Open-World Probabilistic Databases: An Abridged Report*

İsmail İlkan Ceylan
Faculty of Computer Science
Technische Universität Dresden, Germany
ismail.ceylan@tu-dresden.de

Adnan Darwiche and **Guy Van den Broeck**
Computer Science Department
University of California, Los Angeles
{darwiche,guyvdb}@cs.ucla.edu

Abstract

Large-scale probabilistic knowledge bases are becoming increasingly important in academia and industry alike. They are constantly extended with new data, powered by modern information extraction tools that associate probabilities with database tuples. In this paper, we revisit the semantics underlying such systems. In particular, the closed-world assumption of probabilistic databases, that facts not in the database have probability zero, clearly conflicts with their everyday use. To address this discrepancy, we propose an open-world probabilistic database semantics, which relaxes the probabilities of open facts to default intervals. For this open-world setting, we lift the existing data complexity dichotomy of probabilistic databases, and propose an efficient evaluation algorithm for unions of conjunctive queries. We also show that query evaluation can become harder for non-monotone queries.

1 Introduction

Driven by the need to learn from vast amounts of text data, efforts throughout natural language processing, information extraction, databases and AI are coming together to build *large-scale knowledge bases*. Academic systems such as NELL [Mitchell *et al.*, 2015], DeepDive [Sa *et al.*, 2017], Freebase [Bollacker *et al.*, 2008], and Yago [Hoffart *et al.*, 2013] continuously crawl the web to extract relational information. Industry projects such as Microsoft's Probase [Wu *et al.*, 2012], IBM's Watson [Ferrucci, 2012], or Google's Knowledge Vault [Dong *et al.*, 2014] similarly learn structured data from text. These systems have already populated their databases with millions of entities and billions of tuples.

Such knowledge bases are inherently probabilistic. To go from the raw text to structured data, information extraction systems employ a sequence of statistical machine learning techniques, from part-of-speech tagging until relation extraction [Mintz *et al.*, 2009]. For knowledge-base completion statistical relational learning algorithms make use of embeddings [Bordes *et al.*, 2011; Socher *et al.*, 2013] or probabilistic rules [Wang *et al.*, 2013; De Raedt *et al.*, 2015]. In both settings, the output is a predicted fact with its probability. Thus, we need to define a probabilistic semantics for such knowledge bases. The most-basic model is that of *tuple-independent probabilistic databases* (PDBs) [Suciu *et al.*, 2011], which indeed underlies many of these systems [Dong *et al.*, 2014; Sa *et al.*, 2017]. According to the PDB semantics, each database tuple is an independent Bernoulli random variable, and all other tuples have probability zero, enforcing a *closed-world assumption* (CWA) [Reiter, 1978].

This paper revisits the choice for the CWA in probabilistic knowledge bases. We observe that the *CWA is violated* in their deployment, which makes it *problematic to reason, learn, or mine* on top of these databases. We propose an alternative semantics for probabilistic knowledge bases to address these problems, based on the *open-world assumption (OWA)*, which as opposed to the CWA, does not presume that the knowledge of a domain is complete. Our proposal of *open-world probabilistic databases* (OpenPDBs) builds on the theory of imprecise probabilities [Levi, 1980] in the sense that all tuples that are not in the knowledge base, called *open tuples*, take on probabilities from a default *probability interval*. All facts in the open world remain possible, formalized through *lower* and *upper* probabilities, which determines their contribution to the probability of possible worlds. This framework provides more meaningful answers, in terms of upper and lower bounds on the query probability.

Our open-world semantics is supported by a *query evaluation algorithm* for *unions of conjunctive queries* (UCQs). This class of queries, corresponding to monotone DNF, is particularly well-behaved and the focal point of database research. Perhaps the largest appeal of PDBs comes from a breakthrough dichotomy result by [Dalvi and Suciu, 2012], perfectly delineating which UCQs can be answered efficiently in the size of the PDB. Their algorithm runs in polynomial time for all efficient queries, called *safe queries*, and recognizes all others to be #P-hard. Our OpenPDB algorithm extends the PDB algorithm of [Dalvi and Suciu, 2012] and inherits its elegant properties: all safe queries run in polynomial time. When our algorithm fails, the query is #P-hard.

