Liftability of Probabilistic Inference: Upper and Lower Bounds

Manfred Jaeger
Aalborg University
jaeger@cs.aau.dk

Guy Van den Broeck
KU Leuven
guy.vandenbroeck@cs.kuleuven.be

Abstract

We introduce a general framework for defining classes of probabilistic-logic models and associated classes of inference problems. Within this framework we investigate the complexity of inference in terms of the size of logical variable domains, query and evidence, corresponding to different notions of liftability. Surveying existing and introducing new results, we present an initial complexity map for lifted inference. Main results are that lifted inference is infeasible for general quantifier-free first-order probabilistic knowledge bases, but becomes tractable when formulas are restricted to the 2-variable fragment of quantifier-free first-order logic.

1 Introduction

Probabilistic logic models (a.k.a. probabilistic or statistic relational models) provide high-level representation languages for probabilistic models of structured data [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]. While supporting model specifications at an abstract, first-order logic level, inference is typically performed at the level of concrete ground instances of the models, i.e., at the propositional level. This mismatch between model specification and inference methods has been noted early on, and has given rise to numerous proposals for inference techniques that operate at the high level of the first-order model specifications [12, 13, 14, 15, 16, 17, 18, 19]. Inference methods of this nature have collectively become known as "lifted" inference techniques.

The concept of lifted inference is mostly introduced on an informal level: "*...lifted, that is, deals with groups of random variables at a first-order level*" [13]; "*The act of exploiting the high level structure in relational models is called lifted inference*" [20]; "*The idea behind lifted inference is to carry out as much inference as possible without propositionalizing* [15]; "*lifted inference, which deals with groups of indistinguishable variables, rather than individual ground atoms* [21]. While, thus, the term lifted inference emerges as a quite coherent algorithmic metaphor, it is not immediately obvious what its exact technical meaning should be. Since quite a variety of different algorithmic approaches are collected under the label "lifted", and since most of them can degenerate for certain models to ground, or propositional, inference, it is difficult to precisely define the class of lifted inference techniques in algorithmic terms.

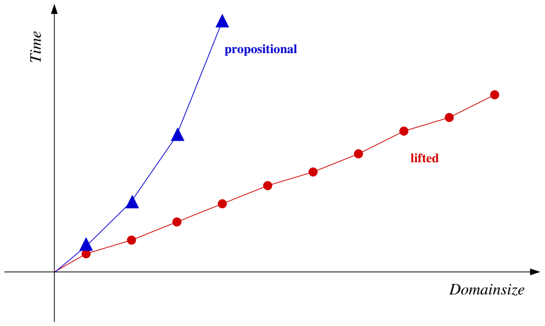

Figure 1: A typical performance evaluation

A more fruitful approach is to formalize the concept of lifted inference in terms of its objectives, rather than in terms of its algorithmic means. Here one observes that lifted inference techniques very consistently are evaluated on, and compared against each other, by how well inference complexity scales as a function of the domain (or population) for which the general model is instantiated. Thus, empirical evaluations of lifted inference techniques are usually presented in the form of domain size vs. inference time plots as shown in Figure 1.

Van den Broeck [19], therefore, has proposed a formal definition of *domain-lifted inference* in terms of polynomial time complexity in the domainsize parameter. Experimental and theoretical analyses of existing lifted inference techniques then show that they provide domain-lifted inference in some cases where basic propositional inference techniques would exhibit exponential complexity. However, until recently, these positive results were mostly limited to examples of individual models, and little was known about the feasibility of lifted inference for certain well-defined classes of models. The first result that shows the feasibility of lifted inference for a class of models is given by Van den Broeck [19].

On the other hand, Jaeger [22] has shown that under certain assumptions on the expressivity of the modeling language, probabilistic inference is not polynomial in the domain size, thereby demonstrating some inherent limitations in terms of worst-case complexity for the goals of lifted inference. The results of Jaeger [22] essentially assume a modeling framework based on directed probabilistic models, and the expressivity requirements amount to a probabilistic version of full first-order predicate logic. Since much recent work in lifted inference is based on undirected probabilistic models, and, more significantly, within these frameworks focuses on fragments without full first-order expressivity, it is not clear to what extent these earlier intractability results are applicable to these ongoing efforts.

In this paper we provide a detailed study of the feasibility of lifted inference. We introduce a general, formal framework for analyzing the complexity of inference from probabilistic-logic models in a weighted model counting framework (Section 2). Extending the general approach taken in Jaeger [22], we then derive new lower complexity bounds that show that domain-lifted inference still is infeasible when the strong assumptions of Jaeger [22] are loosened, and only the expressivity of quantifier-free first-order logic without equality is required (Section 3). The following sections give an overview of existing work. Our new negative result is contrasted with the recent positive liftability results (Section 4) for the 2-variable fragment of probabilistic logic. We then move to a more detailed analysis of the role of evidence in liftability (Section 5). Finally, we give some challenges for future work (Section 6).

