Lifted Inference for Probabilistic Programming

Wannes Meert Guy Van den Broeck Nima Taghipour Daan Fierens
Hendrik Blockeel Jesse Davis Luc De Raedt
Department of Computer Science
KU Leuven
Celestijnenlaan 200A, B-3001 Heverlee, Belgium
`firstname.lastname@cs.kuleuven.be`

Abstract

A probabilistic program often gives rise to a complicated underlying probabilistic model. Performing inference in such a model is challenging. One solution to this problem is lifted inference which improves tractability by exploiting symmetries in the underlying model. Our group is pursuing a lifted approach to inference for probabilistic logic programs.

1 Introduction

The goal of probabilistic programming is to develop languages that facilitate specifying probabilistic models. As these programs often provide increased expressivity, it allows them to compactly describe many different models. Thus, even a simple, high-level program may give rise to a very complicated underlying model where it is challenging to perform inference. One solution to this problem is lifted inference, which improves the tractability of inference by exploiting the fact that the program often imposes many symmetries in the underlying model. Intuitively, lifting employs two central techniques to speed up inference: (1) divide the problem into isomorphic and independent subproblems, solve one instance, and aggregate the result, and (2) count the number of isomorphic configurations for a group of interchangeable objects instead of enumerating all possible configurations. Many different lifted inference algorithms have been proposed [5, 7, 9, 10, 11, 13, 16] and they have been shown to dramatically improve the run time performance of inference.

Our group is pursuing an approach to lifted inference called weighted first-order model counting (WFOMC) [16] which is based on the insight that probabilistic inference can be reframed as a weighted model counting problem [1]. At a high-level, the approach works by *compiling* a program into a target circuit language where certain inferences can be performed efficiently. Specifically, we introduce and use first-order deterministic decomposable negation normal form (FO d-DNNF) circuits. This circuit allows weighted model counting to be done in time polynomial in the number of objects in the world. The compilation process requires a weighted first-order theory as input and returns a FO d-DNNF. Fortunately, it is possible to transform programs written in many existing probabilistic logic programming languages, such as ProbLog [4], ProbLog2.0 [6] and PRISM [12], and statistical relational representations, such as parfactors [11] and Markov Logic [13], into an equivalent weighted first-order theory. This paper details the basic framework for our inference engine and provides an overview of the various research directions we are pursuing to improve it.

2 Lifted Inference Using Weighted Model Counting

We now describe our approach which is based on weighted model counting and knowledge compilation for exact inference.

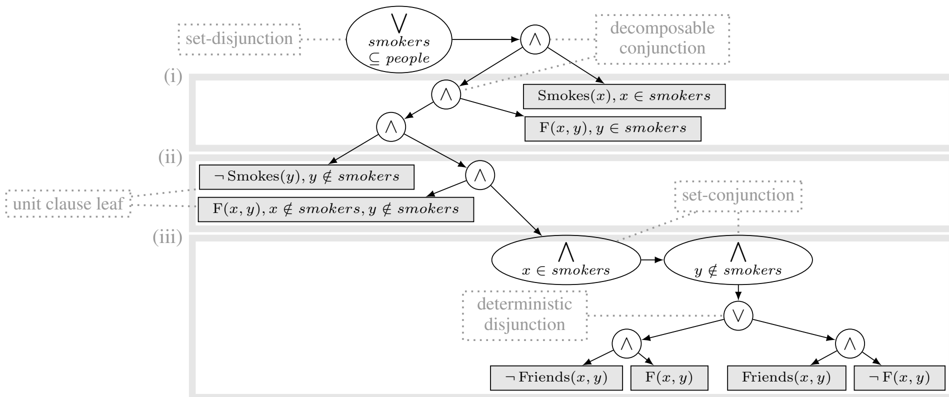

Figure 1: First-Order d-DNNF Circuit for the Formula of Example 1

2.1 WFOMC Representation

A WFOMC problem is a theory in first-order logic and is similar to a Markov logic network (MLN). The difference is that in a WFOMC problem weights can only be associated with predicates. For example, for the predicate Q, only weighted formulas of the form $(w, \mathrm{Q}(x_1, \ldots, x_n))$ are allowed. Complex formulas (containing logical connectives) have to be *hard formulas* with infinite weight. Any MLN can be transformed into a WFOMC problem by adding a new atom to the theory to represent the (truth) value of each weighted complex formula.

