Statistical Relational AI: Papers from the AAAI-14 Workshop

Tractable Probabilistic Knowledge Bases: Wikipedia and Beyond

Mathias Niepert and **Pedro Domingos**
Computer Science & Engineering
University of Washington
Seattle, WA 98195-2350, USA

Abstract

Building large-scale knowledge bases from a variety of data sources is a longstanding goal of AI research. However, existing approaches either ignore the uncertainty inherent to knowledge extracted from text, the web, and other sources, or lack a consistent probabilistic semantics with tractable inference. To address this problem, we present a framework for tractable probabilistic knowledge bases (TPKBs). TPKBs consist of a hierarchy of classes of objects and a hierarchy of classes of object pairs such that attributes and relations are independent conditioned on those classes. These characteristics facilitate both tractable probabilistic reasoning and tractable maximum-likelihood parameter learning. TPKBs feature a rich query language that allows one to express and infer complex relationships between classes, relations, objects, and their attributes. The queries are translated to sequences of operations in a relational database facilitating query execution times in the sub-second range. We demonstrate the power of TPKBs by leveraging large data sets extracted from Wikipedia to learn their structure and parameters. The resulting TPKB models a distribution over millions of objects and billions of parameters. We apply the TPKB to entity resolution and object linking problems and show that the TPKB can accurately align large knowledge bases and integrate triples from open IE projects.

Introduction

A knowledge base continuously acquiring more knowledge and answering complex queries is a vision as old as the field of AI itself. Such a knowledge base would address numerous pressing problems such as data integration, information extraction, and entity resolution. However, there are three mutually conflicting properties that pertain to all KBs: (a) the ability to populate the KB from (uncertain) data; (b) the tractability of reasoning; and (c) the expressiveness of the underlying logic. When the uncertainty is modeled with a probabilistic semantics, the problem of tractability and learnability is even more challenging. Recent advances in tractable probabilistic models, however, have shown that there is the possibility of new and better trade-offs between these conflicting properties.

With this paper, we present a novel framework for tractable probabilistic knowledge bases (TPKBs) that addresses some of the aforementioned challenges. The TPKB facilitates reasoning over inheritance hierarchies and on the level of classes and relations; it provides tractable query answering via a mapping to queries in a relational database; and it features a complex query language. Instead of merely making the model more expressive, we focus on a model that is expressive while featuring a rich query language. We also emphasize the feasibility of parameter learning using large data sets and the ability to model uncertainty of class and relation hierarchies. The tractability is due to the TPKB's property that attributes of objects and object pairs are independent conditioned on classes of objects and object pairs.

Problem Statement and Contributions

In this paper, we say that a (probabilistic) knowledge base is *tractable* if computing the answer to queries takes time polynomial in the size of the PKB's data. Hence, we are concerned with polynomial data complexity (Abiteboul, Hull, and Vianu 1995). There is a large body of work on fragments of first-order logic with polynomial data complexity. Most notably, the databases literature identified numerous tractable fragments for which query processing has polynomial data complexity. The understanding of tractable fragments of first-order logic with probabilistic semantics, however, is not as thoroughly developed. More recently, work on probabilistic databases (Suciu et al. 2011) has characterized classes of unions of conjunctive queries for which query answering is possible in polynomial time. The assumption in probabilistic databases is that the instantiations of a relation (its tuples) are mutually independent. Unfortunately, even in this seemingly restrictive setting, numerous natural conjunctive queries are NP-hard. For instance, consider the query

$$\exists x, y.\mathtt{student}(x, y), \mathtt{bDate}(x, 1950), \mathtt{almaMater}(y, \mathtt{Yale})$$

that asks for the probability of there existing a pairs of objects (x, y) such that x was born in 1950, y's alma mater is Yale, and x was a student of y. This is a natural query that could, for instance, occur as part of an entity disambiguation problem. However, it is known that this query is *not* generally tractable under the assumption of tuple independence (Suciu et al. 2011). Therefore, one ought to be careful when designing probabilistic knowledge bases that need to be both tractable and useful in practice. Contrary to probabilistic databases, a probabilistic knowledge base models

