Weighted Model Integration Using Knowledge Compilation

Pedro Zuidberg Dos Martires, Anton Dries, Luc De Raedt
pedro.zuidbergdosmartires@cs.kuleuven.be

Probabilistic Inference

Probabilistic inference algorithms are targeted towards:

- either continuous distributions: symbolic inference, Hamilton Monte Carlo, variational Bayesian Inference, ...
- or discrete distributions: SAT, weighted model counting, ...

We want to combine state-of-the-art from both
→ best of both worlds!

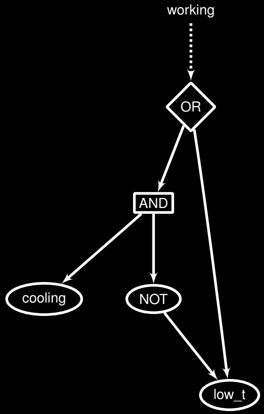
We tackle the problem starting from a discrete perspective.

Knowledge Compilation[1]

State-of-the-art technique for probabilistic inference in discrete domain.

Probabilistic inference is #P-complete.

$$\texttt{working} \leftrightarrow \texttt{cooling} \vee \texttt{low\_t}$$



offline: compile theory (expensive)
online: fast inference (cheap)

- evaluation in linear time
- conditioning in poly-time
- repeated querying

[1] Adnan Darwiche. *Modeling and reasoning with Bayesian networks.*

SMT: Satisfiability Modulo Theory

$$\mathtt{working} \leftrightarrow (\mathtt{cooling} \wedge (\mathtt{t}^2 < 30)) \vee (\mathtt{t} < 5)$$

More complex expressions allowed:

$$(\mathtt{t}^2 < \mathtt{s} + 10)$$

WMI: Probability of SMT Formulas

$$\texttt{working} \leftrightarrow (\texttt{cooling} \wedge (\texttt{t}^2 < 30)) \vee (\texttt{t} < 5)$$

$$p(\texttt{cooling}) = 0.99$$

$$\texttt{t} \sim N_{\texttt{t}}(20, 5)$$

Question:

$$p(\texttt{working}) = ?$$

In general:

$$p(x|e) = \frac{p(e|x)p(x)}{\not{\Sigma}\!\!\int_x p(x, e)}$$

	WMC[2]	prob. prog.[3 4 5]	previous WMI[6 7]	our work
knowledge compilation	✓	✗	✓/✗	✓
density functions	✗	✓	✗	✓
exact	✓	✗	✓	✓
approximate		✓	✗	✓
polynomials	✗	✓	✓	✓
non-linear	✗	✓	✗	✓

[2] Mark Chavira and Adnan Darwiche. "On Probabilistic Inference by Weighted Model Counting".

[3] Timon Gehr, Sasa Misailovic, and Martin Vechev. "PSI: Exact Symbolic Inference for Probabilistic Programs".

[4] Davide Nitti, Tinne De Laet, and Luc De Raedt. "Probabilistic logic programming for hybrid relational domains".

[5] Brian Milch, Bhaskara Marthi, and Stuart Russell. "BLOG: Relational modeling with unknown objects".

[6] Samuel Kolb et al. "Efficient Symbolic Integration for Probabilistic Inference".

[7] Paolo Morettin, Andrea Passerini, and Roberto Sebastiani. "Efficient Weighted Model Integration via SMT-Based Predicate Abstraction".

Contribution

1. Handle probability density functions while applying state-of-the-art knowledge compilation techniques.
2. Two new solvers:
   - Exact solver Symbo: PSI-Solver[8] in back-end (probabilistic computer algebra system)
   - Approximate solver Sampo: Edward[9] in back-end (probabilistic TensorFlow)

[8] Gehr, Misailovic, and Vechev, "PSI: Exact Symbolic Inference for Probabilistic Programs"

[9] Dustin Tran et al. "Edward: A library for probabilistic modeling, inference, and criticism".

Symbo: Exact Symbolic Inference

1. Abstract theory.

$$\texttt{working} \leftrightarrow (\texttt{cooling} \wedge (\texttt{t}^2 < 30)) \vee (\texttt{t} < 5)$$

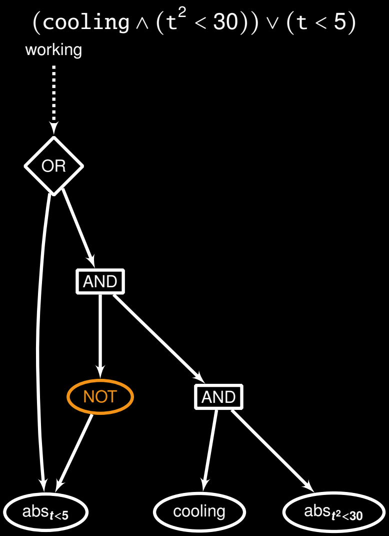
$$\texttt{working} \leftrightarrow (\texttt{cooling} \wedge \mathsf{abs}_{\texttt{t}^2 < 30}) \vee \mathsf{abs}_{\texttt{t} < 5}$$

Introduce fresh Boolean variables for conditions.