Example 1. Consider the following MLN:

$$w \quad \mathrm{Smokes}(x) \wedge \mathrm{Friends}(x, y) \Rightarrow \mathrm{Smokes}(y).$$

In first-order logic, this formula is a hard constraints stating that smokers are only friends with other smokers. Associating a weight w with this formula makes it a soft constraint and it means that smokers are more likely to be friends with other smokers.

The WFOMC representation of the weighted complex formula is

$$\begin{array}{ll} w & \mathrm{F}(x, y) \\ \infty & \mathrm{F}(x, y) \equiv [\mathrm{Smokes}(x) \wedge \mathrm{Friends}(x, y) \Rightarrow \mathrm{Smokes}(y)]\,. \end{array}$$

where we introduce the additional atom $\mathrm{F}(x, y)$ to carry the weight of the MLN formula.

2.2 First-Order Knowledge Compilation and Inference

First-order knowledge compilation compiles a first-order knowledge base into a target circuit language. We use FO d-DNNF as the target circuit language, which represents theories in first-order logic with domain constraints. *Domain constraints* define a finite domain for each logical variable.

A FO d-DNNF circuit is a directed, acyclic graph, where the leaves represent first-order literals and the inner nodes represent formulas. A FO d-DNNF includes the following inner node types: *decomposable conjunction*, a conjunction of children that do not share any random variables, *deterministic disjunction*, a disjunction whose children cannot be true at the same time, *first-order generalizations* of these types of operators, and *recursive conjunction*.

Example 2. Figure 1 illustrates a FO d-DNNF for the formula of Example 1. The circuit introduces a new domain $Smokers$, which is a subset of all $People$. It states that there exists such a $Smokers$ for which (i) all people in $Smokers$ are smokers (ii) no other people are smokers and (iii) smokers are not friends with non smokers.

WFOMC uses a top-down compilation algorithm for transforming a weighted logical theory into a FO d-DNNF. The algorithm applies a sequence of operations that simplify the logical theory; see Van den Broeck *et al.* [16] for an overview of the algorithm. There is no guarantee that every logical theory can be compiled. However, we proved that any logical theory without existential quantifiers and where each formula contains at most two logical variables (the class of 2-WFOMC models) can be compiled [18] . Still, many models outside this class, including many theories used in practice, can be compiled.

The compiled circuit can answer probabilistic queries for any given set of weights and domain-sizes. The marginal probability of a query q for a model M, weight vector w and domain size D is $\mathrm{P}(q|M) = \mathrm{WMC}(q \wedge M, w, D)/\mathrm{WMC}(M, w, D)$, where WMC stands for the weighted model count. The $\mathrm{WMC(q \wedge M)}$ is simply the weight of all possible worlds where q is true. The $\mathrm{WMC(M)}$ is the partition function Z for the model. Darwiche [3] gives a more detailed overview of the weighted model counting approach to probabilistic inference (for the propositional case). The FO d-DNNF circuit is independent of the domain of the logical variables. Furthermore, computing weighted model counts is polynomial in the size of the domains. One advantage of using knowledge compilation for inference is that it exploits context-specific independence and determinism in the MLN.

3 Ongoing Research Directions

We are pursuing the following research objectives to expand the applicability of WFOMC and investigate the theoretical limitations of lifted inference.

Completeness.[1] There is an ongoing effort to identify the different classes of queries and probabilistic logic models which are provebly liftable. The liftability of a class is formally defined using the concept of domain-lifted probabilistic inference [8, 18]. For all WFOMC theories that can be compiled domain-lifted probabilistic inference is guaranteed.

> **Definition 1** (Domain-Lifted Probabilistic Inference)**.** A probabilistic inference procedure is *domain-lifted* for a model M, query q and evidence e iff the inference procedure runs in *polynomial* time in $|D_1|, \ldots, |D_k|$ with D_i the domain of the logical variable $v_i \in \mathrm{vars}(M, q, e)$.

Approximate inference.[2] Although the method introduced above performs exact inference it can also be used in an approximate inference strategy. Van den Broeck *et al.* [17] introduced a lifted variant of the Relax, Compensate and then Recover (RCR) approximate inference method for (ground) probabilistic graphical models [2]. This method is based upon the idea that the structure of a first-order model can be simplified, or relaxed, until it is amenable to exact lifted inference. By iteratively relaxing and compensating for the simplification we obtain an approximate result.