2 Background and Notation

We will now introduce the necessary background on probabilistic-logic models and lifted inference.

2.1 Weighted Feature Models

Similarly as Richardson and Domingos [11], Van den Broeck et al. [17], Gogate and Domingos [18] we assume the following framework: a model, or knowledge base, is given by a set of weighted formulas:

$$KB: \quad \begin{array}{lcl} \phi_1(\boldsymbol{v}_1) & : & w_1 \\ \phi_2(\boldsymbol{v}_2) & : & w_2 \\ \cdots & & \cdots \\ \phi_N(\boldsymbol{v}_N) & : & w_N \end{array} \tag{1}$$

where the ϕ_i are formulas in first-order predicate logic, $w_i \in \mathbb{R}$ are non-negative *weights*, and $\boldsymbol{v}_i = (v_{i,1}, \ldots, v_{i,k_i})$ are the free variables of ϕ_i. The case $k_i = 0$, i.e., ϕ_i is a sentence without free variables, is also permitted. The ϕ_i use a given signature S of relation-, function-, and constant symbols.

An interpretation (or possible world) (D, I) for S consists of a domain D, and an interpretation function I that maps the symbols in S to functions, relations and elements on D. For a tuple $\boldsymbol{d} \in D^{k_i}$ then the truth value $\phi_i(\boldsymbol{d}/\boldsymbol{v}_i)$ is defined, and we write $(D, I) \models \phi_i(\boldsymbol{d})$, or simpler $I \models \phi_i(\boldsymbol{d})$ if $\phi_i(\boldsymbol{d}/\boldsymbol{v}_i)$ is true in (D, I). We use $\mathcal{I}(D, S)$ to denote the set of all interpretations for the signature S over the domain D.

In this paper we are only concerned with finite domains, and assume without loss of generality that $D = D_n := \{1, \ldots, n\}$ for some $n \in \mathbb{N}$.

For $I \in \mathcal{I}(D_n, S)$ let $\#(i, I)$ denote the number of elements $\boldsymbol{d}$ in D^{k_i} for which $I \models \phi_i(\boldsymbol{d})$. The *weight* of I then is

$$w_n^{KB}(I) := \prod_{i=1}^{N} w_i^{\#(i,I)}, \tag{2}$$

where $0^0 = 1$. The probability of I is

$$P_n^{KB}(I) = w_n^{KB}(I)/Z$$

where Z is the normalizing constant (partition function)

$$Z = \sum_{I' \in \mathcal{I}(D_n, S)} w_n^{KB}(I').$$

For a first-order sentence ϕ and $n \in \mathbb{N}$ then

$$P_n^{KB}(\phi) := P(\{I \in \mathcal{I}(D_n, S) \mid I \models \phi\}) \tag{3}$$

is the probability of ϕ in $\mathcal{I}(D_n, S)$.

We call a knowledge base (1) together with the semantics given by (2) and (3) a *weighted feature model*, since it associates weights w_i with model features ϕ_i. Weighted feature models in our sense can be seen as a slight generalization of *weighted model counting* frameworks [17, 23, 18] in which non-zero weights are only associated with literals. Knowledge bases of the form (1) can be translated into weighted model counting frameworks via an introduction of new relation symbols $R_1, \ldots, R_N$, hard constraints $\phi_i(\boldsymbol{v}_i) \leftrightarrow R_i(\boldsymbol{v}_i)$, and weighted formulas $R_i(\boldsymbol{v}_i) : w_i$ [17, 18]. Up to an expansion of the signature, thus, weighted feature models and weighted model counting are equally expressive. *Markov Logic Networks* [11] also are based on knowledge bases of the form (1) with arbitrary formulas ϕ_i allowed. However, the semantics of the model there depends on a transformation of the formulas into conjunctive normal form, and therefore does not exactly correspond to (2) and (3), unless the ϕ_i are clauses. Finally, *Parfactor* models [12, 13], even though they use a syntactically different style of representation, also can be transformed into a knowledge base (1), and vice-versa. Complexity results developed on the basis of weighted feature models, thus, are applicable to a wide range of probabilistic-logic models.

An *probabilistic inference problem* $\mathrm{PI}(KB, n, \phi, \psi)$ for a weighted feature model is given by a knowledge base KB, a domainsize $n \in \mathbb{N}$, and two first-order sentences ϕ, ψ. The solution to the inference problem is the conditional probability $P_n^{KB}(\phi \mid \psi) = P_n^{KB}(\phi \wedge \psi)/P_n^{KB}(\psi)$.

2.2 Liftability

A *class* of inference problems is defined by allowing arguments KB, ϕ, and ψ only from some restricted classes $\mathcal{KB}$, $\mathcal{Q}$ (the query class), and $\mathcal{E}$ (the evidence class), respectively. We use the notation

$$\begin{aligned}&\mathrm{PI}(\mathcal{KB}, \mathcal{Q}, \mathcal{E}) := \\ &\{\mathrm{PI}(KB, n, \phi, \psi) \mid KB \in \mathcal{KB}, n \in \mathbb{N}, \phi \in \mathcal{Q}, \psi \in \mathcal{E}\}\end{aligned}$$

for classes of inference problems.

