

# Probabilistic Inference in Hybrid Domains

Paolo Morettin

Structured Machine Learning Group, University of Trento, Italy  
 paolo.morettin@unitn.it

## Abstract

This extended abstract presents my PhD research project on learning and reasoning in Hybrid Domains. In particular, it focuses on my current work on exact probabilistic inference in these domains, as well as presenting other research directions that are going to be explored.

## 1 Introduction

Hybrid Domains are characterized by a mixture of continuous and discrete variables, as well as logical and algebraic constraints involving them. Reasoning in these domains is a daunting task and currently few approaches have been proposed. Developing methods that can efficiently handle Hybrid Domains is of primary importance, as they would allow to model a wide range of problems in a more expressive way, as well as performing learning and inference tasks more efficiently by leveraging the knowledge encoded in the constraints.

Currently, my research focuses on performing exact probabilistic inference in Hybrid Domains by leveraging Weighted Model Integration, a generalization of Weighted Model Counting to continuous distributions.

## 2 Background

Weighted Model Counting (WMC) [Chavira and Darwiche, 2008] is the problem of computing the weighted sum of the models of a propositional formula.

$$\text{WMC}(\varphi, w) = \sum_{\mu \models \varphi} \text{WEIGHT}(\mu, w)$$

where typically the weight of each model  $\mu$  is computed as the product of the weights associated to its literals  $\ell$ :

$$\text{WEIGHT}(\mu, w) = \prod_{\ell \in \mu} w(\ell)$$

Recently, it was shown that probabilistic inference in the case of discrete distributions can be posed as a WMC problem [Chavira and Darwiche, 2008; Choi *et al.*, 2013; Suciu *et al.*, 2011; Fierens *et al.*, 2013].

Satisfiability Modulo Theories (SMT) [Barrett *et al.*, 2009] generalizes propositional satisfiability to theories such as Linear Arithmetic over Real or Integers. A SMT formula is a

(typically quantifier-free) formula in which the atoms can either be Boolean or theory atoms. Encoding a problem in SMT allows to represent logical and algebraic constraints involving the variables, like:

$$(A \rightarrow ((0 \leq x) \wedge (x < 3))) \wedge (A \vee \neg(x + y < 10))$$

Weighted Model Integration [Belle *et al.*, 2015a] (WMI) is a recent generalization of WMC to the case of continuous variables, in which the problem is encoded as a SMT formula and the weighted sum of models is replaced with the sum of the definite integrals of the weight functions of each model, which act as unnormalized density functions.

$$\text{WMI}(\varphi, w) = \sum_{\mu \models \varphi} \int_{\mu \in \mathcal{LRA}} \text{WEIGHT}(\mu, w) dx$$

The integral is computed over the subspace defined by the truth assignments on the numerical constraints, which in the case of Linear Arithmetic over Reals ( $\mathcal{LRA}$ ) is a convex polytope.

## 3 Weighted Model Integration, Revisited

We present a reformulation of WMI which fixes some theoretical limitations of the original formulation and we provide a novel algorithm which drastically improves the performance. A drawback of the original formulation is that the WMI of the same weight function may give different results on two different  $\mathcal{LRA}$ -equivalent formulas. For instance, consider:

$$w((0 \leq x)) = w((x \leq 3)) = 1$$

$$w((x \leq 1)) = w(\neg(x \leq 1)) = \frac{1}{2}$$

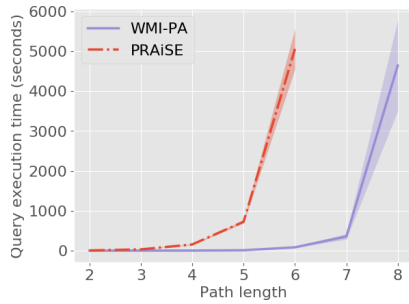
$$\varphi_1 = (0 \leq x) \wedge (x \leq 3)$$

$$\varphi_2 = ((0 \leq x) \wedge (x \leq 1)) \vee (\neg(x \leq 1) \wedge (x \leq 3))$$

Then  $\text{WMI}(\varphi_1, w) = 3 \neq \text{WMI}(\varphi_2, w) = \frac{3}{2}$ . A major source of inefficiency in the original method is that it requires to find (and integrate over) total truth assignments, although in many cases this is not necessary. As an intuitive example, consider two models having the same associated weight and differing only on the truth assignment of a single theory atom (e.g. such atom has constant weight 1 regardless of its truth value). In the original method, this would require to compute two integrals with the same integrand and sum up the results.

Instead of formulating the weight of a model as the product of the weights associated with its literals, we define the notion of *conditional weight function*. In this formulation, the weight function is described by:

Figure 1: Query execution times of WMI-PA and PRAISE for increasing path length.



