Ontology-Mediated Query Answering over Log-Linear Probabilistic Data

Stefan Borgwardt
Faculty of Computer Science
Technische Universität Dresden, Germany
stefan.borgwardt@tu-dresden.de

İsmail İlkan Ceylan and **Thomas Lukasiewicz**
Department of Computer Science
University of Oxford, UK
ismail.ceylan@cs.ox.ac.uk
thomas.lukasiewicz@cs.ox.ac.uk

Abstract

Large-scale knowledge bases are at the heart of modern information systems. Their knowledge is inherently uncertain, and hence they are often materialized as probabilistic databases. However, probabilistic database management systems typically lack the capability to incorporate implicit background knowledge and, consequently, fail to capture some intuitive query answers. Ontology-mediated query answering is a popular paradigm for encoding commonsense knowledge, which can provide more complete answers to user queries. We propose a new data model that integrates the paradigm of ontology-mediated query answering with probabilistic databases, employing a log-linear probability model. We compare our approach to existing proposals, and provide supporting computational results.

1 Introduction

Advances in automated knowledge base construction have led to successful systems, such as DeepDive (Shin et al. 2015), NELL (Mitchell *et al.* 2015), and Google's Knowledge Vault (Dong et al. 2014). They extract *structured knowledge* from multiple sources, through a chain of statistical techniques, and produce *probabilistic knowledge bases* (PKBs). The basic data model underlying these systems is given by *probabilistic databases* (PDBs) (Suciu et al. 2011); see recent surveys focusing on PKBs (Van den Broeck and Suciu 2017; Borgwardt, Ceylan, and Lukasiewicz 2018).

PKBs are inherently *incomplete*, which makes reasoning more challenging. One common way to alleviate incompleteness is to encode *commonsense knowledge*, in the form of logical theories, to allow for deductions that go beyond existing facts in the knowledge base. Unifying first-order logic (FOL) and probability is an old endeavor in artificial intelligence (Halpern 2003); there is a vast literature on models with such capabilities. Here, we confine ourselves to more recent proposals with a special emphasis on large-scale PKBs.

Statistical relational models are *concise*, and *lifted* representations of probabilistic graphical models (Getoor and Taskar 2007). Well-known examples include Markov logic networks (MLNs) (Richardson and Domingos 2006), relational Bayesian networks (Jaeger 1997), and approaches to probabilistic logic programming (PLP). All these models can encode commonsense knowledge, but they are based on the *closed-domain assumption (CDA)* that requires the set of relevant objects to be *finite*, and *known* at design-time, which is not always an easy condition to be met.

Other proposals that can encode commonsense knowledge while allowing an *open domain* include PLP with function symbols (Sato and Kameya 1997; De Raedt, Kimmig, and Toivonen 2007), the probabilistic programming language BLOG (Milch et al. 2005), and ontology-based approaches (Jung and Lutz 2012; Borgwardt, Ceylan, and Lukasiewicz 2017). The latter are further distinguished from the rest by the *open-world assumption*, i.e., they do not interpret the absence of facts as the negation of these facts; this means that the incomplete nature of the PKB is respected.

We build on the rich tradition of *ontology languages*, and propose a robust data model, based on *log-linear* probability distributions, for reasoning in PKBs. We assume the database given as a set of facts with associated *weights*, which is then interpreted as a log-linear model. Our inspiration comes from MLNs, which are expressive probabilistic-logical models that use log-linear distributions. As in MLNs, we restrict the probability distribution to the *known* objects, but additionally use first-order semantics over arbitrary, *possibly infinite* domains, whereby we achieve open-world, open-domain reasoning.

We briefly summarize this paper's main contributions. We introduce a new data model for ontology-mediated query answering over probabilistic data, based on log-linear probability distributions. We compare it to existing data models, including MLNs and PDBs, and highlight the semantic differences. We then show that reasoning in our model can be reduced (via polynomial rewriting techniques) to inference in MLNs, or PDBs. These results are significant given the expressive nature of our formalism. As a consequence of the above reductions, a whole host of computational complexity results from previous models carry over to the new probabilistic data model. We conclude by describing a new approach to learn the weights for our model, based on the principle of *maximum entropy*, to establish the connection with existing PKBs. This is independent of the other results, however; in principle, we could use any other weight learning method.

All proofs can be found at tu-dresden.de/inf/lat/papers.

2 Motivation

In this section, we clarify the motivation of this work on two concrete examples. More specifically, we show that MLNs and PDBs as two of the most popular existing probabilistic data models are unable to directly capture some natural modeling capabilities needed for ontology-mediated query answering over probabilistic data in practice. We first illustrate the problems with the CDA when dealing with logical theories, in particular with existential quantification.

Example 1 (CDA)**.** The first-order logical constraint

$$\forall x\, \mathsf{Emp}(x) \rightarrow \exists y\, \mathsf{Address}(x, y) \tag{1}$$

expresses that every employee has an address. In the closed domain $\mathbf{C} = \{c_1, \ldots, c_n\}$, this formula is equivalent to

$$\forall x\, \mathsf{Emp}(x) \rightarrow \mathsf{Address}(x, c_1) \vee \cdots \vee \mathsf{Address}(x, c_n). \tag{2}$$

There are two problems with this representation. First, it says that all employee's addresses must be one of the *known objects* in the database. If the address of some new employees is still unknown, in each world, they will be randomly assigned the address of another employee. A common remedy is to introduce a fixed number of auxiliary objects into $\mathbf{C}$ that can serve as "unknown addresses". However, it is not reasonable to assume that all objects of interest can be known a priori, and no other objects exist. In particular, it is unclear *how many* additional objects need to be taken into account.

The second problem is the large disjunction in (2). This is very impractical, as it introduces a huge amount of *nondeterminism*. For closed-domain models like MLNs, this is a known problem, and more sophisticated techniques to eliminate existential quantification exist (Van den Broeck, Meert, and Darwiche 2014). However, in the worst case, these techniques also cannot avoid the nondeterminism over the domain of constants. Hence, in theory, MLNs allow for arbitrary first-order formulas, but this is not the case in practice. In fact, almost all MLN implementations operate solely on universally quantified formulas (Niu et al. 2011; Domingos and Lowd 2009).

This inefficiency appears also in ontology languages. For example, (1) can be formulated in the description logic $\mathcal{EL}_\bot$, where reasoning is P-complete, but becomes NP-complete when restricted to closed domains (Gaggl, Rudolph, and Schweizer 2016). In contrast, for ontology languages under the open-world assumption (OWA), it is sufficient to introduce a *single* anonymous individual to satisfy (1), which is a deterministic operation. Although the OWA means that one has to do this in infinitely many interpretations, the deterministic nature of $\mathcal{EL}_\bot$ entails that one can restrict the attention to a single representative *universal model*. ■

The CDA may be reasonable for certain application domains, but as shown in Example 1, this is often not the case (even if we consider trivial domains). Noteworthy is also the fact that the CDA does not necessarily imply efficiency in comparison to open-domain models.

Another problem that is inherent to ontology-based probabilistic models is related to inconsistent worlds, which are usually removed, and the resulting probability distribution is *renormalized* by uniformly distributing the probability of the inconsistent worlds among the consistent ones. However, this does not always select the most reasonable distribution.

Example 2 (Inconsistency)**.** Consider the following PDB $\mathcal{P}$ and theory $\mathcal{T}$:

$$\mathcal{P} \coloneqq \{\langle \mathsf{A}(a) : 0.5\rangle, \langle \mathsf{B}(a) : 0.5\rangle\}$$
$$\mathcal{T} \coloneqq \{\forall x\, \mathsf{A}(x) \rightarrow \mathsf{B}(x)\}.$$

The possible worlds are then given as

$$\mathcal{W}_1 \coloneqq \{\mathsf{A}(a), \mathsf{B}(a)\}, \mathcal{W}_2 \coloneqq \{\mathsf{A}(a), \neg\mathsf{B}(a)\},$$
$$\mathcal{W}_3 \coloneqq \{\neg\mathsf{A}(a), \mathsf{B}(a)\}, \mathcal{W}_4 \coloneqq \{\neg\mathsf{A}(a), \neg\mathsf{B}(a)\}.$$

Without $\mathcal{T}$, each of these worlds has the probability 0.25, by the independence assumptions of $\mathcal{P}$. However, since $\mathcal{W}_2$ is inconsistent with $\mathcal{T}$, its probability is reduced to 0, and the probability of the remaining worlds is renormalized to add up to 1, yielding a probability of 0.33 each. However, this also means that, as an undesired side effect, the probabilities for $\mathsf{A}(a)$ and $\mathsf{B}(a)$ change to 0.33 and 0.66, respectively. ■

In this paper, we propose a different approach to integrate a given PDB into our model (see Section 7). We argue that the observed probabilities should be preserved, and try to find a log-linear distribution that deviates from these input values as little as possible. In Example 2, by assigning both $\mathcal{W}_2$ and $\mathcal{W}_3$ a probability of 0 and the remaining worlds 0.5 each, we obtain a model that satisfies the constraints of both $\mathcal{T}$ and $\mathcal{P}$. This respects both the probabilistic and the logical input, and does not favor one over the other, which results in a more fine-grained approach than simply renormalizing the probabilities of the consistent worlds. However, the new model that we introduce in Section 4 is independent of the precise method used to obtain the probability distribution, and can use the standard renormalization, if so desired.