For queries $\mathcal{Q}$ and evidence $\mathcal{E}$ we will use classes

- $\mathcal{A}$ for single ground atoms,
- $\mathcal{T}$ for terms (conjunctions) of ground literals,
- $\mathcal{T}_{0,1}$ for terms of ground literals with arity 0 or 1,
- $\emptyset$ for empty sets of evidence.

For example, $\mathcal{Q} = \mathcal{A}$ and $\mathcal{E} = \emptyset$ denotes queries for single marginals, without evidence.

Classes $\mathcal{KB}$ are defined by various syntactic restrictions on the formulas ϕ_i in the knowledge base. The fragments of first-order logic (`FOL`) we consider are

- `FFFOL` for function-free `FOL`, i.e., `FOL` without function symbols, and
- `RFOL` for relational `FOL`, i.e., `FFFOL` without constant symbols.

We further distinguish between classes that allow for quantifiers ($\forall\exists$) and the equality predicate ($=$). For example, `FOL`($\forall\exists, =$) is `FOL` with quantifiers and equality and `FOL`($=$) is `FOL` without quantifiers but with equality. A last type of subclass limits the number of logical variables per formula. For example, k-`FOL` is `FOL` with k logical variables per formula.

An algorithm solves a class $\mathrm{PI}(\mathcal{KB}, \mathcal{Q}, \mathcal{E})$, if it computes $P_n^{KB}(\phi \mid \psi)$ for all instances $\mathrm{PI}(KB, n, \phi, \psi)$ in the class.

Van den Broeck [19] proposed one notion of *lifted* inference for first-order probabilistic models, defined in terms of the computational complexity of inference w.r.t. the domains of logical variables.

Definition 2.1 (Domain-Lifted) *An algorithm is domain-lifted, iff for fixed* KB*, ϕ and ψ the computation of $P_n^{\mathrm{KB}}(\phi \mid \psi)$ is polynomial in n.*

Domain-lifted inference does not prohibit the algorithm to be *exponential* in the size of the vocabulary, that is, the number of predicates, arguments and constants in KB, ϕ and ψ. The definition was motivated by the observation that first-order theories are often concise but the presence of large domains causes inference to become intractable, when done at the propositional level.

Definition 2.2 (Completeness) *An algorithm is called a complete domain-lifted inference algorithm for the class* $\mathrm{PI}(\mathcal{KB}, \mathcal{Q}, \mathcal{E})$ *iff it is domain-lifted and solves this class.*

The related notion of liftability talks about the existence of a complete domain-lifted algorithm.

Definition 2.3 (Liftability) *A class* $\mathrm{PI}(\mathcal{KB}, \mathcal{Q}, \mathcal{E})$ *is liftable iff there exists an algorithm that is complete domain-lifted for this class.*

The above definition of liftability pertains only to exact inference. An analogous notion is *approximate liftability*, when there exists an algorithm that ϵ-approximately solves all problems in a class and is domain-lifted. An algorithm ϵ-approximately solves $\mathrm{PI}(\mathcal{KB}, \mathcal{Q}, \mathcal{E})$, if for any $\mathrm{PI}(KB, n, \phi, \psi)$ in the class it returns a number in $[P_n^{KB}(\phi \mid \psi) - \epsilon, P_n^{KB}(\phi \mid \psi) + \epsilon]$.

The following sections will investigate the liftability of different classes of $\mathrm{PI}(\mathcal{KB}, \mathcal{Q}, \mathcal{E})$. The results are summarized in Table 1. Sections 3 and 4 will deal with negative and positive results for classes where $\mathcal{Q}$ and $\mathcal{E}$ are bounded in size, for example the case where we want

$\mathcal{KB}$	$\mathcal{Q}$	$\mathcal{E}$	**Exact Liftable**	**Approx. Liftable**	**Notion**	**Reference**	**Theorem**
$\texttt{RFOL}(\forall\exists,=)$	$\mathcal{A}$	$\emptyset$	✗	✗	domain-lifted	[22]	3.4
$\texttt{RFOL}$	$\mathcal{A}$	$\emptyset$	✗*	✗*	domain-lifted	(new result)	3.5
k-$\texttt{RFOL}$	$\mathcal{A}$	$\emptyset$	?	?	domain-lifted		
2-$\texttt{FFFOL}(=)$	$\mathcal{T}$	$\mathcal{T}$	✓	✓	domain-lifted	[19]	4.2
2-$\texttt{RFOL}$	$\mathcal{A}$	$\mathcal{T}$	✗	?	dqe-lifted	[24]	5.6
2-$\texttt{FFFOL}(=)$	$\mathcal{T}_{0,1}$	$\mathcal{T}_{0,1}$	✓	✓	dqe-lifted	[24]	5.8

Table 1: Liftability results w.r.t. classes of knowledge bases $\mathcal{KB}$, queries $\mathcal{Q}$ and evidence $\mathcal{E}$ (contingent on complexity assumptions). For negative/positive results, the most specific/general class is shown. ✗*: subject to additional condition on polynomial time complexity as function of parameter complexity.

to compute single marginals without evidence. Section 5 then extends the discussion to liftability which takes into account the size of the query and evidence.