In general, both OpenPDBs and PDBs admit the same *data complexity dichotomy between polynomial time and #P*. Moreover, a careful analysis shows that both algorithms run in *linear time* in the number of (closed-world) tuples on a *sorted database*. Even though OpenPDBs model a polynomially

*This is an abridged version of a paper that appeared as Ceylan *et al.* [2016] in the Proceedings of KR 2016.

Inmovie	
w_smith	ali
arquette	scream
pitt	mr_ms_smith
jolie	mr_ms_smith

Couple	
arquette	cox
pitt	jolie
pitt	aniston
kunis	kutcher

Figure 1: Database Tables.

Inmovie		P
w_smith	ali	0.9
j_smith	ali	0.6
arquette	scream	0.7
pitt	mr_ms_smith	0.5
jolie	mr_ms_smith	0.7

Couple		P
arquette	cox	0.6
pitt	jolie	0.8
thornton	jolie	0.6
pitt	aniston	0.9
kunis	kutcher	0.7

Figure 2: Probabilistic Database Tables

larger set of random variables, these can be reasoned about as a whole, and there is no computational blow-up for open-world reasoning. Therefore, assuming that the database constants are sorted, both OpenPDBs and PDBs admit a stronger *data complexity dichotomy between linear time and #P*. The fact that safe PDB queries have linear-time data complexity on a sorted database is perhaps not technically surprising, but this has not been observed in the literature before. This observation is quite important practically though, particularly in the context of open-world probabilistic databases.

We focus on the corresponding decision problem of probabilistic query evaluation and thus on the complexity class PP [Gill, 1977], which can be seen as a decision version of #P. Our analysis entails that the complexity of open-world reasoning can go up significantly with negation. We identify a safe PDB query that becomes NP-complete and an unsafe PDB query that becomes NP^{PP}-complete on OpenPDBs. We also consider query evaluation complexity in terms of the domain size, or equivalently, the size of the open world, keeping both the query and the database fixed. Here, complexities range from polynomial time to the unary alphabet class $\#\text{P}_1$.

2 Preliminaries

Relational Logic. We focus on the *function-free finite-domain* fragment of first-order logic (FOL). An *atom* $\mathtt{P}(t_1, \ldots, t_n)$ consists of predicate $\mathtt{P}/n$ of arity n followed by n arguments, which are either *constants* from a finite domain $\mathcal{D} = \{\mathsf{a}, \mathsf{b}, \ldots\}$ or *logical variables* $\{x, y, \ldots\}$. A *ground atom* does not contain logical variables. A *literal* is an atom or its negation. A *formula* combines atoms with logical connectives and quantifiers $\exists$ and $\forall$. A logical variable x is *quantified* if it is enclosed by a $\forall x$ or $\exists x$. A *free variable* is one that is not quantified. We write $\phi(x, y)$ to denote that x, y are free in ϕ. A formula is *monotone* if it contains no negations. A *substitution* $[x/t]$ replaces all occurrences of x by t in some formula Q, denoted $\mathsf{Q}[x/t]$.

A relational *vocabulary* σ consists of a set of predicates $\mathcal{R}$ and a domain $\mathcal{D}$. We will make use of *Herbrand semantics* [Hinrichs and Genesereth, 2006], as is customary. The *Herbrand base* of σ is the set of all ground atoms that can be constructed from $\mathcal{R}$ and $\mathcal{D}$. An *σ-interpretation* is a truth-value assignment to all the atoms in the Herbrand base of σ, called σ-atoms. An interpretation ω is a model of formula Q when it satisfies Q, defined in the usual way. Satisfaction is denoted by $\omega \models_\sigma \mathsf{Q}$. We omit σ when clear from context.

