Inference for Probabilistic Logic Programming with Continuous Distributions

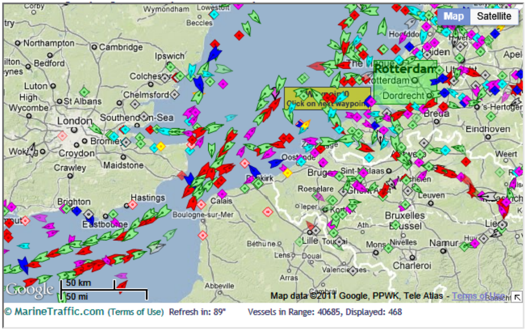
Arjen Hommersom
Open University of the Netherlands
arjen.hommersom@ou.nl
based on joint work with Steffen Michels and Peter Lucas

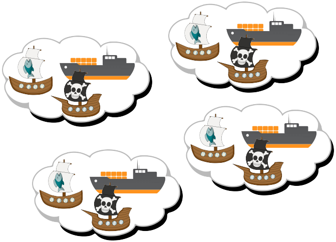
Open Universiteit
www.ou.nl

Motivation: domains with numbers

- Uncertainty over continuous random variables ubiquitous
- E.g. in medicine: the distribution of laboratory test values
  - But also in finance, biology, etc.
- Discretization not always possible during modelling

Relational domains with numbers





[Michels, S., Velikova, M., Hommersom, A., & Lucas, P. J.F. A decision support model for uncertainty reasoning in safety and security tasks. In SMC2013.]

Outline

- PLP with linear constraints
- Distributional clauses
  - Sampling
- Iterative hybrid probabilistic model counting
  - Inference using weighted model counting
  - Generalised weighted model counting
  - Iterative refinement of discretisations
  - Experiments
- Conclusions and outlook

Representation in PRISM

```
widget(X) :- msw(m, M),
             msw(st(M), Z),
             msw(pt, Y),
             X = Y + Z.

% Ranges of RVs
values(m, [a,b]).
values(st(M), real).
values(pt, real).
% PDFs and PMFs:
:- set_sw(m, [0.3, 0.7]),
   set_sw(st(a), norm(2.0, 1.0)),
   set_sw(st(b), norm(3.0, 1.0)),
   set_sw(pt, norm(0.5, 0.1)).
```

(a) Mixture model program

$$G_1 : widget(X)$$
$$\downarrow$$
$$G_2 : msw(m, M), msw(st(M), Z), msw(pt, Y), X = Y + Z.$$
$$\downarrow$$
$$G_3 : msw(st(M), Z), msw(pt, Y), X = Y + Z.$$
$$\downarrow$$
$$G_4 : msw(pt, Y), X = Y + Z.$$
$$\downarrow$$
$$G_5 : X = Y + Z.$$
$$\downarrow$$
$$\diamond$$

(b) Symbolic derivation for goal `widget(X)`

[Islam, Muhammad Asiful, C. R. Ramakrishnan, and I. V. Ramakrishnan. "Inference in probabilistic logic programs with continuous random variables." TPLP 2012].

Main idea

If a derivation with $m = a$, then X is the convolution of Y and Z, i.e.

$$X \sim \mathcal{N}(2.5, 1.1)$$

Similarly, if $m = b$, then:

$$X \sim \mathcal{N}(3.5, 1.1)$$

Since they are mutually exclusive:

$$p(X) = 0.3\mathcal{N}(2.5, 1.1) + 0.7\mathcal{N}(3.5, 1.1)$$

- Implemented in XSB
- Exact inference
- Restricted language
- Restricted to distributions (stable distributions)

Distributional clauses

- Distributional clause:

$$h \sim D \leftarrow b_1, \ldots, b_n$$

- Meaning: each ground instance of the clause $(h \sim D \leftarrow b_1, \ldots, b_n)\theta$ defines the random variable $h\theta$ with distribution $D\theta$ whenever all the $b_i\theta$ hold, where θ is a substitution

- D can be any parameterised distribution, e.g. normal, Gamma, uniform, etc.