Conditioning. Initially, FO d-DNNFs did not support conditioning. Thus answering each query required compiling a new circuit. For FO d-DNNFs it is possible to support conditioning on certain types of evidence [15]. Specifically, a single first-order circuit can answer any query about unary atoms. As a result, if the evidence is on unary relations, inference is now polynomial instead of exponential in the size of the evidence set.

Arbitrary constraints. Exact lifted inference techniques rely on expressions of constraints to denote groups of similar objects. The flexibility and of granularity of the grouping is determined by the expressivity of the constraint language, which is often restricted to pairwise (in)equality constraints The inference methods can be generalized to work with arbitrary constraints and this allows them to capture a broader range of symmetries, leading to more opportunities for lifting [14].

4 Conclusions

We have introduced an approach using lifted inference to deal with the increasing complexity of the underlying probabilistic models in probabilistic programming. Lifted inference's main strength is the ability to exploit symmetries in the model. This approach is not limited to exact inference but can be used in an approximate strategy as well. A JVM-executable and the source code is available at `http://dtai.cs.kuleuven.be/wfomc/`.

[1] The research on completeness is in collaboration with Manfred Jaeger.

[2] The research on approximate inference is in collaboration with Adnan Darwiche and Arthur Choi.

References

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

[2] Arthur Choi and Adnan Darwiche. Relax, compensate and then recover. In *New Frontiers in Artificial Intelligence, volume 6797 of LNCS*, pages 167–180. Springer, 2011.

[3] Adnan Darwiche. *Modeling and Reasoning with Bayesian Networks*. Cambridge University Press, 2009.

[4] Luc De Raedt, Angelika Kimmig, and Hannu Toivonen. ProbLog: A probabilistic Prolog and its application in link discovery. In *Proceedings of IJCAI*, pages 2468–2473, 2007.

[5] Rodrigo de Salvo Braz, Eyal Amir, and Dan Roth. Lifted first-order probabilistic inference. In *Proceedings of IJCAI*, pages 1319–1325, 2005.

[6] Daan Fierens, Guy Van den Broeck, Ingo Thon, Bernd Gutmann, and Luc De Raedt. Inference in probabilistic logic programs using weighted CNF's. In *Proceedings of UAI*, pages 211–220, 2011.

[7] Vinhav Gogate and Pedro Domingos. Probabilistic theorem proving. In *Proceedings of UAI*, 2011.

[8] Manfred Jaeger and Guy Van den Broeck. Liftability of probabilistic inference: Upper and lower bounds. In *Proceedings of StarAI*, 2012.

[9] Kristian Kersting, Babak Ahmadi, and Sriraam Natarajan. Counting belief propagation. In *Proceedings of UAI*, pages 277–284, 2009.

[10] Brian Milch, Luke S. Zettlemoyer, Kristian Kersting, Michael Haimes, and Leslie Pack Kaelbling. Lifted probabilistic inference with counting formulas. In *Proceedings of AAAI*, pages 1062–1068, 2008.

[11] David Poole. First-order probabilistic inference. In *Proceedings of IJCAI*, pages 985–991, 2003.

[12] Taisuke Sato. A statistical learning method for logic programs with distribution semantics. In *Proceedings of ICLP*, pages 715–729, 1995.

[13] Parag Singla and Pedro Domingos. Lifted first-order belief propagation. In *Proceedings of AAAI*, pages 1094–1099, 2008.

[14] Nima Taghipour, Daan Fierens, Jesse Davis, and Hendrik Blockeel. Lifted variable elimination with arbitrary constraints. In *Proceedings of AISTATS*, pages 1194–1202, 2012.

[15] Guy Van den Broeck and Jesse Davis. Conditioning in first-order knowledge compilation and lifted probabilistic inference. In *Proceedings of AAAI*, 2012.

[16] Guy Van den Broeck, Nima Taghipour, Wannes Meert, Jesse Davis, and Luc De Raedt. Lifted probabilistic inference by first-order knowledge compilation. In *Proceedings of IJCAI*, pages 2178–2185, 2011.

[17] Guy Van den Broeck, Arthur Choi, and Adnan Darwiche. Lifted relax, compensate and then recover: From approximate to exact lifted probabilistic inference. In *Proceedings of UAI*, 2012.

[18] Guy Van den Broeck. On the completeness of first-order knowledge compilation for lifted probabilistic inference. In *Proceedings of NIPS*, 2011.