Databases and Queries. Following the standard model-theoretic view [Abiteboul *et al.*, 1995], a *relational database* for vocabulary σ is a σ-interpretation ω. Figure 1 depicts a relational database in terms of tables. Each table corresponds to a predicate and its rows correspond to ground atoms of that predicate, which are also called *records* or *facts*. These atoms are mapped to *true*, while ones not listed in these tables are mapped to *false,* according to the CWA [Reiter, 1978].

The fundamental task in databases is query answering. Given a formula $\mathsf{Q}(x, y, \ldots)$, the task is to find all substitutions (answers) $[x/s, y/t, \ldots]$ such that $\omega \models \mathsf{Q}[x/s, y/t, \ldots]$. Consider for example the query $\mathsf{Q}_1(x, y)$ for spouses that starred in the same movie:

$$\exists z,\ \mathtt{Inmovie}(x, z) \wedge \mathtt{Inmovie}(y, z) \wedge \mathtt{Couple}(x, y).$$

The database in Figure 1 yields $[x/\mathsf{pitt}, y/\mathsf{jolie}]$ as the only answer. This formula is an existentially quantified conjunction of atoms, called a *conjunctive query* (CQ). We concentrate on *Boolean conjunctive queries* (BQs), which have no free variables. Answers to BQs are either *true* or *false*. For example, the BQ

$$\mathsf{Q}_2 = \exists x, y, z\ \mathtt{Inmovie}(x, z) \wedge \mathtt{Inmovie}(y, z) \wedge \mathtt{Couple}(x, y),$$

returns *true* on the database in Figure 1. A Boolean *union of conjunctive queries* (UCQ) is a disjunction of BQs. We will denote the class of *UCQs with negation on atoms* by $\text{UC}\tilde{\text{Q}}$.

Probabilistic Databases. The simplest probabilistic database model is the one based on the tuple-independence assumption as we adopt here [Suciu *et al.*, 2011].

Definition 1. A *probabilistic database* (PDB) $\mathcal{P}$ for a vocabulary σ is a finite set of *tuples* of the form $\langle t : p \rangle$, where t is a σ-atom and $p \in [0, 1]$. Moreover, if $\langle t : p \rangle \in \mathcal{P}$ and $\langle t : q \rangle \in \mathcal{P}$, then $p = q$.

Figure 2 shows an example PDB. The following semantics is based on the *tuple-independence* assumption mentioned earlier [Suciu *et al.*, 2011].

Definition 2. A PDB $\mathcal{P}$ for vocabulary σ induces a *unique probability distribution* over σ-interpretations ω:

$$\text{P}_\mathcal{P}(\omega) = \prod_{t \in \omega} \text{P}_\mathcal{P}(t) \prod_{t \notin \omega} (1 - \text{P}_\mathcal{P}(t)),$$

where

$$\text{P}_\mathcal{P}(t) = \begin{cases} p & \text{if } \langle t : p \rangle \in \mathcal{P} \\ 0 & \text{otherwise.} \end{cases}$$

The choice of setting $\text{P}_\mathcal{P}(t) = 0$ for tuples missing from PDB $\mathcal{P}$ is a *probabilistic version* of the CWA.

Definition 3. The *probability of a BQ* Q w.r.t. a PDB $\mathcal{P}$ is

$$\text{P}_\mathcal{P}(\mathsf{Q}) = \sum_{\omega \models \mathsf{Q}} \text{P}_\mathcal{P}(\omega).$$

For example, considering the PDB in Figure 2 and the query Q_2 defined earlier, we get $\text{P}_\mathcal{P}(\mathsf{Q}_2) = 0.28$.

3 Closed-World PDBs in Practice

As noted by Reiter [1978], CWA presumes a complete knowledge about the domain being represented. We now assess the adequacy of CWA for probabilistic knowledge bases.