Distributional clauses

$$\mathtt{n} \sim \mathtt{uniform}([1,2,3,4,5,6,7,8,9,10]). \tag{7}$$

$$\mathtt{color}(X) \sim \mathtt{uniform}([\mathtt{grey}, \mathtt{blue}, \mathtt{black}]) \leftarrow \mathtt{material}(X) \sim= \mathtt{metal}. \tag{8}$$

$$\mathtt{color}(X) \sim \mathtt{uniform}([\mathtt{black}, \mathtt{brown}]) \leftarrow \mathtt{material}(X) \sim= \mathtt{wood}. \tag{9}$$

$$\mathtt{material}(X) \sim \mathtt{finite}([0.3 : \mathtt{wood}, 0.7 : \mathtt{metal}]) \leftarrow \mathtt{n} \sim= N, \mathtt{between}(1, N, X). \tag{10}$$

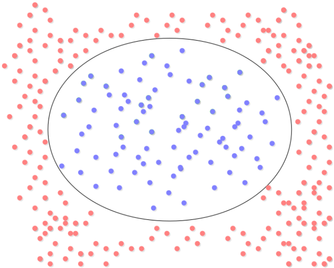
$$\mathtt{drawn}(Y) \sim \mathtt{uniform}(L) \leftarrow \mathtt{n} \sim= N, \mathtt{findall}(X, \mathtt{between}(1, N, X), L). \tag{11}$$

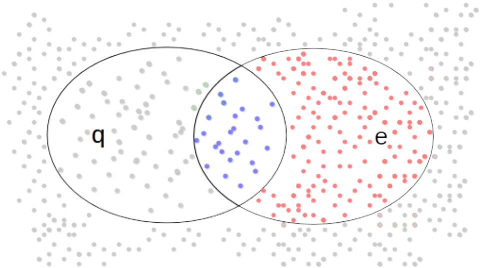
$$\mathtt{size}(X) \sim \mathtt{beta}(2, 3) \leftarrow \mathtt{material}(X) \sim= \mathtt{metal}. \tag{12}$$

$$\mathtt{size}(X) \sim \mathtt{beta}(4, 2) \leftarrow \mathtt{material}(X) \sim= \mathtt{wood}. \tag{13}$$

[Davide Nitti, Tinne De Laet, Luc De Raedt. Probabilistic logic programming for hybrid relational domains. Machine Learning, Springer, 2016]

Sampling: basic idea





$$\widetilde{P}(q) = \frac{\#{\color{blue}\bullet}}{\#{\color{blue}\bullet} + \#{\color{red}\bullet}}$$

$$\widetilde{P}(q \mid e) = \frac{\#{\color{blue}\bullet}}{\#{\color{blue}\bullet} + \#{\color{red}\bullet}}$$

Distributional clauses (DC): inference

1: $(\mathtt{color(2)} \sim= \mathtt{black}); w_q^{(i)} = 1; x^{P(i)} = \emptyset$

↓ 2b on (8) :

2: $(\mathtt{material(2)} \sim= \mathtt{metal}, \mathtt{sample}(\mathtt{color(2)}, \mathcal{D}_{\mathtt{color(2)}}), \mathtt{color(2)} \sim= \mathtt{black}); w_q^{(i)} = 1; x^{P(i)} = \emptyset$

↓ 2b on (10) :

3: $(\mathtt{n} \sim= \mathtt{N}, \mathtt{between(1,N,2)}, \mathtt{sample}(\mathtt{material(2)}, \mathcal{D}_{\mathtt{material(2)}}), \mathtt{material(2)} \sim= \mathtt{metal},$
$\mathtt{sample}(\mathtt{color(2)}, \mathcal{D}_{\mathtt{color(2)}}), \mathtt{color(2)} \sim= \mathtt{black}); w_q^{(i)} = 1;\ x^{P(i)} = \emptyset$

↓ 2b on (7) :

4: $(\mathtt{sample}(\mathtt{n}, \mathcal{D}_{\mathtt{n}}), \mathtt{n} \sim= \mathtt{N}, \mathtt{between(1,N,2)}, \mathtt{sample}(\mathtt{material(2)}, \mathcal{D}_{\mathtt{material(2)}}),$
$\mathtt{material(2)} \sim= \mathtt{metal}, \mathtt{sample}(\mathtt{color(2)}, \mathcal{D}_{\mathtt{color(2)}}), \mathtt{color(2)} \sim= \mathtt{black}); w_q^{(i)} = 1; x^{P(i)} = \emptyset$

↓ 3b :

5: $(\mathtt{n} \sim= \mathtt{N}, \mathtt{between(1,N,2)}, \mathtt{sample}(\mathtt{material(2)}, \mathcal{D}_{\mathtt{material(2)}}), \mathtt{material(2)} \sim= \mathtt{metal},$
$\mathtt{sample}(\mathtt{color(2)}, \mathcal{D}_{\mathtt{color(2)}}), \mathtt{color(2)} \sim= \mathtt{black}); w_q^{(i)} = 1;\ x^{P(i)} = \{\mathtt{n} = 3\}$