Type	PKB expression	FOL expression	RDF expression
Schema	$\mathtt{sub}_{\mathtt{c}}(\mathtt{C},\mathtt{D})$	$\forall x : \mathtt{C}(x) \Rightarrow \mathtt{D}(x)$	$\mathtt{C}$ rdfs:subclassOf $\mathtt{D}$
	$\mathtt{sub}_{\mathtt{r}}(\mathtt{r},\mathtt{s})$	$\forall x, y : \mathtt{r}(x, y) \Rightarrow \mathtt{s}(x, y)$	$\mathtt{r}$ rdfs:subPropertyOf $\mathtt{s}$
Assertion	$\mathtt{ins}_{\mathtt{c}}(\mathtt{Einstein},\mathtt{Person})$	$\mathtt{Person}(\mathtt{Einstein})$	$\mathtt{Einstein}$ rdf:type $\mathtt{Person}$
	$\mathtt{ins}_{\mathtt{r}}(\mathtt{Einstein},\mathtt{Kleiner},\mathtt{advisor})$	$\mathtt{advisor}(\mathtt{Einstein},\mathtt{Kleiner})$	$\mathtt{Einstein}$ $\mathtt{advisor}$ $\mathtt{Kleiner}$
	$\mathtt{attr}_{\mathtt{o}}(\mathtt{Einstein}, 1879, \mathtt{birthYear})$	$\mathtt{birthYear}(\mathtt{Einstein}, 1879)$	$\mathtt{Einstein}$ $\mathtt{birthYear}$ 1879
	$\mathtt{attr}_{\mathtt{oo}}(\mathtt{Eugene},\mathtt{Seattle}, 283, \mathtt{distance})$	$\mathtt{distance}(\mathtt{Eugene},\mathtt{Seattle}, 283)$	reification

Table 1: Example expressions in PKB syntax with corresponding expressions in first-order logic and the resource description framework (RDF). Unlike probabilistic databases, PKBs allow queries involving schema expressions.

and reasons over uncertain inheritance hierarchies of classes and relations, that is, classes and relations can occur in the answer set of queries. Hence, while probabilistic databases are in some ways not restrictive enough they are too restrictive in other ways.

In this paper, we present a tractable probabilistic knowledge base (TPKB) that facilitates efficient query answering. Intuitively, the PKB is a probabilistic model that interconnects tractable local distributions pertaining to objects, classes, and relations in a consistent and tractable manner. The novel TPKB has the following properties

- tractable learning from existing, very large data sets;
- tractable answering of complex queries;
- employment of robust relational database technology;
- default reasoning in the presence of missing data.

We provide experimental results that demonstrate that TPKBs can be applied to common large-scale problems such as entity linking and entity resolution. This shows for the first time that a PKB over millions of objects and billions of parameters can be efficiently queried.

Related Work

PKBs are different from probabilistic databases (Suciu et al. 2011) in that PKBs represent class and relation hierarchies. Moreover, each query over the PKB corresponds to an efficient query in a (probabilistic) database. Hence, every query in the PKB is tractable and an efficient query plan does not have to be generated by a process that can take exponential time.

TPKBs are related to PROBLOG (Raedt, Kimmig, and Toivonen 2007) a probabilistic logic programming language. However, PROBLOG is not tractable and does not support existentially quantified conjunctive queries. URDF (Nakashole et al. 2012) is based on weighted MAX-SAT algorithms and is in principle intractable. It also does not support queries that ask for a distribution over objects, classes, and relations, given a subset of their attributes. Infinite relational models (IRMs) are non-parametric hierarchical models but are not tractable in general (Kemp et al. 2006). We leverage the hierarchy of TPKBs to estimate parameters on the level of classes and relation more robustly. This is related to shrinkage in text classification (McCallum et al. 1998), back off models (Katz 1987), and multilevel models in regression analysis (Gelman and Hill 2007).