3 Preliminaries

We recall FOL, PDBs, and MLNs from a model-theoretic perspective, and highlight their main assumptions.

FOL. We consider a relational vocabulary consisting of finite, mutually disjoint sets $\mathbf{R}$ and $\mathbf{C}$ of *predicates* and *constants*, respectively, and a (possibly infinite) set $\mathbf{V}$ of *variables*. A first-order *formula* is built as usual from *atoms* $\mathsf{R}(s_1, \ldots, s_n)$ over the given vocabulary, *truth constants* $\top$, $\bot$, *operators* $\neg$, $\vee$, $\wedge$, $\rightarrow$, and *quantifiers* $\exists$, $\forall$. A *ground atom* (also *fact* or *tuple*) is an atom where all terms s_i are constants. A *quantifier-free formula* is a formula that does not use quantifiers. A variable in a formula is *quantified* (or *bound*), if it is in the scope of a quantifier; otherwise, it is *free*. A *sentence* is a formula without any free variables. A *theory* (or *ontology*) is a finite set of sentences. A *ground instance* of a formula $\Phi(\mathbf{x})$ with free variables $\mathbf{x}$ is a sentence of the form $\Phi(\mathbf{c})$, where $\mathbf{c}$ are constants from $\mathbf{C}$.

The semantics of FOL is given by means of *interpretations* $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$, where $\Delta^\mathcal{I}$ is a *possibly infinite* domain, and $\cdot^\mathcal{I}$ is an interpretation function that maps every constant a to a domain element $a^\mathcal{I} \in \Delta^\mathcal{I}$, and every predicate R with arity n to a relation $\mathsf{R}^\mathcal{I} \subseteq (\Delta^\mathcal{I})^n$. A sentence Φ is *satisfied* by an interpretation, if $\mathcal{I} \models \Phi$, where $\models$ is the standard first-order satisfaction relation. An interpretation $\mathcal{I}$ is a *model* of

a theory $\mathcal{T}$, denoted $\mathcal{I} \models \mathcal{T}$, if $\mathcal{I}$ satisfies all $\Phi \in \mathcal{T}$. $\mathcal{T}$ is *consistent*, if it has a model. $\mathcal{T}$ *entails* a sentence Φ, written $\mathcal{T} \models \Phi$, if all models of $\mathcal{T}$ are also models of Φ.

Databases and Query Answering. A database $\mathcal{D}$ is a finite set of facts over the (finite) vocabulary. From a model-theoretic perspective, $\mathcal{D}$ is a first-order interpretation, where

(i) the domain is given as $\Delta^{\mathcal{D}} = \mathbf{C}$,

(ii) $c^{\mathcal{D}} = c$, for all constants $c \in \mathbf{C}$,

(iii) $(c_1, ..., c_n) \in \mathsf{R}^{\mathcal{D}}$ iff $\mathsf{R}(c_1, ..., c_n) \in \mathcal{D}$.

Notably, (i) corresponds to the *closed-domain assumption* (CDA), (ii) to the *unique name assumption (UNA)*, and (iii) to the *closed-world assumption (CWA)* of databases.

The core task of databases is *query answering*. A *query* is simply a first-order formula. A *conjunctive query (CQ)* is an existentially quantified formula $\exists \mathbf{x}\, \phi$, where ϕ is a conjunction of atoms. A *union of conjunctive queries (UCQ)* is a disjunction of CQs. A query is *Boolean* if it is a sentence. In query answering, we want to find all *answers* to a query Q over a database $\mathcal{D}$, which are assignments of the free variables in Q to constants in $\mathbf{C}$ such that the resulting ground instance is satisfied in $\mathcal{D}$. We focus on Boolean queries Q and the associated decision problem of *query evaluation*, i.e., deciding whether Q is satisfied in $\mathcal{D}$, denoted $\mathcal{D} \models \mathsf{Q}$.

Ontology-Mediated Queries. Reasoning in FOL is undecidable, which motivated the study of fragments, to navigate the trade-off between high expressivity and low computational complexity. Ontology languages based on Datalog$^{\pm}$ (Calì, Gottlob, and Lukasiewicz 2012; Calì, Gottlob, and Kifer 2013) and description logics (Baader et al. 2007) are widely studied examples of such fragments. *Ontology-mediated query answering* is the task of determining whether a query Q is *entailed* by the database $\mathcal{D}$ with the help of an additional ontology $\mathcal{T}$, i.e., whether $\mathcal{D} \cup \mathcal{T} \models \mathsf{Q}$ holds. Since first-order entailment considers many interpretations with arbitrary domains, neither the CDA nor the CWA hold in this context.

Probabilistic Query Answering. We discuss two basic probabilistic models: PDBs and MLNs. Both of them define probability distributions P over the set of *possible worlds*, which correspond to complete, deterministic states. Formally, a *world* $\mathcal{W}$ is a set that contains, for each fact t over $\mathbf{R}$ and $\mathbf{C}$, either t or its negation $\neg t$ (in contrast to databases, the tuples that do not hold are represented explicitly).

Given a probabilistic model P, the main task is to compute the probability of first-order queries Q, defined as follows:

$$\mathrm{P}(\mathsf{Q}) \coloneqq \sum_{\mathcal{W} \models \mathsf{Q}} \mathrm{P}(\mathcal{W}), \tag{3}$$

where $\models$ denotes first-order satisfaction, i.e., the question whether Q holds in the finite first-order interpretation over the domain $\mathbf{C}$ that is described by the world $\mathcal{W}$.

PDBs. The most elementary probabilistic data model is based on the tuple-independence assumption. It is the basis of the research on PDBs, although more sophisticated models exist (Suciu et al. 2011).

Given a finite relational vocabulary, a *probabilistic database* $\mathcal{P}$ is defined as a set of *probabilistic facts* $\langle t : p \rangle$, where t is a fact and $p \in [0, 1]$, such that $\langle t : p \rangle, \langle t : p' \rangle \in \mathcal{P}$ implies $p = p'$. The probability of a fact t, denoted $\mathrm{P}_{\mathcal{P}}(t)$, is p, if $\langle t : p \rangle \in \mathcal{P}$, and 0, otherwise. The latter case reflects the CWA of PDBs: any fact that is not in the PDB gets assigned the probability 0. By the tuple-independence assumption, the probability of a world $\mathcal{W}$ is given as

$$\mathrm{P}_{\mathcal{P}}(\mathcal{W}) \coloneqq \prod_{t \in \mathcal{W}} \mathrm{P}_{\mathcal{P}}(t) \cdot \prod_{\neg t \in \mathcal{W}} (1 - \mathrm{P}_{\mathcal{P}}(t)). \tag{4}$$

Importantly, *all assumptions* of databases (i)–(iii) are employed in PDBs, while the tuple-independence assumption (iv) is an additional assumption on the probability space.

MLNs. MLNs (Richardson and Domingos 2006) were introduced as a template language for statistical relational AI. An MLN $\mathcal{M}$ is determined by a set $\mathcal{T}$ of first-order formulas and a weight function w assigning a rational weight w_Φ to each $\Phi \in \mathcal{T}$. A formula with weight ∞ is a *hard constraint*, the others are *soft constraints*. $\mathcal{M}$ determines the probability of each world $\mathcal{W}$ as

$$\mathrm{P}_{\mathcal{M}}(\mathcal{W}) \coloneqq \tfrac{1}{Z} \exp\Big(\sum_{\Phi \in \mathcal{T}} w_\Phi n_\Phi(\mathcal{W}) \Big), \tag{5}$$

where $n_\Phi(\mathcal{W})$ is the *number* of ground instances of Φ that are satisfied in $\mathcal{W}$; Z is a normalization factor.

Differently from PDBs, MLNs allow first-order formulas, which introduce dependencies, i.e., there is no assumption of independence of tuples. Another difference is that MLNs do not employ the CWA, i.e., a fact t that is not explicitly given a weight w_t can still get a positive probability, which is not the case in PDBs. However, the assumptions (i) and (ii) are also present in MLNs. Specifically, the CDA remains as an essential ingredient of the semantics, i.e., only instantiations over the finitely many constants from $\mathbf{C}$ are considered; in practice, these are the objects from the input data, possibly extended by a fixed, finite number of additional elements.

4 Log-Linear PKBs

We assume that the probabilistic data is represented in terms of a *weighted database* $\mathcal{D}_w$, which is a finite set of tuples t with associated rational weights w_t. We use *log-linear probabilistic knowledge bases* for reasoning over weighted databases using ontological background knowledge.

Definition 3. A *log-linear PKB* $\mathcal{K} = (\mathcal{D}_w, \mathcal{T})$ consists of a weighted database $\mathcal{D}_w$ and a theory $\mathcal{T}$. It defines the following distribution over worlds $\mathcal{W}$. If $\mathcal{W}$ is consistent with $\mathcal{T}$, i.e., $\mathcal{W} \cup \mathcal{T} \not\models \bot$, then

$$\mathrm{P}_{\mathcal{K}}(\mathcal{W}) \coloneqq \tfrac{1}{Z} \exp\Big(\sum_{t \in \mathcal{W}} w_t \Big), \tag{6}$$

and $\mathrm{P}_{\mathcal{K}}(\mathcal{W}) \coloneqq 0$, otherwise; Z is a normalization factor.