↓ 2a followed by 1a

6: $(\mathtt{sample}(\mathtt{material(2)}, \mathcal{D}_{\mathtt{material(2)}}), \mathtt{material(2)} \sim= \mathtt{metal}, \mathtt{sample}(\mathtt{color(2)}, \mathcal{D}_{\mathtt{color(2)}})$
$\mathtt{color(2)} \sim= \mathtt{black}); w_q^{(i)} = 1;\ x^{P(i)} = \{\mathtt{n} = 3\}$

↓ 3b :

7: $(\mathtt{material(2)} \sim= \mathtt{metal}, \mathtt{sample}(\mathtt{color(2)}, \mathcal{D}_{\mathtt{color(2)}}), \mathtt{color(2)} \sim= \mathtt{black})$
↓ $w_q^{(i)} = 1;\ x^{P(i)} = \{\mathtt{n} = 3, \mathtt{material(2)} = \mathtt{wood}\}$

8: fail, backtracking to 1

↓ 2b on (9) :

9: $(\mathtt{material(2)} \sim= \mathtt{wood}, \mathtt{sample}(\mathtt{color(2)}, \mathcal{D}_{\mathtt{color(2)}}), \mathtt{color(2)} \sim= \mathtt{black})$
$w_q^{(i)} = 1;\ x^{P(i)} = \{\mathtt{n} = 3, \mathtt{material(2)} = \mathtt{wood}\}$

↓ 2a :

10: $(\mathtt{sample}(\mathtt{color(2)}, \mathcal{D}_{\mathtt{color(2)}}), \mathtt{color(2)} \sim= \mathtt{black}); w = 1; x^{P(i)} = \{\mathtt{n} = 3, \mathtt{material(2)} = \mathtt{wood}\}$

IHPMC setting

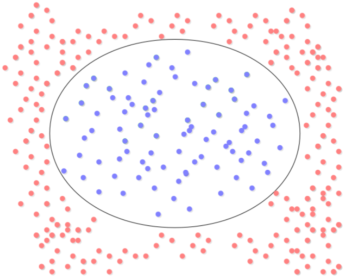
- Distributional clauses, but with fixed distributions

$$\begin{aligned}
&\mathit{fails}(\mathit{Comp}) \leftarrow \mathbf{FailCause}(\mathit{Comp}, \mathit{Cause}) = \mathit{true}\\
&\mathit{fails}(\mathit{Comp}) \leftarrow \mathbf{Temp} > \mathbf{Limit}(\mathit{Comp})\\
&\mathit{fails}(\mathit{Comp}) \leftarrow \mathit{subcomp}(\mathit{Subcomp}, \mathit{Comp}),\ \mathit{fails}(\mathit{Subcomp})\\
\\
&\mathbf{FailCause}(\mathit{engine}, \mathit{noFuel}) \sim \{0.0002: \mathit{true}, 0.9998: \mathit{false}\}\\
&\mathbf{Temp} \sim \Gamma(20.0, 5.0)\\
&\mathbf{Limit}(\mathit{engine}) \sim \mathcal{N}(65.0, 5.0)\\
&\mathit{subcomp}(\mathit{fuelPump}, \mathit{engine})\\
&\mathbf{Limit}(\mathit{fuelPump}) \sim \mathcal{N}(80.0, 5.0)\\
&\ldots
\end{aligned}$$

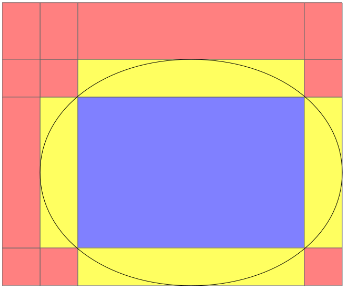
- Probability of query event q, given evidence e: $P(q \mid e)$

$$P\big(\mathit{fails}(\mathit{fuelPump}) \mid \mathit{fails}(\mathit{engine})\big)$$

IHPMC: basic idea



$$\widetilde{P}(q) = \frac{\#\textcolor{blue}{\bullet}}{\#\textcolor{blue}{\bullet} + \#\textcolor{red}{\bullet}}$$



$$\underline{P}(q) = P(\textcolor{blue}{\blacksquare})$$

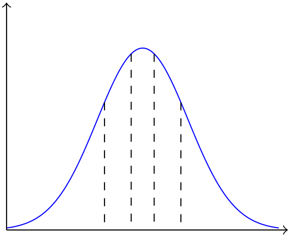
$$\overline{P}(q) = P(\textcolor{blue}{\blacksquare}) + P(\textcolor{yellow}{\blacksquare})$$

$$\widetilde{P}(q) = P(\textcolor{blue}{\blacksquare}) + P(\textcolor{yellow}{\blacksquare})/2 \pm P(\textcolor{yellow}{\blacksquare})/2$$