Truncating and Space Blow-up. Knowledge completion systems continuously crawl the web, and discover new facts. Nevertheless, they retain only a small fraction of the discovered facts in their knowledge base and facts with a probability below some predefined threshold are simply discarded, which is clearly a misuse of the CWA. As only facts with high-probability are stored in their knowledge base, most of the automatically constructed PDBs are hardly probabilistic. Most tuples have a very high probability, placing PDBs into an almost crisp setting in practice. This mode of operation, however, is not an oversight, but a necessity. It is simply not possible to retain all facts in the knowledge base. Consider, for instance the `Sibling` relation over a domain of 7 billion people. Storing a single-precision probability for all `Sibling` facts would require 196 exabytes of memory; two orders of magnitude more than the estimated capacity available to Google [Munroe, 2015].

Distinguishing Queries. Since the CWA is violated in most PDBs, several query answering issues become apparent. Consider, for instance, the queries $\mathsf{Q}_1(\mathsf{pitt},\mathsf{jolie})$ and Q_2. The former query entails the latter, leading us to expect that $\mathrm{P}(\mathsf{Q}_2) > \mathrm{P}(\mathsf{Q}_1(\mathsf{pitt},\mathsf{jolie}))$ in an open-world setting. However, $\mathrm{P}(\mathsf{Q}_2) = \mathrm{P}(\mathsf{Q}_1(\mathsf{pitt},\mathsf{jolie})) = 0.28$ in the PDB of Figure 2. For another example, consider the queries $\mathsf{Q}_1(\mathsf{w\_smith},\mathsf{j\_smith})$ and $\mathsf{Q}_1(\mathsf{thornton},\mathsf{aniston})$. The former is supported by two facts in the PDB of Figure 2, while the latter is supported by none, which should make it less likely. However, both evaluate to the probability 0. Taking these observations to the extreme, the query $\mathtt{Inmovie}(x,y) \wedge \neg\mathtt{Inmovie}(x,y)$ is unsatisfiable, yet it evaluates to the same probability as the satisfiable query $\mathsf{Q}_1(\mathsf{thornton},\mathsf{aniston})$. These counterintuitive results are not synthetic, but also observed in *real-world data*.

Knowledge-Base Completion and Mining. The CWA permeates higher-level tasks that one is usually interested in performing on probabilistic databases. For example, a natural approach to knowledge-base completion learns a probabilistic model from training data. Consider, for example, a probabilistic rule [Wang *et al.*, 2013; De Raedt *et al.*, 2015]

$$\mathtt{Costars}(x,y) \xleftarrow{0.8} \mathtt{Inmov}(x,z), \mathtt{Inmov}(y,z), \mathtt{Couple}(x,y).$$

To evaluate the quality of this rule for predicting the `Costars` relation, the standard approach would be to quantify the conditional likelihood of the rule based on training data [Sutton and McCallum, 2011]:

$$D = \{\mathtt{Costars}(\mathsf{w\_smith},\mathsf{j\_smith}), \mathtt{Costars}(\mathsf{pitt},\mathsf{jolie})\}.$$

The rule predicts $\mathrm{P}(\mathtt{Costars}(\mathsf{w\_smith},\mathsf{j\_smith})) = 0$, due to the CWA, since one fact is missing from the knowledge base. Hence, the rule gets a likelihood score of zero, regardless of its performance on other tuples in the training data. Another high-level task is to mine frequent patterns in the knowledge base; for example to find the pattern $\mathsf{Q}_1(x,y)$ and report it to the data miner. Again, due to the CWA, the frequencies of these patterns will be underestimated [Galárraga *et al.*, 2013].

4 Open-World Probabilistic Databases

Our proposal for open-world probabilistic databases is based on the assumption that facts not appearing in a probabilistic database have probabilities in the interval $[0,\lambda]$, for some threshold probability λ. This is formalized through a *credal set* [Levi, 1980], which is a set of probability distributions.

Definition 4. An *open probabilistic database* is a pair $\mathcal{G} = (\mathcal{P},\lambda)$, where $\mathcal{P}$ is a probabilistic database and $\lambda \in [0,1]$.

The semantics of an open probabilistic database (OpenPDB) is based on completing probabilistic databases.