- a **support** formula  $\chi$ , which defines the subspace of the assignments in which the function is defined (strictly positive),
- a set  $\Psi$  of formulas called **conditions**

such that each (partial) assignment to the atoms of  $\chi$  and  $\Psi$  intuitively represents a branch of a weight function defined by means of nested if-then-elses.

Our algorithm exploits state-of-the-art SMT-based Predicate Abstraction techniques [Graf and Saïdi, 1997] together with this refined formulation of weight function to find and integrate over the partial truth assignments which are relevant in the query computation. The advantage is twofold: first, finding the truth assignments is significantly faster compared to the original method. Second, the number of integrals to be computed is minimal wrt the query.

This approach was applied to the Strategic Road Network dataset<sup>1</sup>, which contains traffic data on the motorways managed by the English Highways Agency. From this dataset we extrapolated polynomial distributions of journey times between all junctions, with a 15 minutes granularity. In this setting, the task is to perform queries of the form:

$$P(t_{arrival} \leq k_1 | path \wedge (t_{departure} = k_2))$$

To the best of our knowledge, the only other system that can perform exact probabilistic inference in this hybrid setting is PRAISE, which implements a method called Probabilistic Inference Modulo Theories [de Salvo Braz et al., 2016]. We compared the execution times of our implementation (WMI-PA) with PRAISE for increasingly longer paths. In this setting, our method is significantly more efficient, as shown in Figure 1.

#### 4 Future work

Future plans include exploiting decomposition and caching techniques to further improve scalability by partitioning the problem in independently-integrable components, as it was done for WMC [Sang et al., 2004; Bacchus et al., 2009]. This could be implemented by building the dependency graph of variables, in which two variables are connected by an edge if they appear in the same atom. Finding a partitioning would require to find the connected components on such graph. One of the main issues here is that in many scenarios, unlike in the

purely propositional case, numerical variables tend to be coupled together by appearing in the same theory atoms. Nevertheless, implementing such techniques would allow to effectively broaden the set of problems that can be solved by our method. Another direction is to leverage the exact approach as a basis to perform approximate inference in Hybrid Domains, as it was done for the original WMI formulation [Belle et al., 2015b].

#### References

[Bacchus et al., 2009] F. Bacchus, S. Dalmao, and T. Pitassi. Solving #SAT and Bayesian inference with backtracking search. *Journal of Artificial Intelligence Research*, 34(1):391–442, 2009.

[Barrett et al., 2009] Clark W Barrett, Roberto Sebastiani, Sanjit A Seshia, and Cesare Tinelli. Satisfiability modulo theories. *Handbook of satisfiability*, 185:825–885, 2009.

[Belle et al., 2015a] V. Belle, A. Passerini, and G. Van den Broeck. Probabilistic inference in hybrid domains by weighted model integration. In *Proc. IJCAI*, 2015.

[Belle et al., 2015b] Vaishak Belle, Guy Van den Broeck, and Andrea Passerini. Hashing-based approximate probabilistic inference in hybrid domains. In *UAI*, pages 141–150, 2015.

[Chavira and Darwiche, 2008] Mark Chavira and Adnan Darwiche. On probabilistic inference by weighted model counting. *Artificial Intelligence*, 172(6-7):772–799, April 2008.

[Choi et al., 2013] Arthur Choi, Doga Kisa, and Adnan Darwiche. Compiling probabilistic graphical models using sentential decision diagrams. In *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pages 121–132. Springer, 2013.

[de Salvo Braz et al., 2016] R. de Salvo Braz, C. O’Reilly, V. Gogate, and R. Dechter. Probabilistic Inference Modulo Theories. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, New York, USA, 2016.

[Fierens et al., 2013] 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. *Theory and Practice of Logic Programming*, 2013.

[Graf and Saïdi, 1997] Susanne Graf and Hassen Saïdi. Construction of abstract state graphs with pvs. In *Proceedings of the 9th International Conference on Computer Aided Verification, CAV ’97*, pages 72–83, London, UK, UK, 1997. Springer-Verlag.

[Sang et al., 2004] Tian Sang, Fahiem Bacchus, Paul Beame, Henry A. Kautz, and Toniann Pitassi. Combining component caching and clause learning for effective model counting. In *SAT*, 2004.

[Suciu et al., 2011] Dan Suciu, Dan Olteanu, Christopher Ré, and Christoph Koch. Probabilistic databases. *Synthesis Lectures on Data Management*, 3(2):1–180, 2011.

<sup>1</sup><https://data.gov.uk/dataset/dft-eng-srn-routes-journey-times>