There are two main differences to MLNs. First, when checking $\mathcal{W} \cup \mathcal{T} \not\models \bot$, we employ standard first-order semantics, in contrast to the CDA used by MLNs. Second, our model allows only hard constraints (except for facts). The reason is that under the open-domain assumption it does not make sense to count the (infinite) number of satisfying assignments of a formula. Instead, our model supports a weaker kind of weighted *sentences*: we can simulate a sentence Φ with weight w_Φ by a fresh 0-ary fact $\mathsf{R}_\Phi()$ with the same

weight and the modified sentence $\mathsf{R}_\Phi() \to \Phi$. For example, an uncertain version of the rule $r = \forall x\, \mathsf{A}(x) \to \mathsf{B}(x)$ in Example 2 is given by $\forall x\, \mathsf{R}() \wedge \mathsf{A}(x) \to \mathsf{B}(x)$ along with a weight $w_{\mathsf{R}()}$ for the fact $\mathsf{R}()$. Intuitively, this then means that r as a whole holds only with some uncertainty in each world, as governed by $w_{\mathsf{R}()}$. This essentially extends the worlds to also cover the sentences in $\mathcal{T}$, i.e., each world specifies which of the sentences from $\mathcal{T}$ should hold in it.

We consider two probabilistic inference problems: query evaluation and maximum a posteriori computation.

Definition 4. Given $\mathcal{K} = (\mathcal{D}_w, \mathcal{T})$ and a first-order query Q, the probability of Q in $\mathcal{K}$ is given by

$$\mathrm{P}_\mathcal{K}(\mathsf{Q}) \coloneqq \sum_{\mathcal{W} \cup \mathcal{T} \models \mathsf{Q}} \mathrm{P}_\mathcal{K}(\mathcal{W}).$$

Query evaluation is the task of deciding $\mathrm{P}_\mathcal{K}(\mathsf{Q}) > p$ for some query Q and threshold $p \in [0, 1)$.

Again, we do not merely sum over all worlds that satisfy the query, but those that, together with $\mathcal{T}$, *entail* the query, according to the open-domain semantics of FOL.

Definition 5. Given a PKB $\mathcal{K} = (\mathcal{D}_w, \mathcal{T})$ and a query Q, a *most probable database (MPD)* is a world $\mathcal{W}$ with $\mathcal{W} \cup \mathcal{T} \models \mathsf{Q}$ that maximizes $\mathrm{P}_\mathcal{K}(\mathcal{W})$. The corresponding decision problem is to decide whether there exists a world $\mathcal{W}$ such that $\mathrm{P}_\mathcal{K}(\mathcal{W}) > p$ and $\mathcal{W} \cup \mathcal{T} \models \mathsf{Q}$, for a given $p \in [0, 1)$.

The MPD problem was investigated for PDBs in (Gribkoff, Van den Broeck, and Suciu 2014; Ceylan, Borgwardt, and Lukasiewicz 2017) and is an extension of maximum a posteriori inference (MAP), which is widely studied for statistical relational models, including MLNs. More precisely, MAP is the special case of MPD where the query is fixed to $\top$.

5 Semantic Results

We recall some techniques for query evaluation over ontologies, as they are crucial ingredients for our results. An *ontology-mediated query (OMQ)* is a pair $(\mathsf{Q}, \mathcal{T})$, where Q is a UCQ, and $\mathcal{T}$ is a theory. A prominent paradigm to evaluate such compound queries is based on the notion of *rewritability*. Formally, a query Q is *FO-rewritable* (resp., *Datalog-rewritable)* w.r.t. $\mathcal{T}$, if there is a first-order query (resp., Datalog query) $\mathsf{Q}_\mathcal{T}$ such that, for every world $\mathcal{W}$ consistent with $\mathcal{T}$ (i.e., $\mathcal{W} \cup \mathcal{T} \not\models \bot$), we have

$$\mathcal{W} \cup \mathcal{T} \models \mathsf{Q} \text{ iff } \mathcal{W} \models \mathsf{Q}_\mathcal{T},$$

i.e., $\mathsf{Q}_\mathcal{T}$ is satisfied in $\mathcal{W}$ when considered as a finite interpretation. Similarly, $\mathcal{W} \cup \mathcal{T} \models \bot$ iff $\mathcal{W} \models \bot_\mathcal{T}$ for a rewriting $\bot_\mathcal{T}$ of $\bot$ (i.e., $\bot_\mathcal{T}$ is a query that encodes inconsistency w.r.t. $\mathcal{T}$). Notably, many ontology languages admit efficient rewritings to one of these query languages (Gottlob and Schwentick 2012; Eiter et al. 2012).

We make the standard *data complexity assumption*, i.e., the set $\mathbf{R}$ and the size of $\mathcal{T}$ are fixed. In particular, the maximal arity of predicates is fixed, and hence there are polynomially many facts over $\mathbf{C}$, and exponentially many worlds. This assumption is standard for PDBs and MLNs, and also central in research on OMQ answering. In the combined complexity, inference in MLNs is already super-exponential, since the size of each possible world is exponential.

We now present our techniques to reduce log-linear PKBs directly to other models, in order to use existing inference methods. In Section 5.1, we describe reductions of the ontological component that produce MLNs, while in Section 5.2 we reduce the probabilistic component to the tuple-independent model of PDBs, inspired by (Gribkoff and Suciu 2016).

5.1 Reductions to MLNs

Let $\mathcal{K} = (\mathcal{D}_w, \mathcal{T})$ be a PKB and Q a query. We start with a simple observation. As explained before, the difference between Equations (5) and (6) lies in the open-domain entailment. It is easy to see that this difference is diminished when $\mathcal{T}$ contains no existential quantifiers (cf. Example 1).

Theorem 6. *If all formulas in $\mathcal{T}$ are of the form $\forall \mathbf{x}\, \phi(\mathbf{x})$, where ϕ is quantifier-free, then we can construct an MLN $\mathcal{M}$ with $\mathrm{P}_\mathcal{K} = \mathrm{P}_\mathcal{M}$ in linear time.*

Thus, our model naturally devolves into a special kind of MLN, if there are no open-domain existential quantifiers. However, closed-domain existential quantification can still be expressed (Van den Broeck, Meert, and Darwiche 2014).

To deal with existential quantifiers over *open* domains, we can encode Datalog rewritings into an MLN and use existing MLN systems for query evaluation over open-domain PKBs.

Theorem 7. *If Q and $\bot$ are Datalog-rewritable w.r.t. $\mathcal{T}$, then query evaluation in log-linear PKBs can be reduced to inference in MLNs in polynomial time.*

Proof Sketch. A naive idea would be to use the Datalog rules in $\mathsf{Q}_\mathcal{T}$ and $\bot_\mathcal{T}$ directly as formulas in an MLN $\mathcal{M}$. However, the rewriting process can introduce additional predicates, which means that $\mathcal{M}$ has "larger" worlds than $\mathcal{K}$. We need to restrict these extended worlds $\mathcal{W}'$ over the signature of $\mathcal{M}$, such that for each original world $\mathcal{W}$ over $\mathcal{K}$, there is a *unique consistent extension* $\mathcal{W}'$.

Fortunately, there exists a theory $\mathcal{T}'$ such that $\mathcal{W} \cup \mathcal{T}'$ has exactly one Herbrand model (corresponding to $\mathcal{W}'$), namely, the minimal Herbrand model of $\mathcal{W} \cup \mathsf{Q}_\mathcal{T} \cup \bot_\mathcal{T}$. This theory, called the *tight completion* of the Datalog program $\mathsf{Q}_\mathcal{T} \cup \bot_\mathcal{T}$ (Wallace 1993) is based on the idea of Clark's completion, which essentially replaces the implication ($\to$) in Datalog rules by equivalence ($\leftrightarrow$). However, since this is only correct for *nonrecursive* Datalog programs, additional care needs to be taken for recursive dependencies.

By assigning the unique extension $\mathcal{W}'$ the same probability as the original world $\mathcal{W}$, we ensure that $\mathcal{M}$ evaluates the Datalog rewriting to the same probability as $\mathrm{P}_\mathcal{K}(\mathsf{Q})$. □

Hence, ontology-based rewriting techniques can *augment MLNs with open-domain existential quantifiers* essentially *for free* (in data complexity). Moreover, there are many rewriting techniques that only result in a polynomial blowup of the formulas (Bienvenu and Ortiz 2015).

5.2 Reductions to PDBs

There is also a close connection between log-linear PKBs and *OMQs over PDBs*, as used in (Borgwardt, Ceylan, and Lukasiewicz 2017). There, the distribution is simply $\mathrm{P}_\mathcal{P}$ for

Datalog$^{\pm}$ Languages	data	fixed-program	bounded-arity
L, S, LF, AF, SF	PP	PP^{NP}	PP^{NP}
A	PP	PP^{NP}	NEXP
GF, F	PP	PP^{NP}	PP^{NP}
G	PP	PP^{NP}	EXP
WS, WA	PP	PP^{NP}	2EXP
WG	EXP	EXP	EXP

Table 1: UCQ evaluation over log-linear PKBs.