Open Universiteit
www.ou.nl

Approximating continuous distributions

- Desired: usage of known PDFs
- Example: $X \sim \mathcal{N}(3.0, 1.0)$
- Aim: discretisation of PDFs



$$\mathbf{X} \sim \{0.2 : \mathbf{X} < 2.15837876642709,$$
$$0.2 : 2.15837876642709 < \mathbf{X} < 2.7466528968642, \ldots\}$$

Generalised distribution semantics

- Sato's distribution semantics
  - $P_f(f_1, f_2, \ldots)$ on binary probabilistic facts
  - $h \leftarrow l_1, \ldots, l_n, f_1, \ldots, f_m$
  - P_f can uniquely be extended to all atoms in rules
- Generalised distribution semantics
  - $P_{\mathbf{V}}(\mathbf{V}_1, \mathbf{V}_2, \ldots)$ on random variables with arbitrary ranges
  - $h \leftarrow l_1, \ldots, l_n, \varphi_1(\mathbf{V}_1, \mathbf{V}_2, \ldots), \ldots, \varphi_m(\mathbf{V}_1, \mathbf{V}_2, \ldots)$
  - $P_{\mathbf{V}}$ can uniquely be extended to all atoms in rules

[Michels, S., Hommersom, A., Lucas, P. J., & Velikova, M. (2015). A new probabilistic constraint logic programming language based on a generalised distribution semantics. Artificial Intelligence, 2015]

Credal sets of these distributions

- Most $P_{\mathbf{V}}$: no analytic solutions
- Credal sets of probability distributions $\mathbf{P}$
- $\mathbf{V_i}(X_1, \ldots, X_n) \sim \{p_1 : \varphi_1, \ldots, p_l : \varphi_m\}$ define *non-empty* $\mathbf{P}$, e.g.

$$\mathbf{X} \sim \{0.2 : \mathbf{X} < 2, 0.8 : \mathbf{X} \geq 2\}$$

- Clauses extend each distribution in the credal set (e.g. $q \leftarrow \mathbf{X} > 0$)
- Exact inference conditions (bounds: $min/max\{P(q) \mid P \in \mathbf{P}\}$)
  - *finite-support condition* [Sato, 1995]
    (counterexample: $q(X) \leftarrow \mathbf{V_1} > X,\ q(X+1)$)
  - *finite-dimensional-constraints condition*
    (counterexample: $q \leftarrow \underset{i,j \in \mathbb{N}}{\forall} i > j \rightarrow \mathbf{V_i} > \mathbf{V_j}$)
  - *disjoint-events-decidability condition*
    ($e_1 \cap e_2 = \emptyset$ decidable, implementation:
    $check(\varphi_1 \wedge \varphi_2) = unsat$)

Example (adapted from [Binder et al., 1997])

$$\mathbf{Yield}(\mathit{apple}) \sim \text{(a discretized normal)}$$
$$\mathbf{Yield}(\mathit{banana}) \sim \ldots$$

$$\mathbf{Support}(\mathit{apple}) \sim \{0.3\colon \mathit{yes}, 0.7\colon \mathit{no}\}$$
$$\mathbf{Support}(\mathit{banana}) \sim \{0.5\colon \mathit{yes}, 0.5\colon \mathit{no}\}$$

$$\mathit{basic\_price}(\mathit{apple}, 250 - 0.007 \cdot \mathbf{Yield}(\mathit{apple}))$$
$$\mathit{basic\_price}(\mathit{banana}, 200 - 0.006 \cdot \mathbf{Yield}(\mathit{banana}))$$

$$\mathit{price}(\mathit{Fruit}, \mathit{BPrice} + 50) \leftarrow$$
$$\mathit{basic\_price}(\mathit{Fruit}, \mathit{BPrice}), \mathbf{Support}(\mathit{Fruit}) = \mathit{yes}$$
$$\mathit{price}(\mathit{Fruit}, \mathit{BPrice}) \leftarrow$$
$$\mathit{basic\_price}(\mathit{Fruit}, \mathit{BPrice}), \mathbf{Support}(\mathit{Fruit}) = \mathit{no}$$

Example (adapted from [Binder et al., 1997])

$$\begin{aligned} \textbf{Max\_price}(\mathit{apple}) &\sim \text{(a discretised Gamma)} \\ \textbf{Max\_price}(\mathit{banana}) &\sim \ldots \end{aligned}$$

$$\mathit{buy}(\mathit{Fruit}) \leftarrow \mathit{price}(\mathit{Fruit}, P),\ P \leq \textbf{Max\_price}(\mathit{Fruit})$$

$$\begin{aligned} P(\mathit{buy}(\mathit{apple})) &\approx 0.464 \pm 0.031 \\ P(\mathit{buy}(\mathit{banana})) &\approx 0.162 \pm 0.031 \\ P(\mathit{buy}(\mathit{apple}) \vee \mathit{buy}(\mathit{banana})) &\approx 0.552 \pm 0.054 \end{aligned}$$

