft constraint,
weight = exp(3.73)

3.75 Smoker(x) $\wedge$ Friend(x,y) $\Rightarrow$ Smoker(y)

An MLN is a set of constraints (w, $\Gamma(\mathbf{x})$), where w=weight, $\Gamma(\mathbf{x})$=FO formula

Weight of a world = product of exp(w), for all MLN rules (w, $\Gamma(\mathbf{x})$)
and grounding $\Gamma(\mathbf{a})$ that hold in that world

[Richardson’06]

Markov Logic

Capture knowledge through soft constraints (a.k.a. “features”):

Hard constraint

∞ Smoker(x) $\Rightarrow$ Person(x)

Soft constraint,
weight = exp(3.73)

3.75   Smoker(x) $\wedge$ Friend(x,y) $\Rightarrow$ Smoker(y)

An MLN is a set of constraints (w, $\Gamma(\mathbf{x})$), where w=weight, $\Gamma(\mathbf{x})$=FO formula

Weight of a world = product of exp(w), for all MLN rules (w, $\Gamma(\mathbf{x})$)
and grounding $\Gamma(\mathbf{a})$ that hold in that world

Probability of a world = Weight / Z
Z = sum of weights of all worlds (no longer a simple expression!)

[Richardson’06]

Discussion

- Probabilistic databases = independence
MLN = complex correlations

- To translate weights to probabilities we need to divide by Z, which often is difficult to compute

- However, we can reduce the Z-computation problem to WFOMC (next)

Z → WFOMC(Δ)

1. **Formula Δ**

2. **Weight function w(.)**

Z → WFOMC(Δ)

1. **Formula Δ**

If all MLN constraints are hard: $\Delta = \bigwedge_{(\infty, \Gamma(\mathbf{x})) \in MLN} (\forall \mathbf{x}\ \Gamma(\mathbf{x}))$

2. **Weight function w(.)**

$Z \rightarrow \text{WFOMC}(\Delta)$

1. Formula Δ

If all MLN constraints are hard: $\Delta = \bigwedge_{(\infty, \Gamma(\mathbf{x})) \in MLN} (\forall \mathbf{x}\ \Gamma(\mathbf{x}))$

If $(w_i, \Gamma_i(\mathbf{x}))$ is a soft MLN constraint, then:

a) Remove $(w_i, \Gamma_i(\mathbf{x}))$ from the MLN

b) Add new probabilistic relation $F_i(\mathbf{x})$

c) Add hard constraint $(\infty, \forall \mathbf{x}\ (F_i(\mathbf{x}) \Leftrightarrow \Gamma_i(\mathbf{x})))$

2. Weight function w(.)

$Z \rightarrow \text{WFOMC}(\Delta)$

1. **Formula Δ**

If all MLN constraints are hard: $\Delta = \bigwedge_{(\infty, \Gamma(\mathbf{x})) \in MLN} (\forall \mathbf{x}\ \Gamma(\mathbf{x}))$

If $(w_i, \Gamma_i(\mathbf{x}))$ is a soft MLN constraint, then:

a) Remove $(w_i, \Gamma_i(\mathbf{x}))$ from the MLN
b) Add new probabilistic relation $F_i(\mathbf{x})$
c) Add hard constraint $(\infty, \forall \mathbf{x}\ (F_i(\mathbf{x}) \Leftrightarrow \Gamma_i(\mathbf{x})))$

2. **Weight function w(.)**

For all constants **A**, relations F_i,
set $w(F_i(\mathbf{A})) = \exp(w_i)$, $w(\neg F_i(\mathbf{A})) = 1$

Better rewritings in
[Jha'12],[V.d.Broeck'14]

$Z \rightarrow \mathrm{WFOMC}(\Delta)$

1. Formula Δ

If all MLN constraints are hard: $\Delta = \bigwedge_{(\infty, \Gamma(\mathbf{x})) \in MLN} (\forall \mathbf{x}\ \Gamma(\mathbf{x}))$

If $(w_i, \Gamma_i(\mathbf{x}))$ is a soft MLN constraint, then:

a) Remove $(w_i, \Gamma_i(\mathbf{x}))$ from the MLN
b) Add new probabilistic relation $F_i(\mathbf{x})$
c) Add hard constraint $(\infty, \forall \mathbf{x}\ (F_i(\mathbf{x}) \Leftrightarrow \Gamma_i(\mathbf{x})))$

2. Weight function w(.)

For all constants **A**, relations F_i,
set $w(F_i(\mathbf{A})) = \exp(w_i)$, $w(\neg F_i(\mathbf{A})) = 1$

Theorem: $Z = \mathrm{WFOMC}(\Delta)$

Better rewritings in [Jha'12],[V.d.Broeck'14]

Example

1. <u>**Formula Δ**</u>

2. <u>**Weight function w(.)**</u>

Example

1. **Formula Δ**

∞	Smoker(x) ⇒ Person(x)

2. **Weight function w(.)**

Example

∞ Smoker(x) ⇒ Person(x)

1. **Formula Δ**

Δ = ∀x (Smoker(x) ⇒ Person(x))

2. **Weight function w(.)**

Example

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

1. **Formula Δ**

Δ = ∀x (Smoker(x) ⇒ Person(x))

2. **Weight function w(.)**

Example

1. **Formula Δ**

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

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

2. **Weight function w(.)**

Example

1. 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)])

2. 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

Example

1. 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)])

2. 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(Δ)

Lessons

- Weighed Model Counting:
  - Unified framework for probabilistic inference tasks
  - Independent variables
- Weighed FO Model Counting:
  - Formula described by a concise FO sentence
  - Still independent variables
- MLNs:
  - Weighted formulas
  - Correlations!
  - Can be converted to WFOMC

Lessons

- Weighed Model Counting:
  - Unified framework for probabilistic inference tasks
  - Independent variables
- Weighed FO Model Counting:
  - Formula described by a concise FO sentence
  - Still independent variables
- MLNs:
  - Weighted formulas
  - Correlations!
  - Can be converted to WFOMC

Tuple-independence is not a severe representational restriction!
It is a convenience for building inference algorithms.

Symmetric vs. Asymmetric

Symmetric WFOMC:

- In every relation R, all tuples have same weight
- Example: converting MLN "without evidence" into WFOMC leads to a symmetric weight function

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
	. . .	. . .	

Asymmetric WFOMC:

- Each relation R is given explicitly
- Example: Probabilistic Databases
- Example: MLN's plus evidence

[Gribkoff'14]

Comparison

	MLNs	Prob. DBs
Random variable is a	Ground atom	DB Tuple
Weights w associated with	Formulas	DB Tuples
Typical query Q is a	Single atom	FO formula/SQL
Data is encoded into	Evidence (Query)	Distribution
Correlations induced by	Model formulas	Query
Model generalizes across domains?	Yes	No
Query generalizes across domains?	No	Yes
Sum of weights of worlds is 1 (normalized)?	No	Yes