3 Lower Complexity Bounds

All complexity results in this section are obtained by reducing the *spectrum recognition problem* to probabilistic inference problems. Full proofs for this section can be found in Jaeger [25]. We first briefly review the fundamental concepts about spectra of first-order logic sentences, and the complexity class ETIME.

Definition 3.1 *Let ψ be a sentence in first-order logic. The* spectrum *of ψ is the set of integers $n \in \mathbb{N}$ for which ψ is satisfiable by an interpretation of size n.*

Example 3.2 *Let $\psi = \psi_1 \wedge \psi_2 \wedge \psi_3$, where*

$$\begin{array}{lcl} \psi_1 & \equiv & \forall x, y \;\; u(x,y) \Leftrightarrow u(y,x) \\ \psi_2 & \equiv & \forall x \; \exists y \;\; y \neq x \;\wedge\; u(x,y) \\ \psi_3 & \equiv & \forall x, y, y' \;\; (u(x,y) \wedge u(x,y') \Rightarrow y = y') \end{array}$$

ψ expresses that the binary relation u defines an undirected graph (ψ_1) in which every node is connected to exactly one other node (ψ_2, ψ_3). Thus, ψ describes a pairing *relation that is satisfiable exactly over domains of even size:* $\mathrm{spec}(\psi) = \{n \mid n$ *even*$\}$.

The complexity class ETIME consists of problems solvable in time $O(2^{cn})$, for some constant c. The corresponding nondeterministic class is NETIME. Based on a characterization of NETIME in terms of first-order spectra given by Jones and Selman [26], we obtain:

Proposition 3.3 *If NETIME $\neq$ ETIME, then there exists a first-order sentence ϕ, such that $\{n \mid n \in$* $\mathrm{spec}(\phi)\}$ *can not be recognized by a deterministic algorithm in time polynomial in n.*

Thus, by reducing the spectra-recognition problem to a class of inference problems $\mathrm{PI}(\mathcal{KB}, \mathcal{Q}, \mathcal{E})$, one establishes that the latter is not polynomial in the domainsize (under the assumption ETIME $\neq$ NETIME).

We first state a result for knowledge bases using $\texttt{RFOL}(\forall\exists,=)$. This is rather straightforward, and (for exact inference) already implied by the results of [22].

Theorem 3.4 *If NETIME $\neq$ ETIME, then there does not exist an algorithm that 0.25-approximately solves* $\mathrm{PI}(\texttt{RFOL}(\forall\exists,=), \mathcal{A}, \emptyset)$ *in time polynomial in the domainsize.*

The following theorem extends Theorem 3.4 to knowledge bases with only quantifier-free formulas without equality. At the same time a slight weakening is introduced by imposing an additional condition on the complexity in terms of representation size of the weight parameters in *KB*.

Theorem 3.5 *If NETIME $\neq$ ETIME, then there does not exist an algorithm that 0.25-approximately solves* $\mathrm{PI}(\texttt{RFOL}, \mathcal{A}, \emptyset)$ *in time polynomial in n and the representation size $l := \sum_{i=1}^{N} \log(w_i)$ of the weight parameters in the* KB *(1).*

The basic strategy for proving this theorem is to replace quantifiers with relational encodings of Skolem functions, and the equality predicate $=$ with an ordinary binary relation $E(\cdot,\cdot)$. The problem, then, is to enforce that the newly introduced relations behave like functions, respectively like the equality relation. Given the expressive power only of $\texttt{RFOL}$, this, naturally, is not possible to do exactly. However, using suitable weighted formulas, one can ensure that interpretations in which the Skolem relations and the E relation do not show the desired properties have a negligible weight. The weights for the formulas constraining the Skolem and E relations, now, have to be calibrated as a function of the domainsize n. Thus, for a given first-order formula ψ, the problem $n \in \mathit{spec}(\psi)$ is reduced to an inference problem $\mathrm{PI}(\mathit{KB}(n), n, \phi, \emptyset)$, where for different n the $\mathit{KB}(n)$ only differ with respect to the values of the weights w_i.

4 Positive Liftability Result

Van den Broeck [19] showed that one particular algorithm for lifted probabilistic inference is complete for the class 2-$\texttt{FFFOL}(=)$ of weighted model counting problems without quantifiers, with equality and two logical variables per formula. The algorithm is called *first-order knowledge compilation* and the proof of its completeness also is a constructive proof that this class of models is liftable. For the first time, one could show that domain-lifted inference is always possible for a non-trivial class of models, based on its syntactic properties.