Datalog$^{\pm}$ Languages	data	fixed-program	bounded-arity
L, S, LF, AF, SF	NP	NP	Σ_2^{P}
A	NP	NP	P^{NE}
GF, F	NP	NP	Σ_2^{P}
G	NP	NP	EXP
WS, WA	NP	NP	2EXP
WG	EXP	EXP	EXP

Table 2: MPD for UCQs over log-linear PKBs.

a PDB $\mathcal{P}$, but $\mathcal{T}$ is viewed as part of an OMQ $(\mathsf{Q}, \mathcal{T})$. The probability of this is defined as

$$\text{P}_{\mathcal{P}}(\mathsf{Q}, \mathcal{T}) \coloneqq \sum_{\mathcal{W} \cup \mathcal{T} \models \mathsf{Q}} \text{P}_{\mathcal{P}}(\mathcal{W}),$$

where $\models$ is again the open-domain entailment relation (cf. Definition 4). In that setting, inconsistent worlds can have a positive probability, and simple renormalization is applied:

$$\text{P}^{\text{n}}_{\mathcal{P}}(\mathsf{Q}, \mathcal{T}) \coloneqq \frac{\text{P}_{\mathcal{P}}(\mathsf{Q}, \mathcal{T}) - \text{P}_{\mathcal{P}}(\bot, \mathcal{T})}{1 - \text{P}_{\mathcal{P}}(\bot, \mathcal{T})}.$$

Theorem 8. *Query evaluation in log-linear PKBs can be reduced to OMQ evaluation over tuple-independent PDBs, and vice versa, in polynomial time.*

Proof Sketch. We convert each weight w_t into a probability $\frac{\exp(w_t)}{(1+\exp(w_t)}$, which accounts for the factor $\exp(w_t)$ in (6) and the absence of $1 - \exp(w_t)$ when compared to (4). Moreover, since PKBs do not make the closed-world assumption, we need to add all tuples t that do not occur in $\mathcal{D}_w$ with a neutral probability of 0.5. In data complexity, there are only polynomially many such tuples. Under these transformations, the normalization in the resulting PDB coincides with the normalization factor Z in (6).

For the other direction, we replace each tuple probability p by the weight $\log(p) - \log(1-p)$, and assign all tuples that do not occur in the PDB the weight $-\infty$. □

We can adapt the reduction in Theorem 8 to obtain the following result for the MPD problem.

Theorem 9. *The MPD problem for log-linear PKBs can be reduced to the MPD problem for OMQs over PDBs, and vice versa, in polynomial time.*

6 Complexity Results

So far, we have shown several generic reductions to existing probabilistic-logical models. We now analyse the *complexity* of query evaluation over log-linear PKBs in more detail.

6.1 General Results

By the reductions of Theorem 8, we can import the complexity results for OMQ evaluation over PDBs from (Borgwardt, Ceylan, and Lukasiewicz 2017; Ceylan 2017):

Corollary 10. *If query answering in a Datalog$^{\pm}$ language $\mathcal{L}$ is $\mathfrak{C}$-complete, then query evaluation over log-linear PKBs $(\mathcal{D}_w, \mathcal{T})$ with $\mathcal{T} \in \mathcal{L}$ is* PP*-hard, $\mathfrak{C}$-hard and in* $\text{PP}^{\mathfrak{C}}$.

This yields a data complexity of PP for UCQs in Datalog, $\mathcal{EL}_{\bot}$, or *guarded* (G) Datalog$^{\pm}$ (Dantsin et al. 2001; Calì, Gottlob, and Lukasiewicz 2012), and even expressive languages like weakly-acyclic (WA) or weakly-sticky (WS) Datalog$^{\pm}$ (Fagin et al. 2005; Calì, Gottlob, and Pieris 2012). Table 1 gives an overview of the complexity of query evaluation in log-linear PKBs for some selected ontology languages of the Datalog$^{\pm}$ family, where *fixed-program* refers to the assumption that only the theory (not the query) is viewed as fixed, while in the *bounded-arity* case the arity of all predicates is fixed. The reductions in Theorem 8 are still polynomial under these assumptions. The table includes tight complexity results beyond Corollary 10 (Borgwardt, Ceylan, and Lukasiewicz 2017; Ceylan 2017): UCQ evaluation is PP^{NP}-hard in the fixed-program case, even if the theory is empty, and is in NEXP in the bounded-arity case for acyclic (A) Datalog$^{\pm}$ theories.

We get similar results in Table 2 for the MPD problem via (Ceylan, Borgwardt, and Lukasiewicz 2017; Ceylan 2017).

Corollary 11. *If query answering in a Datalog$^{\pm}$ language $\mathcal{L}$ is $\mathfrak{C}$-complete, then MAP for log-linear PKBs $(\mathcal{D}_w, \mathcal{T})$ with $\mathcal{T} \in \mathcal{L}$ is* NP*-hard, $\mathfrak{C}$-hard and in* $\text{NP}^{\mathfrak{C}}$.

Additionally, MAP is Σ_2^{P}-hard in the bounded-arity case in all the languages of Table 2, it is only in NP in the fixed-program case, and is P^{NE}-hard for acyclic Datalog$^{\pm}$.

6.2 Tractability Results

The reduction of Theorem 8 adds polynomially many tuples t with probability 0.5 to $\mathcal{P}$, which is not efficient. Fortunately, if we can transform the query into a UCQ, we can apply a *lifted inference* method from (Ceylan, Darwiche, and Van den Broeck 2016) for default probabilities in PDBs. This only applies to so-called *safe UCQs*, but guarantees query evaluation in polynomial time. Safe UCQs are defined by a syntactic restriction that can be checked in polynomial time; for details, see (Ceylan, Darwiche, and Van den Broeck 2016). In contrast to Theorem 7, this result puts stronger restrictions on the query, but allows for query evaluation in P.

Theorem 12. *If Q, $\bot$ are FO-rewritable w.r.t. $\mathcal{T}$ and $\mathsf{Q}_{\mathcal{T}} \vee \bot_{\mathcal{T}}$ and $\bot_{\mathcal{T}}$ are safe UCQs, then $\text{P}_{\mathcal{K}}(\mathsf{Q})$ can be evaluated in* P.

7 Learning the Weights

As is common for probabilistic models, we need to address the question of how to obtain the parameters w_t and Z. There is a large body of research on learning log-linear models using the principle of *maximum entropy* (ME) in the presence of (logical) constraints (Bacchus et al. 1996; Kuželka et al. 2018). Of all distributions consistent with the observations, the idea is to select the one with maximum entropy; in information-theoretic terms, this is the one that makes the fewest additional dependency assumptions necessary to be consistent with our knowledge base (Paskin 2001).

We also follow this direction here, but we propose a particular ME formulation that draws the input data from existing PKBs like NELL and Knowledge Vault (Mitchell *et al.* 2015; Dong et al. 2014), represented as PDBs. Importantly, this is a process that can be done once, in an off-line processing phase, before the actual query answering takes place.

Definition 13 (ME Problem)**.** Given a PDB $\mathcal{P}$ and a theory $\mathcal{T}$, find the probability distribution P that

$$\begin{aligned} &\textit{maximizes } H(\mathrm{P}) \coloneqq -\sum_{\mathcal{W}} \mathrm{P}(\mathcal{W}) \log \mathrm{P}(\mathcal{W}) \qquad (7)\\ &\textit{subject to } \sum_{\mathcal{W}} \mathrm{P}(\mathcal{W}) = 1\\ &\qquad \mathrm{P}(\mathcal{W}) = 0 \text{ for all worlds } \mathcal{W} \text{ with } \mathcal{W} \cup \mathcal{T} \vDash \bot\\ &\qquad \mathrm{P}(t) = [p - \ell_t, p + u_t] \text{ for all } \langle t : p\rangle \in \mathcal{P}\\ &\qquad \sum_{\langle t:p\rangle \in \mathcal{P}} \ell_t + u_t = \varepsilon \end{aligned}$$

over the non-negative variables $\mathrm{P}(\mathcal{W})$, ℓ_t, and u_t.

The first constraint requires that the probability of all worlds adds up to 1. Second, inconsistent worlds should have the probability 0. The remaining constraints say that the probabilities from the PDB should be respected as much as possible, depending on a parameter ε (ℓ_t and u_t are additional slack variables). If $\varepsilon = 0$, then $\mathrm{P}_{\mathcal{K}}(t) = p$ for all $\langle t : p\rangle \in \mathcal{P}$, which cannot always be ensured: if the probabilities of many tuples are fixed to a specific value, then there may not be enough consistent worlds to realize all of them. In this case, we can increase ε to guarantee the existence of a solution.

Finding ε. The value for ε is obtained in a preprocessing step, by a separate optimization; cf. (Hansen et al. 1995):

$$\begin{aligned} &\textit{Minimize } \varepsilon \coloneqq \sum_{\langle t:p\rangle \in \mathcal{P}} \ell_t + u_t \qquad (8)\\ &\textit{subject to } \sum_{\mathcal{W}} \mathrm{P}(\mathcal{W}) = 1\\ &\qquad \mathrm{P}(\mathcal{W}) = 0 \text{ for all worlds } \mathcal{W} \text{ with } \mathcal{W} \cup \mathcal{T} \vDash \bot\\ &\qquad \mathrm{P}(t) = [p - \ell_t, p + u_t] \text{ for all } \langle t : p\rangle \in \mathcal{P}. \end{aligned}$$

That is, we minimize ε with the to obtain the most constrained feasible region for (7) that is still non-empty. Since (8) is a *linear program*, it is easier to solve. However, there may be (infinitely) many solutions that yield the same ε.