Inference algorithm: weighted model counting

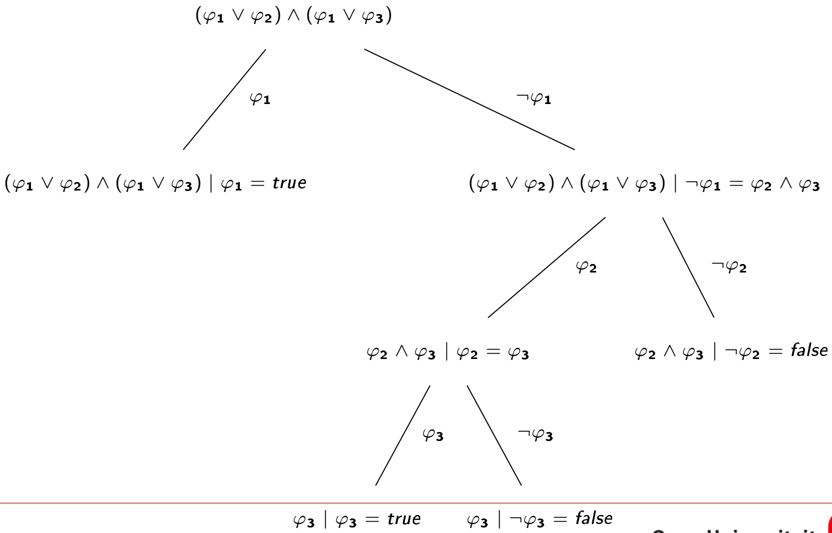
- Weighted Boolean formula consists of
  - A propositional formula ϕ
  - Weights on every literal $w(l)$
- Weighted model counting (WMC) gives

$$weight(\phi) = \sum_{M \models \phi} \prod_{l \in M} w(l)$$

- Any discrete PLP query can be converted to a WMC problem
  - Equivalent formula involving independent random variables only
  - Exploiting local structure $\Rightarrow$ especially important in PLP
  - Implemented in Problog2

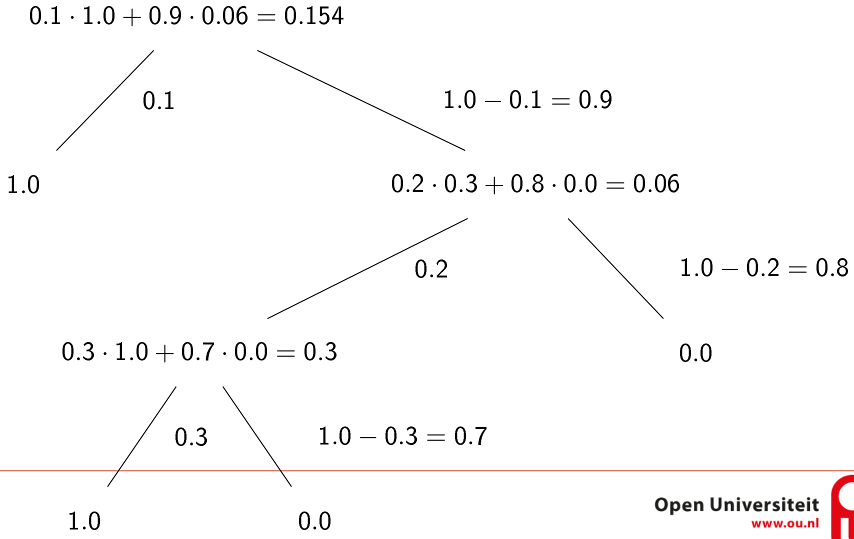
[Fierens, Daan, et al. Inference and learning in probabilistic logic programs using weighted Boolean formulas. TPLP 2015]

DPLL example



DPLL for WMC example

$P(\varphi_1) = 0.1 \qquad P(\varphi_2) = 0.2 \qquad P(\varphi_3) = 0.3$



Open Universiteit
www.ou.nl

Inference with credal sets

- Generalised WMC

WMC	GWMC
True/False choices	Choices in definitions
Simplification of CNF	Simplification of constraint
Single probabilities	Probability tuples