Definition 5. A λ-*completion* of probabilistic database $\mathcal{P}$ is another probabilistic database obtained as follows. For each atom t that does not appear in $\mathcal{P}$, we add tuple $\langle t : p\rangle$ to $\mathcal{P}$ for some $p \in [0,\lambda]$.

While a closed probabilistic database induces a unique probability distribution, an open probabilistic database induces a set of probability distributions.

Definition 6. An open probabilistic database $\mathcal{G} = (\mathcal{P},\lambda)$ induces a set of probability distributions $\mathrm{K}_\mathcal{G}$ such that distribution P belongs to $\mathrm{K}_\mathcal{G}$ iff P is induced by some λ-completion of probabilistic database $\mathcal{P}$.

Note that the set of distributions $\mathrm{K}_\mathcal{G}$ is credal and represents the semantics of OpenPDB $\mathcal{G}$. Intuitively, an OpenPDB represents all possible ways to extend a PDB with new tuples from the open world, with the restriction that the probability of these unknown tuples can never be larger than λ. Thus, query evaluation for OpenPDB yields interval-based answers.

Definition 7. The *probability interval of a Boolean query* Q in OpenPDB $\mathcal{G}$ is $\mathrm{K}_\mathcal{G}(\mathsf{Q}) = [\underline{\mathrm{P}}_\mathcal{G}(\mathsf{Q}), \overline{\mathrm{P}}_\mathcal{G}(\mathsf{Q})]$, where

$$\underline{\mathrm{P}}_\mathcal{G}(\mathsf{Q}) = \min_{\mathrm{P}\in\mathrm{K}_\mathcal{G}} \mathrm{P}(\mathsf{Q}) \quad \text{and} \quad \overline{\mathrm{P}}_\mathcal{G}(\mathsf{Q}) = \max_{\mathrm{P}\in\mathrm{K}_\mathcal{G}} \mathrm{P}(\mathsf{Q}).$$

Reiter [1978] introduced the *open-world assumption* as the opposite of the CWA. Under the OWA, a set of tuples no longer corresponds to a single interpretation. Instead, a database corresponds to the set of interpretations that extend it. A similar effect is achieved by OpenPDBs: a set of probabilistic tuples no longer corresponds to a single distribution. Instead, a probabilistic database corresponds to the set of distributions that extend it. In restricting the probabilities of open tuples to lie in $[0,\lambda]$, OpenPDBs follow a rich literature on interval-based probabilities [Halpern, 2003], credal networks [Cozman, 2000] and default reasoning [Reiter, 1980].

5 OpenPDBs in Practice

We now discuss some implications of the open-world setting. Consider the queries Q_1 and Q_2 again. We have already noted that $\mathsf{Q}_1(\mathsf{pitt},\mathsf{jolie})$ entails Q_2, leading us to expect that $\mathrm{P}(\mathsf{Q}_2) > \mathrm{P}(\mathsf{Q}_1(\mathsf{pitt},\mathsf{jolie}))$ assuming our knowledge is not complete. This is indeed the case for OpenPDBs (for upper probabilities) since there are many worlds with non-zero probability that entail Q_2 but not $\mathsf{Q}_1(\mathsf{pitt},\mathsf{jolie})$. We have also observed that an unsatisfiable query is in some cases as likely as a satisfiable one in the closed world. In the open-world setting, the upper probability of a satisfiable query will be greater than the upper probability of an unsatisfiable query. In

fact, any unsatisfiable query will still have a zero upper probability in OpenPDBs. For some real-world examples, based on NELL, we refer the reader to [Ceylan *et al.*, 2016].

The Bayesian learning paradigm is a popular view on machine learning, where the learner maintains beliefs about the world as a probability distribution, and updates these beliefs based on data, to obtain a posterior distribution. In the context of knowledge base completion systems, we observe the following. Given a PDB at time t, such systems gather data D^t to obtain a new model $\mathrm{P}^{t+1}_{\mathcal{P}}(.) = \mathrm{P}^{t}_{\mathcal{P}}(.\,|\,D^t)$. Systems continuously add facts f, that is, set $\mathrm{P}^{t+1}_{\mathcal{P}}(f) > 0$, whereas previously $\mathrm{P}^{t}_{\mathcal{P}}(f) = 0$; an impossible induction for Bayesian learning. This problem is resolved by the open database semantics. Now, facts are not a priori impossible, and adding them does not conflict with the prior beliefs.