Example 14. Consider the PDB $\mathcal{P}$ and the theory $\mathcal{T}$ where

$$\begin{aligned} \mathcal{P} &= \{\langle \mathsf{A}(a) : 0.9\rangle, \langle \mathsf{B}(a) : 0.1\rangle\},\\ \mathcal{T} &= \{\forall x\, \mathsf{A}(x) \to \mathsf{B}(x)\}, \end{aligned}$$

and $\mathcal{W}_1, \ldots, \mathcal{W}_4$ from Example 2. If $\varepsilon = 0$, then (7) has no solution: $\mathcal{W}_2 \cup \mathcal{T} \vDash \bot$, and hence $0.9 = \mathrm{P}(A(a)) = \mathrm{P}(\mathcal{W}_1)$ and $0.1 = \mathrm{P}(\mathsf{B}(a)) = \mathrm{P}(\mathcal{W}_1) + \mathrm{P}(\mathcal{W}_3)$ are contradictory.

Any solution for (8) must satisfy $\varepsilon = l_{\mathsf{A}(a)} + u_{\mathsf{B}(a)} = 0.8$, i.e., $\mathrm{P}(\mathsf{A}(a)) = 0.9 - l_{\mathsf{A}(a)} = 0.1 + u_{\mathsf{B}(a)} = \mathrm{P}(\mathsf{B}(a))$. This happens for infinitely many values of $l_{\mathsf{A}(a)}$ and $u_{\mathsf{B}(a)}$. ■

A solution of (8) also contains information that can be used to simplify (7): If $u_t > 0$ for some solution to (8), then there are no solutions with $\ell_t > 0$, and vice versa. That is, we know the only possible direction of change for each tuple.

Lemma 15. *For solutions* $\mathrm{P}^1, \mathrm{P}^2$ *of* (8) *and all* $\langle t : p\rangle \in \mathcal{P}$, $\mathrm{P}^1(t) - p > 0$ *implies* $\mathrm{P}^2(t) - p \geq 0$.

This helps with the problematic inequalities $\mathrm{P}(t) \geq p - \ell_t$, $\mathrm{P}(t) \leq p + u_t$ in (7), for which we do not know which of them will be *active*, i.e., satisfied as an equality, which makes this problem much harder (Kazama and Tsujii 2005). However, once we have a solution P of (8), Lemma 15 allows us to do the following, for each $\langle t : p\rangle \in \mathcal{P}$: If $\mathrm{P}(t) - p > 0$, then we fix ℓ_t to 0 since no solution can yield $\mathrm{P}(t) < p$. Moreover, every solution then has to satisfy $\mathrm{P}(t) = p + u_t$, i.e., we can replace two inequalities from (7) with one *equality*. Dual arguments apply, if $\mathrm{P}(t) - p < 0$, where we get $\mathrm{P}(t) = p - \ell_t$. In the following, we write this equality as $\mathrm{P}(t) = p + x_t$, where x_t is either u_t or $-\ell_t$, depending on which case applies.

Unfortunately, a solution with $\mathrm{P}(t) - p = 0$ does not contain information about whether we need ℓ_t or u_t. However, for each such tuple $\langle t : p\rangle \in \mathcal{P}$, we can solve (8) two more times (with the previous simplifications) to see whether there is another solution with either $\mathrm{P}(t) \geq \delta$ or $\mathrm{P}(t) \leq \delta$ (where δ is very small) and the same ε-value. We then set $x_t \coloneqq u_t$ or $x_t \coloneqq -\ell_t$, respectively. If neither case applies, then we know that $\mathrm{P}(t)$ must remain equal to p, and we set $x_t \coloneqq 0$.

Solving the ME Problem. We thus obtain a simplified version of (7), where we can replace $\mathrm{P}(t) = [p - \ell_t, p + u_t]$ with $\mathrm{P}(t) = p + x_t$, where each x_t is either u_t, $-\ell_t$, or 0 (with $u_t, \ell_t \geq 0$). The usual method to solve such a problem is to find Lagrange multipliers λ_0, λ_ε, λ_t for all $\langle t : p\rangle \in \mathcal{P}$, and $\lambda_{\mathcal{W}}$ for all inconsistent worlds $\mathcal{W}$, where the constraints of (7) are satisfied and the gradient of the following expression vanishes (Jaynes 1957):

$$\begin{aligned} &-\Big(\sum_{\mathcal{W}} \mathrm{P}(\mathcal{W}) \log \mathrm{P}(\mathcal{W})\Big) - \lambda_0\Big(\sum_{\mathcal{W}} \mathrm{P}(\mathcal{W}) - 1\Big)\\ &-\sum_{\langle t:p\rangle \in \mathcal{P}} \lambda_t\Big(\Big(\sum_{\mathcal{W} \mid t \in \mathcal{W}} \mathrm{P}(\mathcal{W})\Big) - (p + x_t)\Big)\\ &-\sum_{\mathcal{W} \cup \mathcal{T} \vDash \bot} \lambda_{\mathcal{W}}\, \mathrm{P}(\mathcal{W}) - \lambda_\varepsilon\Big(\Big(\sum_{\langle t:p\rangle \in \mathcal{P}} |x_t|\Big) - \varepsilon\Big). \end{aligned}$$

Solving these equations, we obtain the expression

$$\mathrm{P}(\mathcal{W}) = \tfrac{1}{Z} \exp\Big(\sum_{t \in \mathcal{W}} \lambda_t - \lambda_{\mathcal{W}}\Big),$$

where the normalization factor Z incorporates λ_0 and λ_ε, and $\lambda_{\mathcal{W}} = 0$, if $\mathcal{W}$ is consistent with $\mathcal{T}$ (Jaynes 1957).

The parameters $\lambda_{\mathcal{W}}$ of inconsistent worlds $\mathcal{W}$ end up being ∞, forcing their probability to $\exp(-\infty) = 0$, in accordance with the constraints. Unfortunately, the number of the

parameters $\lambda_{\mathcal{W}}$ is exponential in the size of $\mathcal{K}$. However, the number of variables of the optimization problem is already exponential (for each world $\mathcal{W}$, we have to find $\mathrm{P}(\mathcal{W})$). Thus, the constraints $\mathrm{P}(\mathcal{W}) = 0$ actually help to solve the optimization problem faster, since some variables can be fixed to 0 from the beginning. The drawback is that we have to check the inconsistency ($\mathcal{W} \cup \mathcal{T} \models \bot$) of all worlds $\mathcal{W}$.

Then, the probability of a *consistent* world $\mathcal{W}$ (w.r.t. $\mathcal{T}$) under the ME distribution can be written as

$$\mathrm{P}(\mathcal{W}) = \tfrac{1}{Z} \exp\Big(\sum_{t \in \mathcal{W}} \lambda_t \Big), \tag{9}$$

i.e., we obtain a log-linear PKB with weights $w_t := \lambda_t$.

To compute the parameters λ_0 and λ_t, and hence the full probability distribution, one can use gradient-based approaches, e.g., the LMVM method, as described in (Malouf 2002). This is not a trivial task, and in general requires exponential time in the size of $\mathcal{K}$, but it is feasible to implement.

Note that this is only one way to obtain the weights for $\mathcal{D}_w$ from an existing PDB. It is straightforward to adapt this approach to the standard renormalization technique employed in PDBs. One could also integrate the parameter ε into the objective function, to achieve a tradeoff between entropy and slack, but more research is need on how precisely this should be done, and how to solve the resulting optimization problem.

8 Discussion and Related Work

There is a vast literature on combinations of logic and probability. Apart from the above-mentioned semantic differences, the main difference to previous proposals lies in the *methods* we propose. In Section 5.1, we have shown how to reduce a log-linear PKB to an MLN. Thus, we can use existing reasoners (Niu et al. 2011; Domingos and Lowd 2009) while incorporating the open-domain assumption. This shows that ontology-based rewriting techniques can augment MLNs with open-domain existential quantifiers essentially for free. In Section 5.2, we have described another reduction to ontology-mediated querying over PDBs that allows us to employ existing rewritability notions and algorithms for PDBs. In Section 7, we have described how to convert PDBs into our log-linear models. In spite of clear connections to previous research, none of the constructions in this paper have been considered before. We discuss the most closely related proposals, and clarify the differences to our approach.

MLNs. For statistical relational models, the closest work is MLNs (Richardson and Domingos 2006), where our motivation stems from. Our approach differs from MLNs in our use of ontological reasoning, in particular using the open-domain assumption for entailment. This is beyond the capabilities of MLNs under the CDA (see Example 1), which motivated the study of MLNs over infinite domains (Singla and Domingos 2007). That work also differs from ours since it allows *only universal quantifiers* to range over an infinite domain. Inference in MLNs (and in PDBs) can be translated to *weighted model counting* (Van den Broeck and Suciu 2017), which has recently been extended towards open-domains, but this is also restricted to a universal fragment of FOL (Belle 2017).

PLP. Function-free PLP is very common, and it is based on the CDA. PLP with function symbols is Turing-complete (De Raedt, Kimmig, and Toivonen 2007), and hence strong conditions need to be imposed on the programs to keep the distributions well-behaved, i.e., essentially finite (Sato and Kameya 1997). This problem also appears in *probabilistic programming*; for example, the programming language BLOG (Milch et al. 2005) allows reasoning over open domains, but defines rather strong restrictions on the language.