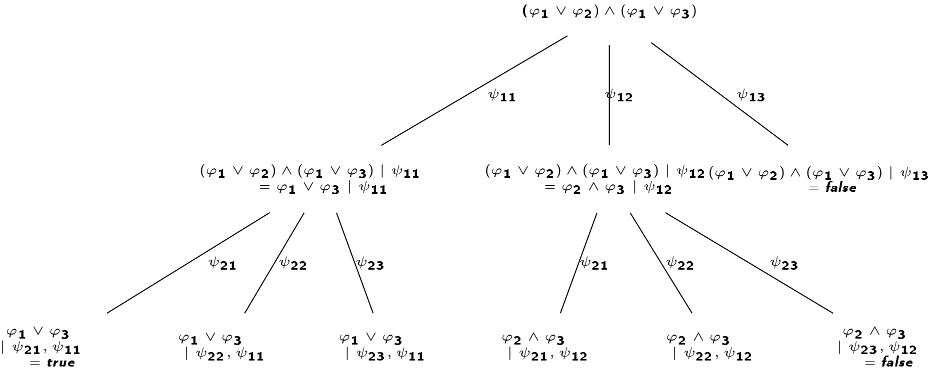
- Same complexity
- Exploitation of additional structure (determinism, ...)

Generalised WMC example

$$\varphi_1 = \mathbf{V}_1 < 0.75 \quad \varphi_2 = \mathbf{V}_1 < 1.25 \quad \varphi_3 = \mathbf{V}_1 + 0.25 \cdot \mathbf{V}_2 < 1.375$$

$$\mathbf{V}_1 \sim \{0.7\colon 0 \leq \mathbf{V}_1 \leq 1, 0.2\colon 1 \leq \mathbf{V}_1 \leq 2, 0.1\colon 2 \leq \mathbf{V}_1 \leq 3\}$$

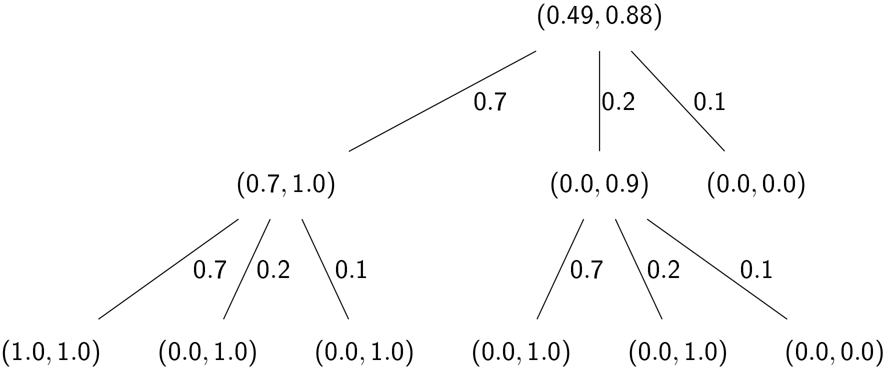
$$\mathbf{V}_2 \sim \{0.7\colon 0 \leq \mathbf{V}_2 \leq 1, 0.2\colon 1 \leq \mathbf{V}_2 \leq 2, 0.1\colon 2 \leq \mathbf{V}_2 \leq 3\}$$



Generalised WMC example (2)

$$\mathbf{V_1} \sim \{0.7 \colon 0 \leq \mathbf{V_1} \leq 1, 0.2 \colon 1 \leq \mathbf{V_1} \leq 2, 0.1 \colon 2 \leq \mathbf{V_1} \leq 3\}$$

$$\mathbf{V_2} \sim \{0.7 \colon 0 \leq \mathbf{V_2} \leq 1, 0.2 \colon 1 \leq \mathbf{V_2} \leq 2, 0.1 \colon 2 \leq \mathbf{V_2} \leq 3\}$$



Open Universiteit
www.ou.nl

Iterative Hybrid Probabilistic Model Counting

- Generalizes the idea above in an *iterative* way
  - Automatic refinement of continuous distributions
  - Constructs a *hybrid probability tree* on the fly

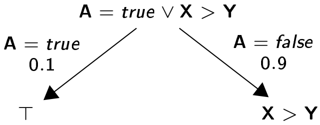
- Main theoretical result:
  Approximations with arbitrary precision can be computed in finite time!
  For all events q and e and every maximal error ϵ, IHPMC can in finite time find an approximation such that:

$$P(q \mid e) - \underline{P}(q \mid e) \leq \epsilon \text{ and}$$
$$\overline{P}(q \mid e) - P(q \mid e) \leq \epsilon$$

[Steffen Michels, Arjen Hommersom, Peter J. F. Lucas. Approximate Probabilistic Inference with Bounded Error for Hybrid Probabilistic Logic Programming. IJCAI'16]

Example HPT

Hybrid Probability Tree (HPT)