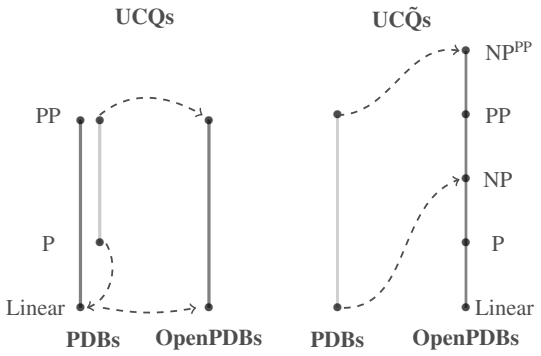

Figure 3: Complexity map for PDBs and OpenPDBs on a sorted database.

6 Query Evaluation in OpenPDBs

OpenPDBs model an infinite set of PDBs, and thus, it may seem like an unsurmountable task to efficiently compute intervals $\mathrm{K}_{\mathcal{G}}(\mathsf{Q})$. Fortunately, the problem is simplified by a strong property of credal sets as we employ them here: probability bounds always come from extreme points [Cozman, 2000]. For OpenPDBs, this means the following.

Definition 8. An *extreme distribution* $\mathrm{P} \in \mathrm{K}_{\mathcal{G}}$ is a distribution where $\mathrm{P}(t) = \overline{\mathrm{P}}_{\mathcal{G}}(t)$ or $\mathrm{P}(t) = \underline{\mathrm{P}}_{\mathcal{G}}(t)$ for all tuples t.

Proposition 9. *For any OpenPDB $\mathcal{G}$ and a query Q, there exist extreme distributions $\underline{\mathrm{P}}, \overline{\mathrm{P}} \in \mathrm{K}_{\mathcal{G}}$ such that $\underline{\mathrm{P}}(\mathsf{Q}) = \underline{\mathrm{P}}_{\mathcal{G}}(\mathsf{Q})$, and $\overline{\mathrm{P}}(\mathsf{Q}) = \overline{\mathrm{P}}_{\mathcal{G}}(\mathsf{Q})$.*

Thus, for query answering, it suffices to consider a finite set of distributions, which can be represented by λ-completions where the open-world tuples have an extreme probability. As for PDBs, query answering in OpenPDBs is computationally challenging. We first define the decision problem of interest.

Definition 10. Given an OPDB $\mathcal{G}$, query Q and probability p, the *upper (resp., lower) probabilistic query evaluation* problem is to decide whether $\overline{\mathrm{P}}_{\mathcal{G}}(\mathsf{Q}) \geq p$ (*resp.,* $\underline{\mathrm{P}}_{\mathcal{G}}(\mathsf{Q}) < p$).

Proposition 9 suggests a naive query answering algorithm: generate all extreme distributions P, compute $\mathrm{P}(\mathsf{Q})$, and report the minimum and maximum. This procedure will terminate in time exponential in the number of open-world tuples. For UCQs, however, the monotonicity of the queries allows us to further simplify query evaluation. We can choose the minimal (resp., maximal) bound for every tuple, which minimizes (resp., maximizes) the probability of the UCQ.

Theorem 11. *Given OpenPDB $\mathcal{G} = (\mathcal{P}, \lambda)$ and UCQ Q, let $\mathcal{P}' = \mathcal{P} \cup \{\langle t : \lambda \rangle \mid t \text{ is an open tuple}\}$ be a λ-completion. Then, $\mathrm{K}_{\mathcal{G}}(\mathsf{Q}) = [\mathrm{P}_{\mathcal{P}}(\mathsf{Q}), \mathrm{P}_{\mathcal{P}'}(\mathsf{Q})]$.*

Theorem 11 shows that we can reduce OpenPDB query evaluation to query evaluation on PDBs with only a polynomial blow-up in the data size. Unfortunately, this can be impractical for PDBs with a large domain. A more goal-oriented algorithm, described next, avoids this blow-up.