Log-Linear Probabilistic Models. Log-linear models have a strong theoretical representation as solutions to maximum-entropy problems (Bacchus et al. 1996; Potyka and Thimm 2017; Kuželka et al. 2018). We exploit this connection in Section 7, and obtain weights from the probabilities given in PDBs using an ME formulation. The principle of ME embodies several commonsense reasoning principles (Paris 1998): insensitivity to renaming, indifference to irrelevant information, and the assumption of independence in the absence of explicit information to the contrary (Paskin 2001).

Ontologies. The most closely related work in probabilistic ontologies is given in (Niepert, Noessner, and Stuckenschmidt 2011), where the description logic $\mathcal{EL}^{++}$ is extended with log-linear distributions. Less closely related is the work (Gottlob et al. 2013), which combines an ontology with an MLN, and the work (Peñaloza and Potyka 2016) that combines linear probabilistic constraints with description logics. Our approach is more general than these in the sense that our results apply to a class of ontology languages. Most importantly, we present reducibility results to MLNs and PDBs (see Section 5), which are novel.

Previous combinations of PDBs with ontologies are given in (Jung and Lutz 2012; Borgwardt, Ceylan, and Lukasiewicz 2017). The difference to our approach lies in the treatment of inconsistency (see Example 2). These models employ standard renormalization, whereas we follow a more fine-grained approach (see Section 7). More general models than the ones based on tuple-independent PDBs (Ceylan, Lukasiewicz, and Peñaloza 2016; Ceylan and Peñaloza 2017) additionally allow to encode conditional dependencies, but also differ from our approach in the handling of inconsistency.

9 Summary and Outlook

We introduced a new data model for ontology-mediated query answering over probabilistic data, and compared this model to existing proposals. Since reasoning in our model can be reduced to inference in MLNs or PDBs, we obtain a host of complexity results. We described an approach to learn the weights in our model, based on the principle of ME, to establish the connection of the new model with existing PKBs. We leave as future work an implementation, combining existing gradient-based optimization methods with efficient rewriting techniques and PDB or MLN inference engines.

Acknowledgments

This work was supported by the German Research Foundation (DFG) within the project BA 1122/19-1 (GOASQ), by The Alan Turing Institute under the UK EPSRC grant EP/N510129/1, and by the EPSRC grants EP/R013667/1, EP/L012138/1, and EP/M025268/1.

References

Baader, F.; Calvanese, D.; McGuinness, D. L.; Nardi, D.; and Patel-Schneider, P. F., eds. 2007. *The Description Logic Handbook*.

Bacchus, F.; Grove, A. J.; Halpern, J. Y.; and Koller, D. 1996. From statistical knowledge bases to degrees of belief. *AIJ* 87(1):75–143.

Belle, V. 2017. Open-universe weighted model counting. In *AAAI*.

Bienvenu, M., and Ortiz, M. 2015. Ontology-mediated query answering with data-tractable description logics. In *Reasoning Web*.

Borgwardt, S.; Ceylan, İ. İ.; and Lukasiewicz, T. 2017. Ontology-mediated queries for probabilistic databases. In *AAAI*.

Borgwardt, S.; Ceylan, İ. İ.; and Lukasiewicz, T. 2018. Recent advances in querying probabilistic knowledge bases. In *IJCAI-ECAI*.

Calì, A.; Gottlob, G.; and Kifer, M. 2013. Taming the infinite chase: Query answering under expressive relational constraints. *JAIR* 48:115–174.

Calì, A.; Gottlob, G.; and Lukasiewicz, T. 2012. A general Datalog-based framework for tractable query answering over ontologies. *J. Web Sem.* 14:57–83.

Calì, A.; Gottlob, G.; and Pieris, A. 2012. Towards more expressive ontology languages: The query answering problem. *AIJ* 193:87–128.

Ceylan, İ. İ., and Peñaloza, R. 2017. The Bayesian ontology language $\mathcal{BEL}$. *J. Autom. Reas.* 58(1):67–95.

Ceylan, İ. İ.; Borgwardt, S.; and Lukasiewicz, T. 2017. Most probable explanations for probabilistic database queries. In *IJCAI*.

Ceylan, İ. İ.; Darwiche, A.; and Van den Broeck, G. 2016. Open-world probabilistic databases. In *KR*.

Ceylan, İ. İ.; Lukasiewicz, T.; and Peñaloza, R. 2016. Complexity results for probabilistic Datalog$^{\pm}$. In *ECAI*.

Ceylan, İ. İ. 2017. *Query Answering in Probabilistic Data and Knowledge Bases*. Doctoral thesis, TU Dresden.

Dantsin, E.; Eiter, T.; Gottlob, G.; and Voronkov, A. 2001. Complexity and expressive power of logic programming. *ACM Comput. Surv.* 33(3):374–425.

De Raedt, L.; Kimmig, A.; and Toivonen, H. 2007. ProbLog: A probabilistic Prolog and its application in link discovery. In *IJCAI*.

Domingos, P., and Lowd, D. 2009. *Markov Logic: An Interface Layer for Artificial Intelligence*. Morgan & Claypool.

Dong, X.; Gabrilovich, E.; Heitz, G.; Horn, W.; Lao, N.; Murphy, K.; Strohmann, T.; Sun, S.; and Zhang, W. 2014. Knowledge Vault: A Web-scale approach to probabilistic knowledge fusion. In *SIGKDD*.

Eiter, T.; Ortiz, M.; Šimkus, M.; Tran, T.-K.; and Xiao, G. 2012. Query rewriting for Horn-$\mathcal{SHIQ}$ plus rules. In *AAAI*.

Fagin, R.; Kolaitis, P. G.; Miller, R. J.; and Popa, L. 2005. Data exchange: semantics and query answering. *TCS* 336(1):89–124.

Gaggl, S. A.; Rudolph, S.; and Schweizer, L. 2016. Fixed-domain reasoning for description logics. In *ECAI*.

Getoor, L., and Taskar, B. 2007. *Introduction to Statistical Relational Learning*. The MIT Press.

Gottlob, G., and Schwentick, T. 2012. Rewriting ontological queries into small nonrecursive datalog programs. In *KR*.

Gottlob, G.; Lukasiewicz, T.; Martinez, M. V.; and Simari, G. I. 2013. Query answering under probabilistic uncertainty in Datalog$^{\pm}$ ontologies. *AMAI* 69(1):37–72.

Gribkoff, E., and Suciu, D. 2016. Slimshot: In-database probabilistic inference for knowledge bases. *VLDBE* 9(7):552–563.

Gribkoff, E.; Van den Broeck, G.; and Suciu, D. 2014. The most probable database problem. In *BUDA*.

Halpern, J. Y. 2003. *Reasoning about uncertainty*. MIT Press.

Hansen, P.; Jaumard, B.; Nguetse, G.-B. D.; and De Aragao, M. P. 1995. Models and algorithms for probabilistic and Bayesian logic. In *IJCAI*.

Jaeger, M. 1997. Relational Bayesian Networks. In *UAI*.

Jaynes, E. T. 1957. Information theory and statistical mechanics. *Phys. Rev.* 106:620–630.

Jung, J. C., and Lutz, C. 2012. Ontology-based access to probabilistic data with OWL QL. In *ISWC*.

Kazama, J., and Tsujii, J. 2005. Maximum entropy models with inequality constraints: A case study on text categorization. *ML* 60(1):159–194.

Kuželka, O.; Wang, Y.; Davis, J.; and Schockaert, S. 2018. Relational marginal problems: Theory and estimation. In *AAAI*.

Malouf, R. 2002. A comparison of algorithms for maximum entropy parameter estimation. In *COLING*, 1–7.

Milch, B.; Marthi, B.; Russell, S.; Sontag, D.; Ong, D. L.; and Kolobov, A. 2005. Blog: Probabilistic models with unknown objects. In *IJCAI*.

Mitchell *et al.*, T. 2015. Never-ending learning. In *AAAI*.

Niepert, M.; Noessner, J.; and Stuckenschmidt, H. 2011. Log-linear description logics. In *IJCAI*.

Niu, F.; Ré, C.; Doan, A.; and Shavlik, J. 2011. Tuffy: Scaling up statistical inference in Markov Logic Networks using an RDBMS. *PVLDB* 4(6):373–384.

Paris, J. 1998. Common sense and maximum entropy. *Synthese* 117(1):75–93.

Paskin, M. A. 2001. Maximum-entropy probabilistic logics. Technical Report UCB/CSD-01-1161.

Peñaloza, R., and Potyka, N. 2016. Probabilistic reasoning in the description logic $\mathcal{ALCP}$ with the principle of maximum entropy. In *SUM*.

Potyka, N., and Thimm, M. 2017. Inconsistency-tolerant reasoning over linear probabilistic knowledge bases. *IJAR* 88:209–236.

Richardson, M., and Domingos, P. 2006. Markov logic networks. *ML* 62(1):107–136.

Sato, T., and Kameya, Y. 1997. PRISM: A language for symbolic-statistical modeling. In *IJCAI*, 1330–1335.

Shin, J.; Wu, S.; Wang, F.; De Sa, C.; Zhang, C.; and Ré, C. 2015. Incremental knowledge base construction using DeepDive. *PVLDB* 8(11):1310–1321.

Singla, P., and Domingos, P. 2007. Markov logic in infinite domains. In *UAI*.

Suciu, D.; Olteanu, D.; Ré, C.; and Koch, C. 2011. *Probabilistic Databases*. Morgan & Claypool.