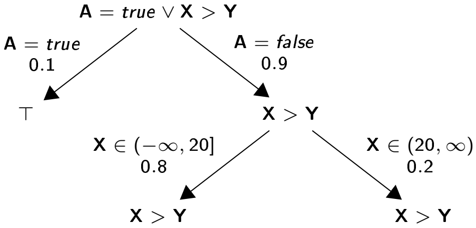


$P(\mathbf{A} = \mathit{true} \lor \mathbf{X} > \mathbf{Y}) \in [0.1, 1.0]$
$\widetilde{P}(\mathbf{A} = \mathit{true} \lor \mathbf{X} > \mathbf{Y}) = 0.55 \pm 0.45$

- Similar to binary WMC
- Exploits logical structure
- Search towards hyperrectangles with high probability

Example HPT

Hybrid Probability Tree (HPT)



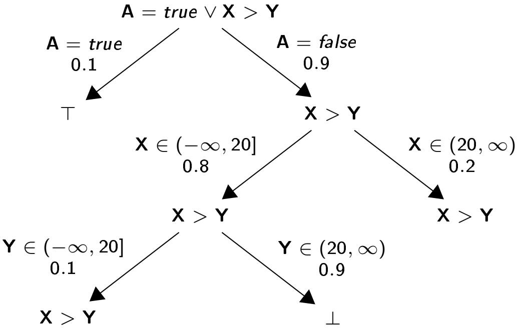
$P(\mathbf{A} = \mathit{true} \vee \mathbf{X} > \mathbf{Y}) \in [0.1, 1.0]$
$\widetilde{P}(\mathbf{A} = \mathit{true} \vee \mathbf{X} > \mathbf{Y}) = 0.55 \pm 0.45$

$P(\mathbf{A} = \mathit{true} \vee \mathbf{X} > \mathbf{Y}) \in [0.1, 1.0]$
$\widetilde{P}(\mathbf{A} = \mathit{true} \vee \mathbf{X} > \mathbf{Y}) = 0.55 \pm 0.45$

- Similar to binary WMC
- Exploits logical structure
- Search towards hyperrectangles with high probability

Example HPT

Hybrid Probability Tree (HPT)



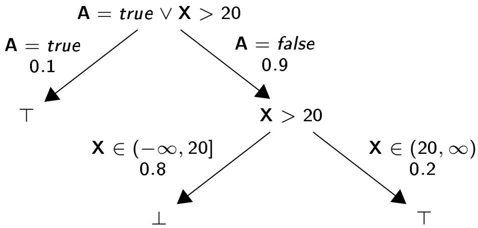
$$P(\mathbf{A} = \mathit{true} \vee \mathbf{X} > \mathbf{Y}) \in [0.1, 1.0]$$
$$\widetilde{P}(\mathbf{A} = \mathit{true} \vee \mathbf{X} > \mathbf{Y}) = 0.55 \pm 0.45$$

$$P(\mathbf{A} = \mathit{true} \vee \mathbf{X} > \mathbf{Y}) \in [0.1, 1.0]$$
$$\widetilde{P}(\mathbf{A} = \mathit{true} \vee \mathbf{X} > \mathbf{Y}) = 0.55 \pm 0.45$$

$$P(\mathbf{A} = \mathit{true} \vee \mathbf{X} > \mathbf{Y}) \in [0.1, 0.352]$$
$$\widetilde{P}(\mathbf{A} = \mathit{true} \vee \mathbf{X} > \mathbf{Y}) = 0.226 \pm 0.126$$

- Similar to binary WMC
- Exploits logical structure
- Search towards hyperrectangles with high probability

Special case: one-dimensional constraints



$$P(\mathbf{A} = true \vee \mathbf{X} > \mathbf{Y}) \in [0.1, 1.0]$$
$$\widetilde{P}(\mathbf{A} = true \vee \mathbf{X} > \mathbf{Y}) = 0.55 \pm 0.45$$

$$P(\mathbf{A} = true \vee \mathbf{X} > \mathbf{Y}) = 0.28$$

- Equivalent to Hybrid ProbLog
- There exists a finite HPT where all leafs are either $\top$ or $\bot$
- Inference is *exact*

[Gutmann, Bernd, Manfred Jaeger, and Luc De Raedt. Extending ProbLog with Continuous Distributions. ILP'10]