7 Overview of the Results and Outlook

Lifted Inference. OpenPDBs is supported with an efficient lifted inference algorithm $\mathbf{Lift}^{\mathbf{R}}_{\mathbf{O}}$, which is an adaptation of the $\mathbf{Lift}^{\mathbf{R}}$ algorithm presented in Gribkoff *et al.* [2014]. This algorithm assumes that the database is sorted and the query is ranked as in [Dalvi and Suciu, 2012] and can compute the probability of a query in time linear in the size of the data. The main difference between $\mathbf{Lift}^{\mathbf{R}}_{\mathbf{O}}$ and $\mathbf{Lift}^{\mathbf{R}}$ is in the treatment of the open tuples. Although OpenPDBs model a polynomially larger set of random variables, due to the symmetries, these can be reasoned about as a whole, and there is no computational blow-up for open-world reasoning.

Data Complexity. Our key complexity results are illustrated in Figure 3. Briefly, the decision problem of UCQ query evaluation either has linear time data complexity or is PP-complete on a sorted database, depending on the query. This implies a dichotomy between polynomial time and PP. Moreover, these complexities are the same for PDBs and OpenPDBs. For queries with negation, some safe PDBs queries can remain safe, and some can become NP-complete on OpenPDBs. Some UCQ̃s that are PP-complete on PDBs can remain PP-complete, and some can become $\mathrm{NP}^{\mathrm{PP}}$-complete.

Domain Complexity. The domain complexity of OpenPDB query evaluation is the complexity for a fixed query and database, as a function of the size of the domain $\mathcal{D}$. Beame *et al.* [2015] study this complexity in the context of a task called *weighted first-order model counting*. This task is reducible to OpenPDB query evaluation when the database is empty. We refer to Beame *et al.* [2015] for full details, but note that there exists an FO^3 query and set of probabilistic tuples with $\#\mathrm{P}_1$-complete domain complexity on OpenPDBs.

Outlook. A key challenge is to restrict the open world to exclude spurious possible worlds, and thus limit the probability mass of open tuples. One way of restricting the models is to pose constraints on the probability distributions. Another approach is to use an explicit formalism for restricting the models such as ontological rules [Borgwardt *et al.*, 2017].

Acknowledgements

We thank Lakshay Rastogi for his help. This work is partially supported by NSF grants #IIS-1514253, #IIS-1657613, #IIS-1633857, ONR grant #N00014-15-1-2339, and DARPA XAI grant #N66001-17-2-4032. İsmail İlkan Ceylan is supported by the DFG (GRK 1907). This research was conducted when he was a visiting student at UCLA.

References

[Abiteboul *et al.*, 1995] Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of databases*, volume 8. Addison-Wesley Reading, 1995.

[Beame *et al.*, 2015] Paul Beame, Guy Van den Broeck, Dan Suciu, and Eric Gribkoff. Symmetric Weighted First-Order Model Counting. In *Proceedings of PODS*, 2015.

[Bollacker *et al.*, 2008] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of SIGMOD*, 2008.

[Bordes *et al.*, 2011] Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. Learning structured embeddings of knowledge bases. In *Proceedings of AAAI*, 2011.

[Borgwardt *et al.*, 2017] Stefan Borgwardt, İsmail İlkan Ceylan, and Thomas Lukasiewicz. Ontology-mediated queries for probabilistic databases. In *Proceedings of AAAI*, 2017.

[Ceylan *et al.*, 2016] İsmail İlkan Ceylan, Adnan Darwiche, and Guy Van Den Broeck. Open-World Probabilistic Databases. In *Proceedings of KR*, 2016.