Van den Broeck, G., and Suciu, D. 2017. Query processing on probabilistic data: A survey. *FTD* 7(3/4):197–341.

Van den Broeck, G.; Meert, W.; and Darwiche, A. 2014. Skolemization for weighted first-order model counting. In *KR*.

Wallace, M. 1993. Tight, consistent, and computable completions for unrestricted logic programs. *JLP* 15:243–273.

A Proofs

Proof of Theorem 6

Let $\mathcal{M}$ consist of the weighted facts from $\mathcal{D}_w$, and all sentences $\Phi \in \mathcal{T}$ with weight ∞. It is enough to show that, for all worlds $\mathcal{W}$, we have $\mathcal{W} \cup \mathcal{T} \models \bot$ (under the open-domain assumption) iff for some $\forall \mathbf{x}\, \phi(\mathbf{x}) \in \mathcal{T}$ there is a ground instance $\phi(\mathbf{c})$ that is not satisfied in $\mathcal{W}$. The "if"-direction is trivial. Assume now that all ground instances of the formulas in $\mathcal{T}$ are satisfied in $\mathcal{W}$. Then $\mathcal{W}$ itself can be seen as a (finite) first-order interpretation that satisfies $\mathcal{W} \cup \mathcal{T}$, which shows that $\mathcal{W} \cup \mathcal{T}$ is consistent. □

Proof of Theorem 7

A *(safe) Datalog rule* is a first-order formula of the form $A_1 \wedge \cdots \wedge A_n \rightarrow B$, where $A_1, \ldots, A_n, B$ are atoms over $\mathbf{R}$ and additional predicates (so-called *IDB predicates*), such that B uses an IDB predicate and only variables that also occur in $A_1, \ldots, A_n$; all variables are implicitly universally quantified. A *Datalog query* Q is a set of Datalog rules with a distinguished (0-ary) *goal (IDB) predicate* G_Q, and it is satisfied by a world $\mathcal{W}$ if $\mathcal{W} \cup \mathsf{Q} \models G_\mathsf{Q}$; equivalently, if the *minimal Herbrand model* of $\mathcal{W} \cup \mathsf{Q}$ satisfies G_Q. This is essentially closed-domain entailment, since Datalog rules cannot introduce new objects.

Let now $\mathsf{Q}_\mathcal{T}$ be a Datalog rewriting of Q w.r.t. $\mathcal{T}$ with goal predicate G_Q, and similarly for $\bot_\mathsf{Q}$ and $G_\bot$. To prove the claim, we construct an MLN $\mathcal{M}$ with $\mathrm{P}_\mathcal{K}(\mathsf{Q}) = \mathrm{P}_\mathcal{M}(G_\mathsf{Q})$, in polynomial time. In principle, we want to construct $\mathcal{M}$ by simply viewing all rules in $\mathsf{Q}_\mathcal{T}$ and $\bot_\mathcal{T}$ as hard constraints. The problem is that these Datalog queries may contain additional (IDB) predicates, which means that $\mathcal{M}$ has "larger" worlds than $\mathcal{K}$. Intuitively, we additionally need to restrict these extended worlds $\mathcal{W}'$ over the signature of $\mathcal{M}$ in such a way that, for each original world $\mathcal{W}$ over $\mathcal{K}$, there is a *unique consistent extension* $\mathcal{W}'$, which corresponds exactly to the minimal Herbrand model of $\mathcal{W} \cup \mathsf{Q}_\mathcal{T} \cup \bot_\mathcal{T}$. Then, we can restrict this world to be inconsistent iff it contains $G_\bot$ (by including the hard constraint $G_\bot \rightarrow \bot$), and let it have the same probability as the original world $\mathcal{W}$.

Fortunately, there exists a first-order theory $\mathcal{T}'$ such that $\mathcal{W} \cup \mathcal{T}'$ has exactly one Herbrand model (corresponding to one of our extended worlds $\mathcal{W}'$), namely the minimal Herbrand model of $\mathcal{W} \cup \mathsf{Q}_\mathcal{T} \cup \bot_\mathcal{T}$. This theory, called the *tight completion* of the Datalog program $\mathsf{Q}_\mathcal{T} \cup \bot_\mathcal{T}$ (Wallace 1993) is based on the idea of Clark's completion, which essentially replaces the implication ($\rightarrow$) in Datalog rules by equivalence ($\leftrightarrow$). However, since this is only correct for *non-recursive* Datalog programs, additional care needs to be taken for recursive dependencies. For this purpose, $\mathcal{T}'$ needs access to the theory of natural numbers with the successor predicate. Fortunately, since the maximal recursion depth is bounded polynomially in the size of $\mathbf{R}$ and $\mathbf{C}$, we need only finitely many natural numbers. One can now either view the (finite) successor predicate as a *built-in* predicate of the MLN system, or express it via nonrecursive Datalog rules, assuming only a total order on the constants $\mathbf{C}$ to be given, as done in (Gottlob and Schwentick 2012).

Let now $\mathcal{M}$ be the MLN that consists of the weighted facts from $\mathcal{D}_w$, all sentences from the tight completion $\mathcal{T}'$ of $\mathsf{Q}_\mathcal{T} \cup \bot_\mathcal{T}$ with weight ∞, and the additional hard constraint $G_\bot \rightarrow \bot$. Then, for every world $\mathcal{W}$ over $\mathcal{K}$, there is a unique extension of $\mathcal{W}$ to a world $\mathcal{W}'$ over $\mathcal{M}$ (including all ground instances of IDB predicates over $\mathbf{C}$), which corresponds to the minimal Herbrand model of $\mathcal{W} \cup \mathsf{Q}_\mathcal{T} \cup \bot_\mathcal{T}$; all other extensions of $\mathcal{W}$ are inconsistent w.r.t. the hard constraints, and thus have probability 0 in $\mathcal{M}$. Now, $\mathcal{W}'$ contains $G_\bot$ iff $\mathcal{W} \cup \mathsf{Q}_\mathcal{T} \cup \bot_\mathcal{T} \models G_\bot$ iff $\mathcal{W} \cup \mathcal{T} \models \bot$ (the Datalog rules in $\mathsf{Q}_\mathcal{T}$ are irrelevant here, since they can be assumed to use a disjoint set of IDB predicates). That is, worlds that are inconsistent in $\mathcal{K}$ have no consistent extensions in $\mathcal{M}$. This also means that we always have $\mathrm{P}_\mathcal{M}(\mathcal{W}') = \mathrm{P}_\mathcal{K}(\mathcal{W})$ since the equations (5) and (6) are based on the same weights (and hence the same normalization factor Z).

For consistent worlds $\mathcal{W}$, we similarly get that $\mathcal{W}'$ contains G_Q iff $\mathcal{W} \cup \mathsf{Q}_\mathcal{T} \cup \bot_\mathcal{T} \models G_\mathsf{Q}$ iff $\mathcal{W} \cup \mathcal{T} \models \mathsf{Q}$. Hence,

$$\mathrm{P}_\mathcal{M}(G_\mathsf{Q}) = \sum_{\mathcal{W}' \models G_\mathsf{Q}} \mathrm{P}_\mathcal{M}(\mathcal{W}') = \sum_{\mathcal{W} \cup \mathcal{T} \models \mathsf{Q}} \mathrm{P}_\mathcal{K}(\mathcal{W}) = \mathrm{P}_\mathcal{K}(\mathsf{Q}),$$

which concludes the proof. □

Proof of Theorem 8

We reduce a PKB $\mathcal{K} = (\mathcal{D}_w, \mathcal{T})$ to a PDB $\mathcal{P}$ that contains all weighted tuples t from $\mathcal{D}_w$ with probability $\frac{\exp(w_t)}{(1+\exp(w_t))}$. This accounts for the presence of $\exp(w_t)$ in the computation of $\mathrm{P}_\mathcal{K}(\mathsf{Q})$, and at the same time deals with the absence of $1 - \exp(w_t)$ in (6) compared to (4) (see also (Gribkoff and Suciu 2016)). Moreover, PDBs and log-linear PKBs differ in the closed-*world* assumption. Hence, we also need to add all tuples t that do not occur in $\mathcal{D}_w$ with the neutral probability 0.5 to $\mathcal{P}$. In data complexity, there are only polynomially many such tuples. If we define $w_t := 0$ for all tuples t that do not occur in $\mathcal{K}$, then we also have $\frac{\exp(w_t)}{1+\exp(w_t)} = 0.5 = \mathrm{P}_\mathcal{P}(t)$ for these tuples. Thus, we obtain

$$\begin{aligned}
\mathrm{P}_\mathcal{P}(\bot, \mathcal{T}) &= \sum_{\mathcal{W} \cup \mathcal{T} \models \bot} \prod_{t \in \mathcal{W}} \tfrac{\exp(w_t)}{1+\exp(w_t)} \cdot \prod_{\neg t \in \mathcal{W}} \tfrac{1}{1+\exp(w_t)} \\
&= \Big(\prod_t \tfrac{1}{1+\exp(w_t)} \Big) \cdot \sum_{\mathcal{W} \cup \mathcal{T} \models \bot} \prod_{t \in \mathcal{W}} \exp(w_t) \\
&= \Big(\prod_t \tfrac{1}{1+\exp(w_t)} \Big) Z',
\end{aligned}$$

where Z' is the "dual" of the normalization factor in (9):

$$Z = \sum_{\mathcal{W} \cup \mathcal{T} \not\models \bot} \prod_{t \in \mathcal{W}} \exp(w_t) \quad \text{and} \quad Z' = \sum_{\mathcal{W} \cup \mathcal{T} \models \bot} \prod_{t \in \mathcal{W}} \exp(w_t).$$