Theorem 4.1 *First-order knowledge compilation is a complete domain-lifted algorithm for the class* $\mathrm{PI}(2\text{-}\texttt{FFFOL}(=), \mathcal{T}, \mathcal{T})$.

Corollary 4.2 *The class* $\mathrm{PI}(2\text{-}\texttt{FFFOL}(=), \mathcal{T}, \mathcal{T})$ *is domain-liftable.*

First-order knowledge compilation compiles a first-order knowledge base into a target circuit language, *FO d-DNNF* [17], which represents formulas in first-order logic with *domain constraints*. Domain constraints are constraints that 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. The inner nodes of a FO d-DNNF have certain properties. For example, conjunctions have to be *decomposable*, meaning that its conjuncts do not share any random variables, and disjunctions have to be *deterministic*, meaning that its disjuncts cannot be true at the same time. Other types of inner nodes are *first-order* generalizations of these concepts.

First-order knowledge compilation can be used to answer probabilistic queries. Given a knowledge base *KB*, query ϕ and evidence ψ, the first-order knowledge compilation approach to lifted inference compiles a FO d-DNNF for the theories $KB \wedge \psi$ and $KB \wedge \psi \wedge \phi$. These two circuits can be used to answer queries $\mathrm{PI}(KB, n, \phi, \psi)$ for any domain size n: the probability $P_n^{KB}(\phi \mid \psi)$ for a weight vector w is

$$P_n^{KB}(\phi \mid \psi) = \frac{\mathrm{WMC}(KB \wedge \psi \wedge \phi, w, n)}{\mathrm{WMC}(KB \wedge \psi, w, n)} \tag{4}$$

where WMC stands for the weighted model count of the logical theory. The $\mathrm{WMC}(KB \wedge \psi \wedge \phi, w, n)$ is the weight of all possible worlds where query and evidence are true. The $\mathrm{WMC}(KB \wedge \psi, w, n)$ is the weight of all possible worlds where the evidence is true. In the case when there is no evidence, this is simply the partition function Z of the model *KB*. Moreover, answering $\mathrm{PI}(KB, n, \phi, \psi)$ this way can be done polynomially in n, because evaluating the weighted model count of a FO d-DNNF is polynomial in n. Darwiche [27] gives a more detailed overview of the weighted model counting approach to probabilistic inference (for the propositional case).

Transforming a logical theory into a FO d-DNNF is done by a top-down compilation algorithm, which applies a sequence of operations that simplify the logical theory. See Van den Broeck et al. [17] for an overview of the compilation algorithm. To prove Theorem 4.1, it is sufficient to show that the compilation algorithm is able to compile formulas $KB \wedge \psi$ and $KB \wedge \psi \wedge \phi$ for any $\mathrm{PI}(2\text{-}\texttt{FFFOL}(=), \mathcal{T}, \mathcal{T})$. Once the circuit is compiled, the properties of FO d-DNNFs allow for the computation of any $\mathrm{WMC}(KB \wedge \psi \wedge \phi, w, n)$ and $\mathrm{WMC}(KB \wedge \psi, w, n)$.

The first step of the proof shows that the compilation algorithm can remove any ground atoms and atoms with a single logical variable argument from the theory to be compiled. Once these are removed, all atoms in a $KB \in 2\text{-}\texttt{FFFOL}(=)$ must have two logical variable arguments. Van den Broeck [19] then goes on to show that the grounding of such a theory consists of disconnected components with identical weighted model counts and that symmetry arguments allow us to always compile such theories.

The class 2-$\texttt{FFFOL}(=)$ is not trivial and surprisingly expressive, containing many models in practical use today. In can contain important concepts such as symmetric, anti-symmetric, total and homophily relations:

$$\begin{aligned} \mathrm{friends}(X, Y) &\Rightarrow \mathrm{friends}(Y, X) \\ \mathrm{parent}(X, Y) &\Rightarrow \neg\, \mathrm{parent}(Y, X) \\ \neg \leq (X, Y) &\Rightarrow \leq (Y, X) \\ \mathrm{smokes}(X) \wedge \mathrm{friends}(X, Y) &\Rightarrow \mathrm{smokes}(Y) \end{aligned}$$

Furthermore, all models currently used in the lifted inference literature are in this class. Still, there are useful models that are not in 2-$\texttt{FFFOL}(=)$, most notably models with existential quantifiers and concepts such as transitivity and generalized homophily:

$$\mathrm{friends}(X, Y) \wedge \mathrm{friends}(Y, Z) \Rightarrow \mathrm{friends}(X, Z) \tag{5}$$

$$\mathrm{likes}(X, Z) \wedge \mathrm{friends}(X, Y) \Rightarrow \mathrm{likes}(Y, Z) \tag{6}$$

which contain three logical variables per formula. For these formulas, the proof of Van den Broeck [19] fails in the second step: their groundings are all strongly connected and they do not decompose. Many models not in 2-$\texttt{FFFOL}(=)$ can still be compiled to FO d-DNNF, but there are no guarantees in the form of a completeness or liftability result.