[Cozman, 2000] Fabio G. Cozman. Credal networks. *Artificial Intelligence*, 120(2):199–233, 2000.

[Dalvi and Suciu, 2012] Nilesh Dalvi and Dan Suciu. The dichotomy of probabilistic inference for unions of conjunctive queries. *Journal of ACM*, 59(6):1–87, 2012.

[De Raedt *et al.*, 2015] Luc De Raedt, Anton Dries, Ingo Thon, Guy Van den Broeck, and Mathias Verbeke. Inducing probabilistic relational rules from probabilistic examples. In *Proceedings of IJCAI*, 2015.

[Dong *et al.*, 2014] Xin Luna Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. Knowledge Vault: A Web-Scale Approach to Probabilistic Knowledge Fusion. In *Proceedings of ACM SIGKDD*, 2014.

[Ferrucci, 2012] David A. Ferrucci. Introduction to "This is Watson". *IBM Journal of Research and Development*, 56(3.4), 2012.

[Galárraga *et al.*, 2013] Luis Antonio Galárraga, Christina Teflioudi, Katja Hose, and Fabian Suchanek. Amie: association rule mining under incomplete evidence in ontological knowledge bases. In *Proceedings of WWW*, 2013.

[Gill, 1977] John Gill. Computational complexity of probabilistic turing machines. *SIAM Journal on Computing*, 6(4):675–695, 1977.

[Gribkoff *et al.*, 2014] Eric Gribkoff, Guy Van den Broeck, and Dan Suciu. Understanding the Complexity of Lifted Inference and Asymmetric Weighted Model Counting. In *Proceedings of UAI*, 2014.

[Halpern, 2003] Joseph Y Halpern. *Reasoning about uncertainty*. MIT Press, 2003.

[Hinrichs and Genesereth, 2006] Timothy Hinrichs and Michael Genesereth. Herbrand logic. Technical Report LG-2006-02, Stanford University, 2006.

[Hoffart *et al.*, 2013] Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, and Gerhard Weikum. YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. In *Proceedings of IJCAI*, 2013.

[Levi, 1980] Isaac Levi. *The Enterprise of Knowledge*. MIT Press, 1980.

[Mintz *et al.*, 2009] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of ACL-IJCNLP*, 2009.

[Mitchell *et al.*, 2015] T Mitchell, W Cohen, E Hruschka, P Talukdar, J Betteridge, A Carlson, B Dalvi, and M Gardner. Never-Ending Learning. In *Proceedings of AAAI*, 2015.

[Munroe, 2015] Randall Munroe. Google's datacenters on punch cards, 2015.

[Reiter, 1978] Raymond Reiter. On closed world data bases. *Logic and Data Bases*, pages 55–76, 1978.

[Reiter, 1980] Raymond Reiter. A logic for default reasoning. *Artificial Intelligence*, 13(1):81–132, 1980.

[Sa *et al.*, 2017] Christopher De Sa, Alexander Ratner, Christopher Ré, Jaeho Shin, Feiran Wang, Sen Wu, and Ce Zhang. Incremental knowledge base construction using deepdive. *VLDB J.*, 26(1):81–105, 2017.

[Socher *et al.*, 2013] Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. Reasoning with neural tensor networks for knowledge base completion. In *Proceedings of NIPS*, pages 926–934, 2013.

[Suciu *et al.*, 2011] Dan Suciu, Dan Olteanu, Christopher Ré, and Christoph Koch. Probabilistic Databases, 2011.

[Sutton and McCallum, 2011] Charles Sutton and Andrew McCallum. An introduction to conditional random fields. *Machine Learning*, 4(4):267–373, 2011.

[Wang *et al.*, 2013] William Yang Wang, Kathryn Mazaitis, and William W Cohen. Programming with personalized pagerank: a locally groundable first-order probabilistic logic. In *Proceedings of CIKM*, 2013.

[Wu *et al.*, 2012] Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Q. Zhu. Probase: A probabilistic taxonomy for text understanding. In *Proceedings of SIGMOD*, pages 481–492. ACM, 2012.