Moreover, we have that

$$\prod_t (1 + \exp(w_t)) = \sum_\mathcal{W} \prod_{t \in \mathcal{W}} \exp(w_t) = Z + Z',$$

and thus $\mathrm{P}_{\mathcal{P}}(\bot, \mathcal{T}) = \frac{Z'}{(Z+Z')}$. We similarly obtain

$$\begin{aligned}\mathrm{P}^{\mathsf{n}}_{\mathcal{P}}(\mathsf{Q}, \mathcal{T}) &= \frac{\mathrm{P}_{\mathcal{P}}(\mathsf{Q}, \mathcal{T}) - \mathrm{P}_{\mathcal{P}}(\bot, \mathcal{T})}{1 - \mathrm{P}_{\mathcal{P}}(\bot, \mathcal{T})} \\ &= \frac{\frac{1}{Z+Z'} \cdot \big(\sum_{\mathcal{W} \cup \mathcal{T} \models \mathsf{Q}} \prod_{t \in \mathcal{W}} \exp(w_t)\big) - \frac{Z'}{Z+Z'}}{1 - \frac{Z'}{Z+Z'}} \\ &= \frac{\big(\sum_{\mathcal{W} \cup \mathcal{T} \models \mathsf{Q}} \prod_{t \in \mathcal{W}} \exp(w_t)\big) - Z'}{(Z + Z') - Z'}.\end{aligned}$$

Since $\mathcal{W} \cup \mathcal{T} \models \bot$ implies $\mathcal{W} \cup \mathcal{T} \models \mathsf{Q}$, by (9) we have

$$\begin{aligned}&\sum_{\mathcal{W} \cup \mathcal{T} \models \mathsf{Q}} \prod_{t \in \mathcal{W}} \exp(w_t) \\ &= \Big(\sum_{\mathcal{W} \cup \mathcal{T} \not\models \bot, \mathcal{W} \cup \mathcal{T} \models \mathsf{Q}} \prod_{t \in \mathcal{W}} \exp(w_t)\Big) + \Big(\sum_{\mathcal{W} \cup \mathcal{T} \models \bot} \prod_{t \in \mathcal{W}} \exp(w_t)\Big) \\ &= Z \cdot \mathrm{P}_{\mathcal{K}}(\mathsf{Q}) + Z',\end{aligned}$$

and hence we conclude that

$$\mathrm{P}^{\mathsf{n}}_{\mathcal{P}}(\mathsf{Q}, \mathcal{T}) = \tfrac{Z \cdot \mathrm{P}_{\mathcal{K}}(\mathsf{Q})}{Z} = \mathrm{P}_{\mathcal{K}}(\mathsf{Q}),$$

i.e., the query probability under $\mathrm{P}_{\mathcal{K}}$ coincides with the normalized OMQ probability under $\mathrm{P}_{\mathcal{P}}$.

For the other direction, let $\mathcal{P}$ be a PDB, $\mathcal{T}$ a theory, and construct a log-linear PKB $\mathcal{K} = (\mathcal{D}_w, \mathcal{T})$ by assigning each tuple t the weight $w_t := \log(\mathrm{P}_{\mathcal{P}}(t)) - \log(1 - \mathrm{P}_{\mathcal{P}}(t))$. Each tuple that is not present in $\mathcal{P}$ gets probability 0, and hence weight $-\infty$. Hence, we again have $\frac{\exp(w_t)}{1+\exp(w_t)} = 0.5 = \mathrm{P}_{\mathcal{P}}(t)$, and the above arguments show that the query probability in $\mathcal{K}$ is the same as the normalized OMQ probability in $\mathcal{P}$. □

Proof of Theorem 9

Consider the PKB $\mathcal{K}$ and PDB $\mathcal{P}$ from Theorem 8. It suffices to show that $\mathrm{P}_{\mathcal{K}}(\mathcal{W}) > \mathrm{P}_{\mathcal{K}}(\mathcal{W}')$ iff $\mathrm{P}_{\mathcal{P}}(\mathcal{W}) > \mathrm{P}_{\mathcal{P}}(\mathcal{W}')$, for all worlds $\mathcal{W}, \mathcal{W}'$ that are consistent with $\mathcal{T}$ and satisfy the query Q. Recall that

$$\mathrm{P}_{\mathcal{P}}(\mathcal{W}) = \tfrac{1}{Z+Z'} \prod_{t \in \mathcal{W}} \exp(\lambda_t),$$

which is related to $\mathrm{P}_{\mathcal{K}}(\mathcal{W})$ by a factor of $\frac{Z}{Z+Z'}$ (see (9)). Hence, if $\mathrm{P}_{\mathcal{K}}(\mathcal{W}) > \mathrm{P}_{\mathcal{K}}(\mathcal{W}')$, then

$$\mathrm{P}_{\mathcal{P}}(\mathcal{W}) = \tfrac{Z}{Z+Z'}\,\mathrm{P}_{\mathcal{K}}(\mathcal{W}) > \tfrac{Z}{Z+Z'}\,\mathrm{P}_{\mathcal{K}}(\mathcal{W}') = \mathrm{P}_{\mathcal{P}}(\mathcal{W}'),$$

and vice versa. □

Proof of Theorem 12

We can use Theorem 8 to obtain a PDB $\mathcal{P}$ with default probabilities. Since $\mathsf{Q}_{\mathcal{T}}$ is a first-order rewriting of Q w.r.t. $\mathcal{T}$, which is only correct in consistent worlds, $\mathrm{P}_{\mathcal{P}}(\mathsf{Q}_{\mathcal{T}})$ may not include the probability of all inconsistent worlds, which is why we consider $\mathrm{P}_{\mathcal{P}}(\mathsf{Q}_{\mathcal{T}} \vee \bot_{\mathcal{T}})$ instead, which captures all inconsistent worlds in addition to those that satisfy $\mathsf{Q}_{\mathcal{T}}$ (i.e., entail Q w.r.t. $\mathcal{T}$). Similarly, $\mathrm{P}_{\mathcal{P}}(\bot_{\mathcal{T}})$ is equal to $\mathrm{P}_{\mathcal{P}}(\bot, \mathcal{T})$.

By Theorem 8, we can thus evaluate the following expression using the lifted algorithm from (Ceylan, Darwiche, and Van den Broeck 2016), in order to compute $\mathrm{P}_{\mathcal{K}}(\mathsf{Q})$:

$$\frac{\mathrm{P}_{\mathcal{P}}(\mathsf{Q}_{\mathcal{T}} \vee \bot_{\mathcal{T}}) - \mathrm{P}_{\mathcal{P}}(\bot_{\mathcal{T}})}{1 - \mathrm{P}_{\mathcal{P}}(\bot_{\mathcal{T}})}.$$

□

Proof of Lemma 15

In a solution of (8), for each t one of the variables ℓ_t and u_t must be 0, and the other satisfies $\mathrm{P}(t) = p + u_t$ or $\mathrm{P}(t) = p - \ell_t$, respectively; that is, solutions to (8) are fully determined by the probability distribution P, and the objective function can be written as $\sum_{\langle t:p \rangle \in \mathcal{P}} |\mathrm{P}(t) - p|$.

Assume now that P^1 and P^2 are two solutions with optimal value ε, and $\mathrm{P}^1(t_0) - p_0 > 0$, but $\mathrm{P}^2(t_0) - p_0 < 0$, for some $\langle t_0 : p_0 \rangle \in \mathcal{P}$. We define another distribution P^λ by $\mathrm{P}^\lambda(\mathcal{W}) := \lambda\,\mathrm{P}^1(\mathcal{W}) + (1-\lambda)\,\mathrm{P}^2(\mathcal{W})$, which implies that $\mathrm{P}^\lambda(t) := \lambda\,\mathrm{P}^1(t) + (1-\lambda)\,\mathrm{P}^2(t)$ for all tuples $\langle t : p \rangle \in \mathcal{P}$. In particular, if we set $\lambda := \frac{p_0 - \mathrm{P}^2(t_0)}{\mathrm{P}^1(t_0) - \mathrm{P}^2(t_0)}$, then $\mathrm{P}^\lambda(t_0) = p_0$. Clearly, P^λ still satisfies the constraints of (8). The value of the objective function for P^λ is

$$\begin{aligned}\sum_{\langle t:p \rangle \in \mathcal{P}} |\mathrm{P}^\lambda(t) - p| &< \sum_{\langle t:p \rangle \in \mathcal{P}} \lambda|\mathrm{P}^1(t) - p| + (1-\lambda)|\mathrm{P}^2(t) - p| \\ &= \lambda\varepsilon + (1-\lambda)\varepsilon = \varepsilon\end{aligned}$$

since $|\mathrm{P}^\lambda(t_0) - p_0| = 0 < \lambda|\mathrm{P}^1(t_0) - p_0| + (1-\lambda)|\mathrm{P}^2(t_0) - p_0|$ and $\leq$ holds for all other tuples. This contradicts the assumption that P^1 and P^2 are solutions of (8). □