5 Unbounded Queries and Evidence

We will now turn our attention to a notion of lifted inference where the query and evidence are not bound in size.

Definition 5.1 (DQE-Lifted) *An algorithm is domain-, query- and evidence-lifted (dqe-lifted), iff for fixed* KB *the computation of $P_n^{\mathrm{KB}}(\phi \mid \psi)$ is polynomial in n and the size of ϕ and ψ.*

This definition is motivated by the fact that domain-lifted inference algorithms tend to perform poorly in the presence of evidence, which breaks the symmetries in the first-order model. When sufficient amounts of evidence are present, lifted inference algorithms will start to behave identically to their propositional counterparts, that is, first-order variable elimination [12] will perform variable elimination [28], first-order knowledge compilation to FO d-DNNF [17] will perform propositional knowledge compilation to d-DNNF [27], etc.

In this section, we will talk about general terms ($\mathcal{T}$) and terms with literals of arity zero and one specifically ($\mathcal{T}_{0,1}$) and investigate notions of completeness and liftability for dqe-lifted inference.

5.1 Negative Results

den Broeck and Davis [24] prove a negative result, that computing conditional probabilities is #P-hard, by showing that #2SAT is reducible to it. A k-CNF formula is a CNF with k literals per clause. kSAT is the problem of deciding the satisfiability of a k-CNF formula. The #kSAT problem involves counting the number of satisfying assignments to a k-CNF formula. This task is also called model counting.

Example 5.2 *The following formula is in 2-CNF:*

$$(a \vee b) \wedge (a \vee \neg c) \wedge (\neg c \vee \neg d)$$

2SAT is decidable in polynomial time. However, #2SAT is #P-complete [29], which implies that it is not solvable in polynomial time unless $P = NP$.

Lemma 5.3 *Each #2SAT problem can be solved by querying a knowledge base* KB *that represents a uniform distribution over the models of the logical theory*

$$\begin{aligned} &\neg q \vee \quad \mathrm{p}(X) \vee \quad \mathrm{p}(Y) \vee \neg\, \mathrm{c}_1(X,Y) \\ &\neg q \vee \quad \mathrm{p}(X) \vee \neg\, \mathrm{p}(Y) \vee \neg\, \mathrm{c}_2(X,Y) \\ &\neg q \vee \neg\, \mathrm{p}(X) \vee \neg\, \mathrm{p}(Y) \vee \neg\, \mathrm{c}_3(X,Y), \end{aligned} \tag{7}$$

The $\mathrm{p}(X)$ atoms represent propositions in the 2-CNF, uniquely identified by the constants in the domain of the variable X. The $\mathrm{c}_i(X,Y)$ atoms encode which propositions appear together in the three possible types of clauses for 2-CNF. Adding a positive c_i literal to the evidence ψ includes the clause of type i for the given propositions in the 2-CNF. For example, conditioning the query on evidence $\mathrm{c}_1(a,b)$ adds $\mathrm{p}(a) \vee \mathrm{p}(b)$ to the theory. Conditioning on a negative c-literal omits the clause for the given propositions from the theory. For example, conditioning on $\neg\, \mathrm{c}_2(a,b)$ excludes $\mathrm{p}(a) \vee \neg\, \mathrm{p}(b)$ from the theory.

Theorem 5.4 *For any $\mathcal{KB}$ that includes a uniform distribution over the models of Theory 7, solving $\mathrm{PI}(\mathcal{KB}, \mathcal{A}, \mathcal{T})$ is #P-hard.*

Querying for $\mathrm{P}(\mathrm{q} \mid \psi)$, where ψ assigns a truth value to every c_i atom, returns $c/(c+2^n)$, where c is the model count of the 2-CNF represented by ψ and 2^n the number of possible assignments to the n propositions in the 2-CNF. This allows us to solve arbitrary #2SAT problems by solving for c.

Example 5.5 *Computing $P_n^{\mathrm{KB}}(q \mid \psi)$ where* KB *is Theory 7, n is four (the number of propositions in Example 5.2) and ψ is*

$$\begin{aligned} &\mathrm{c}_1(a,b) \wedge \neg\, \mathrm{c}_1(a,a) \wedge \cdots \wedge \neg\, \mathrm{c}_1(d,d) \wedge \\ &\mathrm{c}_2(a,c) \wedge \neg\, \mathrm{c}_2(a,a) \wedge \cdots \wedge \neg\, \mathrm{c}_2(d,d) \wedge \\ &\mathrm{c}_3(c,d) \wedge \neg\, \mathrm{c}_3(a,a) \wedge \cdots \wedge \neg\, \mathrm{c}_3(d,d) \end{aligned}$$

returns $c/(c+2^n)$, where c is the model count of

$$(\mathrm{p}(a) \vee \mathrm{p}(b)) \wedge (\mathrm{p}(a) \vee \neg\, \mathrm{p}(c)) \wedge (\neg\, \mathrm{p}(c) \vee \neg\, \mathrm{p}(d)),$$

which is isomorphic to the 2-CNF of Example 5.2.