Open Universiteit
www.ou.nl

Evaluation

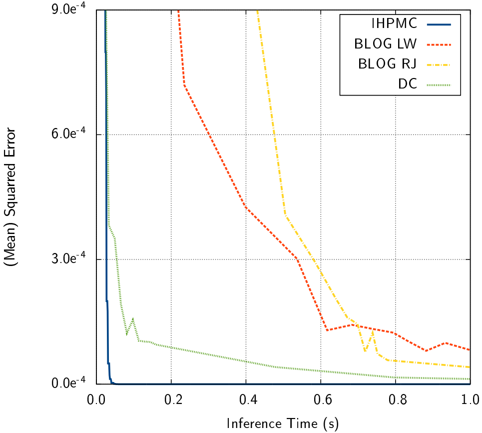
- Diagnosis Problem

$$
\begin{aligned}
\textbf{Break}(\_) &\sim \{p\colon \mathit{true}, 1-p\colon \mathit{false}\} \\
\textbf{Temp} &\sim \mathcal{N}(20.0, 5.0) \\
\textbf{Limit}(\_) &\sim \mathcal{N}(\mu, \quad 5.0)
\end{aligned}
$$

$$
\begin{aligned}
&\mathit{fails}(i) \leftarrow \textbf{Break}(i) = \mathit{true} \\
&\mathit{fails}(i) \leftarrow \textbf{Temp} > \textbf{Limit}(i) \\
&\mathit{fails}(i) \leftarrow i \neq 0,\ \mathit{fails}(i-1)
\end{aligned}
$$

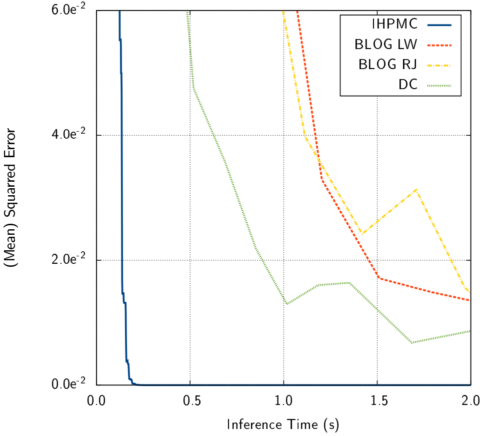
- Comparison with state-of-the-art samplers
  - *BLOG*: Rejection Sampler
  - *BLOG*: Likelihood Weighting Sampler
  - *Distributional Clauses*: Sequential Monte Carlo Sampler

No evidence



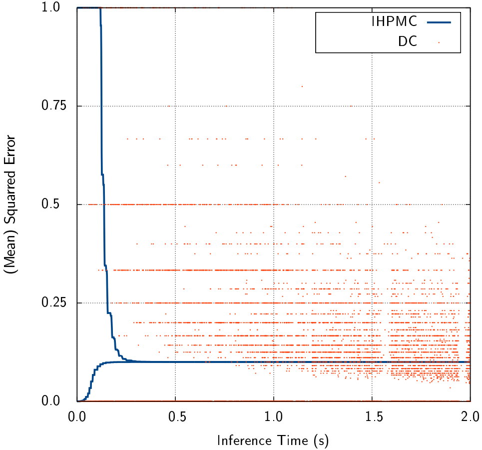
$$P(fails(9)),\ p = 0.01,\ \mu = 60.0$$

Rare observed event



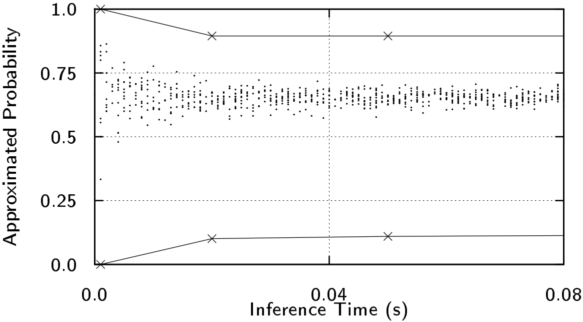
$$P(\textit{fails}(9) \mid \textit{fails}(0)),\ p = 0.0001,\ \mu = 60.0$$

Approximations rare observed event



$$P(fails(9) \mid fails(0)),\ p = 0.0001,\ \mu = 60.0$$

When IHPMC fails. . .



Summary of inference methods

Method	Exact	Rejection / Importance Sampling	MCMC	**IHPMC**
Works for	limited number of problems	(virtually) all problems	(virtually) all problems	large class of hybrid problems
Quality guarantee	no error	probabilistic	none	bounded error
Structure-sensitive	yes	no	hand-tailored solution often required	yes
Sensitive to rare evidence	no	yes	no	no

Conclusions and future work

Conclusions:

- Nowadays there are good alternatives for inference in hybrid PLPs
- For special cases exact inference possible
- Sampling generic and often performs well
- IHPMC provides alternative to sampling
  - insensitive to rare observed events
  - no hand-tailoring
  - bounded error
  - may fail, **but lets the user know!**
  - Try it: `http://www.steffen-michels.de/ihpmc`

Possible directions for future research:

- Variational inference: scalable technique for approximate inference
- Conditioning on continuous variables