Symbo: Exact Symbolic Inference

1. Abstract theory.
2. Compile formula.

$(\texttt{cooling} \wedge \mathrm{abs}_{t^2<30}) \vee \mathrm{abs}_{t<5} \leftrightarrow$

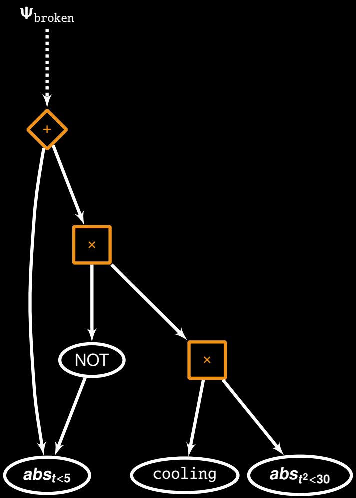


Avoid double counting.

Symbo: Exact Symbolic Inference

1. Abstract theory.
2. Compile formula.
3. To arithmetic circuit.

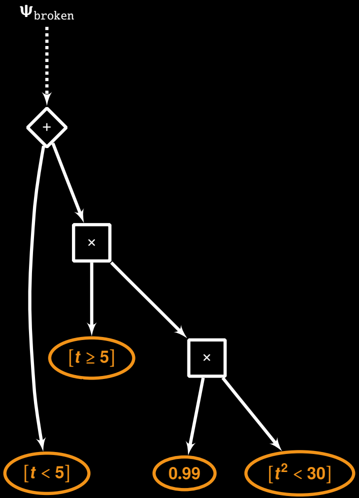
$$(\mathtt{cooling} \wedge (\mathtt{t}^2 < 30)) \vee (\mathtt{t} < 5)$$



Symbo: Exact Symbolic Inference

1. Abstract theory.
2. Compile formula.
3. To arithmetic circuit.
4. Label the leaves.

$(\mathtt{cooling} \wedge (\mathtt{t}^2 < 30)) \vee (\mathtt{t} < 5)$

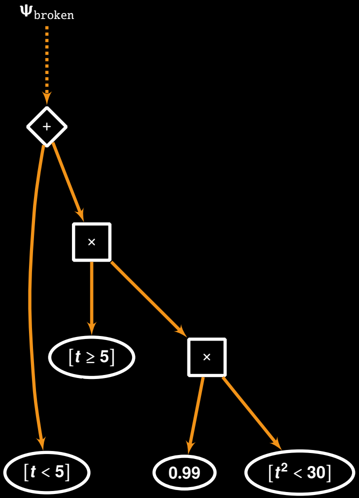


Symbo: Exact Symbolic Inference

1. Abstract theory.
2. Compile formula.
3. To arithmetic circuit.
4. Label the literals.
5. Evaluate.

$$[\mathtt{t}<5] + 0.99[\mathtt{t}^2<30][\mathtt{t}\geq 5]$$

$(\mathtt{cooling} \wedge (\mathtt{t}^2 < 30)) \vee (\mathtt{t} < 5)$



Algebraic Model Counting[10]

Generalized framework for probabilistic inference:

- define specific semiring $(A, \oplus, \otimes, e^{\oplus}, e^{\otimes})$ for specific task

Link to belief propagation:

- sum-product: $\oplus$ is normal addition
- max-product: $\oplus$ is maximization

We defined a custom probability density semiring with custom elements:

$$A := \{(a, V(a))\}$$

$$a = [\mathtt{t}{<}5] + 0.99[\mathtt{t}^2{<}30][\mathtt{t}{\geq}5]$$
$$V(a) = \{t\}$$

[10] Angelika Kimmig, Guy Van den Broeck, and Luc De Raedt. "Algebraic Model Counting".