Because Theory 7 is in 2-`RFOL`, we have the following.

Corollary 5.6 $\mathrm{PI}(2\text{-}\mathtt{RFOL}, \mathcal{A}, \mathcal{T})$ *is not (dqe-)liftable, unless P=NP.*

The result of den Broeck and Davis [24] highlights an important limitation of all exact lifted inference methods: computing probabilities with evidence on binary relations cannot be polynomial in the size of the evidence, unless P=NP. This result identifies a sharp contrast between lifted and propositional inference algorithms: whereas (partial) evidence in undirected models simplifies inference in the propositional case, it can complicate inference in the lifted case.

5.2 Positive Results

For a weaker class of queries and evidence, den Broeck and Davis [24] showed a positive result, namely that inference can be polynomial in the domain size and the size of the query and evidence when these contain only literals of arity zero and one.

Theorem 5.7 *First-order knowledge compilation is a complete dqe-lifted algorithm for the class* PI(2-$\texttt{FFFOL}(=), \mathcal{T}_{0,1}, \mathcal{T}_{0,1}$).

Corollary 5.8 PI(2-$\texttt{FFFOL}(=), \mathcal{T}_{0,1}, \mathcal{T}_{0,1}$) *is liftable.*

Again, the proof is constructive, by showing that first-order knowledge compilation can compile circuits for a given KB, independent of ψ and ϕ and answer any query $P_n^{KB}(\phi \mid \psi)$ polynomially in the size of the domains, ϕ and ψ, provided that they contain no literals of arity two or higher.

This means that lifted inference is possible when we have (partial) type information about every object in the world. Evidence on properties (e.g., whether someone smokes) has arity one (e.g., smokes(X)). This induces a type for each object, which is the set of properties of that object. When there are k properties (e.g., smokes, drinks, etc.), each object can belong to one of 2^k types, and we can have partial evidence about the type of each object, which induces 3^k equivalence classes. Theorem 5.7 shows that probabilistic reasoning with such information is polynomial in the number of objects in the world.

6 Discussion and Conclusion

In this paper we have collected some existing and new results that are the beginning of a systematic analysis of complexity of inference from probabilistic-logic models. The general factorization of classes of inference problems according to knowledge base, query, and evidence-classes allows us to accommodate a variety of different complexity aspects in a coherent framework, with a particular focus on the question of liftability of inference. Furthermore, the framework we have introduced aligns the complexity of inference analysis with well-established concepts of syntactic complexity of predicate logic formulas, notably in terms of quantifier complexity and number of variables.

In a slightly different spirit, Domingos and Webb [30] introduce "Tractable Markov Logic (TML)", which also is a syntactically restricted probabilistic-logic modeling language. TML is not directly defined as a fragment of first-order logic fitting our weighted feature model framework, and therefore its placement into our complexity map of Table 1 is not immediate, but still seems quite feasible based on a precise first-order representation of TML syntax. It is remarkable, though, that TML is related to probabilistic description logics, and that description logics, in turn, are well-known to be representable in the 2-variable fragment of first-order logic [31]. Thus, it may well turn out that TML and PI(2-$\texttt{FFFOL}(=), \mathcal{T}, \mathcal{T}$) actually exploit the same underlying source of tractability, and that a combination of the two could lead to the identification of a tractable class that has both a natural and concise characterization, and that is more expressive than either of the two classes alone.

Further important open questions include:

- Complexity results for the 3-variable fragment 3-$\texttt{RFOL}$. Formulas with three variables provide important added expressivity compared to the 2-variable fragment. The transitive clause (Formula 5) and generalized homophily (Formula 6) are in 3-$\texttt{RFOL}$. Liftability for at least some inference classes PI(3-$\texttt{RFOL}$, ?, ?) would be an important extension.

- Complexity results for $\texttt{RFOL}$ with bounded weight parameters. Our lower complexity bound of Theorem 3.5 is based on the assumptions that weights can be arbitrarily small or big. For practical modeling tasks it would not be a significant limitation to restrict weights to lie between a certain lower and upper bound (in addition to zero weights for hard constraints). While, on the one hand, our results do not imply non-liftability for such weight-bounded knowledge bases, it is hard to imagine how a concrete exact inference algorithm could exploit weight-boundedness, since such an algorithm would then have to exhibit non-polynomial complexity with respect to the representation size of the weights. However, in the case of approximate inference, it is feasible that disallowing extreme weights improves worst-case approximation quality and therefore also approximate liftability.

Acknowledgments

Guy Van den Broeck is supported by the Research Foundation-Flanders (FWO-Vlaanderen).

References

[1] J. S. Breese. Construction of belief and decision networks. *Computational Intelligence*, 8(4):624–647, 1992.

[2] D. Poole. Probabilistic horn abduction and Bayesian networks. *Artificial Intelligence*, 64:81–129, 1993.

[3] T. Sato. A statistical learning method for logic programs with distribution semantics. In *Proceedings of the 12th International Conference on Logic Programming (ICLP'95)*, pages 715–729, 1995.

[4] L. Ngo, P. Haddawy, and J. Helwig. A theoretical framework for context-sensitive temporal probability model construction with application to plan

projection. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 419–426, 1995.

[5] M. Jaeger. Relational bayesian networks. In *Proceedings of the 13th Conference of Uncertainty in Artificial Intelligence (UAI-13)*, pages 266–273, 1997.

[6] N. Friedman, L. Getoor, D. Koller, and A. Pfeffer. Learning probabilistic relational models. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI-99)*, 1999.

[7] K. Kersting and L. De Raedt. Towards combining inductive logic programming with bayesian networks. In *Proceedings of the 11th International Conference on Inductive Logic Programming (ILP-01)*, volume 2157 of *LNAI*, pages 118–131, 2001.

[8] B. Milch, B. Marthi, S. Russell, D. Sontag, D.L. Ong, and A. Kolobov. Blog: Probabilistic logic with unknown objects. In *Proc. 19th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1352–1359, 2005.

[9] J. Vennekens, M. Denecker, and M. Bruynooghe. Representing causal information about a probabilistic process. In *Logics in Artificial Intelligence, 10th European Conference, JELIA 2006, Proceedings*, volume 4160 of *Lecture Notes in Computer Science*, pages 452–464. Springer, 2006.

[10] B. Taskar, P. Abbeel, and D. Koller. Discriminative probabilistic models for relational data. In *Proc. of UAI 2002*, 2002.

[11] M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 62(1-2):107 – 136, 2006.

[12] D. Poole. First-order probabilistic inference. In *Proceedings of IJCAI-03*, 2003.

[13] R. de Salvo Braz, E. Amir, and D. Roth. Lifted first-order probabilistic inference. In *Proceedings of the Nineteenth Int. Joint Conf. on Artificial Intelligence (IJCAI-05)*, pages 1319–1325, 2005.

[14] B. Milch, L. S. Zettlemoyer, K. Kersting, M. Haimes, and L. P. Kaelbling. Lifted probabilistic inference with counting formulas. In *Proc. of AAAI 2008*, 2008.

[15] J. Kisyński and D. Poole. Lifted aggregation in directed first-order probabilistic models. In *Proc. of IJCAI 2009*, 2009.

[16] A. Jha, V. Gogate, A. Meliou, and D. Suciu. Lifted inference seen from the other side: The tractable features. In *Proc. of NIPS*, 2010.

[17] G. Van den Broeck, N. Taghipour, W. Meert, J. Davis, and L. De Raedt. Lifted probabilistic inference by first-order knowledge compilation. In *Proc. of IJCAI 2011*, 2011.

[18] V. Gogate and P. Domingos. Probabilistic theorem proving. In *Proc. of UAI 2011*, 2011.

[19] G. Van den Broeck. On the completeness of first-order knowledge compilation for lifted probabilistic inference. In *Proc. of the 25th Annual Conf. on Neural Information Processing Systems (NIPS)*, 2011.

[20] U. Apsel and R. I. Brafman. Extended lifted inference with joint formulas. In *Proceedings of the Twenty-Seventh Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-11)*, pages 11–18. 2011.

[21] P. Singla, A. Nath, and P. Domingos. Approximate lifted belief propagation. In *Proc. of AAAI-10 Workshop on Statistical Relational AI*, 2010.

[22] M. Jaeger. On the complexity of inference about probabilistic relational models. *Artificial Intelligence*, 117:297–308, 2000.

[23] D. Fierens, G. Van den Broeck, I. Thon, B. Gutmann, and L. De Raedt. Inference in probabilistic logic programs using weighted cnf's. In *Proc. of UAI 2011*, 2011.

[24] G. Van den Broeck and J. Davis. Conditioning in first-order knowledge compilation and lifted probabilistic inference. In *Proceedings of the Twenty-Sixth AAAI Conference on Articial Intelligence*, 2012. To appear.

[25] M. Jaeger. Lower complexity bounds for lifted inference, 2012. http://arxiv.org/abs/1204.3255.

[26] N. D. Jones and A. L. Selman. Turing machines and the spectra of first-order formulas with equality. In *Proceedings of the Fourth ACM Symposium on Theory of Computing*, pages 157–167, 1972.

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

[28] N. L. Zhang and D. Poole. Exploiting causal independence in bayesian network inference. *Journal of Artificial Intelligence Research*, 5:301–328, 1996.

[29] L. G. Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8(3):410–421, 1979.

[30] P. Domingos and W. A. Webb. A tractable first-order probabilistic logic. In *Proc. of AAAI-12*, 2012. To appear.

[31] U. Hustadt, R.A. Schmidt, and L. Georgieva. A survey of decidable first-order fragments and description logics. *Journal of Relational Methods in Computer Science*, 1:251–276, 2004.