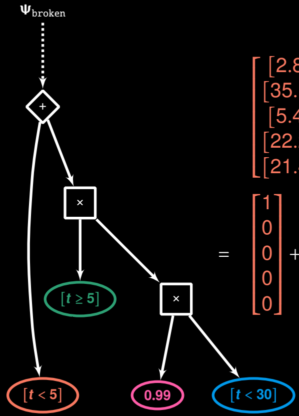
Symbo: Exact Symbolic Inference

1. Abstract theory.
2. Compile formula.
3. To arithmetic circuit.
4. Label the leaves.
5. Evaluate.
6. Multiply by the weight of the continuous variables.
7. Integrate.

$$p(\mathtt{working}) = \int \left([\mathtt{t}<5] + 0.99[\mathtt{t}^2<30][\mathtt{t}\geq 5] \right) N_{\mathtt{t}}(20, 5) d\mathtt{t}$$

Integrals become easily intractable.

Sampo: Approximate MC Inference



$$t \approx [2.8, 35.1, 5.4, 22.2, 21.4]$$

$$\begin{bmatrix} [2.8<5] \\ [35.1<5] \\ [5.4<5] \\ [22.2<5] \\ [21.4<5] \end{bmatrix} + \begin{bmatrix} [2.8\geq5] \\ [35.1\geq5] \\ [5.4\geq5] \\ [22.2\geq5] \\ [21.4\geq5] \end{bmatrix} \times \begin{bmatrix} 0.99 \\ 0.99 \\ 0.99 \\ 0.99 \\ 0.99 \end{bmatrix} \times \begin{bmatrix} [7.84<30] \\ [1232.01<30] \\ [29.16<30] \\ [492.84<30] \\ [457.96<30] \end{bmatrix}$$

$$= \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \times \begin{bmatrix} 0.99 \\ 0.99 \\ 0.99 \\ 0.99 \\ 0.99 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0.99 \\ 0 \\ 0 \end{bmatrix}$$

Sampo: Approximate MC Inference

$$\begin{bmatrix} [2.8<5] \\ [35.1<5] \\ [5.4<5] \\ [22.2<5] \\ [21.4<5] \end{bmatrix} + \begin{bmatrix} [2.8\geq5] \\ [35.1\geq5] \\ [5.4\geq5] \\ [22.2\geq5] \\ [21.4\geq5] \end{bmatrix} \times \begin{bmatrix} 0.99 \\ 0.99 \\ 0.99 \\ 0.99 \\ 0.99 \end{bmatrix} \times \begin{bmatrix} [7.84<30] \\ [1232.01<30] \\ [29.16<30] \\ [492.84<30] \\ [457.96<30] \end{bmatrix}$$

$$= \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \times \begin{bmatrix} 0.99 \\ 0.99 \\ 0.99 \\ 0.99 \\ 0.99 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0.99 \\ 0 \\ 0 \end{bmatrix}$$

$$p(\mathtt{broken}) = \frac{1}{5} \sum_{i=1}^{5} \Psi^{\mathrm{MC}}_{\mathtt{broken},i} = 1.99/5 = 0.398$$

This is pure vector calculus and can be executed on the GPU!
→ cheap probabilistic inference
→ embarrassingly parallelizable

Symbo vs. PSI[11]

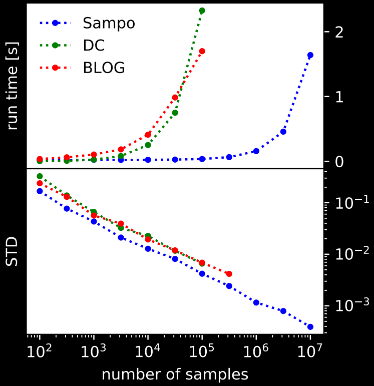
How does symbolico-logic inference compare to pure symbolic inference?

- Symbo is faster on 9/10 benchmark problems than PSI, excluding knowledge compilation
- Symbo is faster on 7/10 benchmark problems than PSI, including knowledge compilation

Logical reasoning generally improves symbolic inference!

[11] Gehr, Misailovic, and Vechev, “PSI: Exact Symbolic Inference for Probabilistic Programs”

Sampo vs. DC[12] vs. BLOG[13]



Sampling on the GPU → constant time complexity
Avoid sampling categorical variables → reduction in variance

---

[12] Nitti, De Laet, and De Raedt, “Probabilistic logic programming for hybrid relational domains”

[13] Milch, Marthi, and Russell, “BLOG: Relational modeling with unknown objects”

Constributions

- Unified framework for knowledge compilation and weighted model integration based on semirings and AMC.
- Introduced two solvers that beat state-of-the-art.
- Sampo is the first sampling based algorithm for WMI.

Future Work

- Integrate Symbo and Sampo into full-fledged probabilistic programming language
- investigate thoroughly relationship to related work[14] [15].

[14] Kolb et al., "Efficient Symbolic Integration for Probabilistic Inference"

[15] Morettin, Passerini, and Sebastiani, "Efficient Weighted Model Integration via SMT-Based Predicate Abstraction"