Open information extraction (Etzioni et al. 2011) and other IE projects (Carlson et al. 2010; Lehmann et al. 2012) often use ad-hoc approaches and heuristics and do not provide a consistent joint distribution and query language. There exist several statistical relational systems employing relational database technology to facilitate queries over structured data (Wang et al. 2010; Noessner, Niepert, and Stuckenschmidt 2013). However, the proposed systems are intractable in genera;. Recent work on statistical relational learning has focused on tractable probabilistic graphical models, that is, probabilistic models for which inference is efficient by design. Examples are PRISM (Sato and Kameya 1997), tractable Markov logic (Domingos and Webb 2012), particular tractable fragments of probabilistic logics (Van den Broeck 2011; Niepert and Van den Broeck 2014), and probabilistic soft logic (Kimmig et al. 2012). None of these languages features complex query languages *and* uncertain inheritance hierarchies.

Probabilistic description logic programs (Lukasiewicz 2007) combine DL programs under the answer set and well-founded semantics with independent choice logic (Poole 2008). Particular variants of light-weight description logics with probabilistic semantics (Gutiérrez-Basulto et al. 2011; Niepert, Noessner, and Stuckenschmidt 2011; Noessner and Niepert 2011) have been proposed. However, these formalisms are too expressive to be tractable for the types of complex queries needed in large-scale applications, do not allow the modeling of numerical attributes, and do not address the problem of parameter learning.

There is related work in the context of information extraction and relation learning. Notable representative publications of this line of research are tensor factorizations of YAGO (Nickel, Tresp, and Kriegel 2012) and universal schemas (Riedel et al. 2013). These approaches do not facilitate a consistent probabilistic semantics and expressive query languages.

A Tractable Probabilistic Knowledge Base

A probabilistic knowledge base (PKB) models a probability distribution over possible worlds where each such possible world models an instantiation of a logical knowledge base. The design of a tractable PKB, therefore, involves (a) the logical characterization of a possible world and (b) a parameterization of possible worlds rendering probabilistic reasoning tractable. Table 1 lists several expressions in PKB syntax and in first-order logic and simplified RDF syntax.

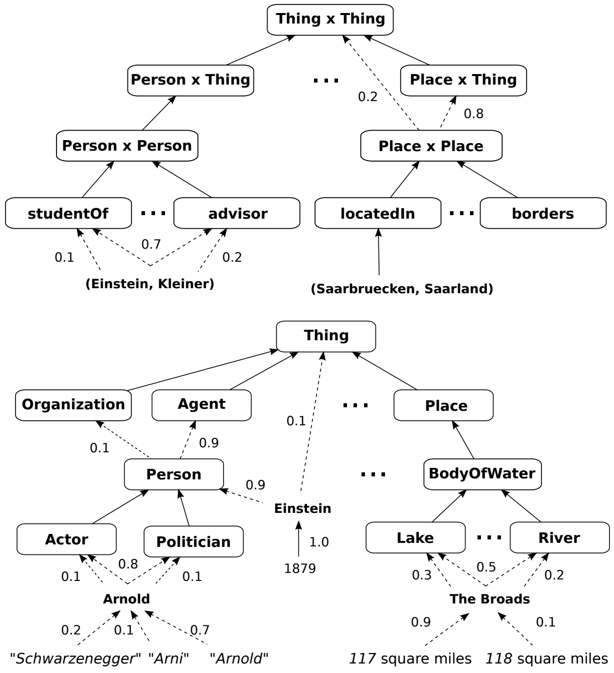

Figure 1: The TPKB models a distribution over possible worlds where each such world corresponds to a fully materialized hierarchical knowledge base modeling objects, classes, relations, and properties. Arrows correspond to *prior* conditional probabilities for subclass, subrelation, class, relation, and property assertions. In each possible world and for every entity, exactly one of the assertions modeled by outgoing dashed arrows holds. Solid arrows represent deterministic assertions.

1. The $\mathtt{dsub}_\mathtt{c}$ and $\mathtt{dsub}_\mathtt{r}$ assertions form trees on $\mathbf{C}$ and $\mathbf{R}$ with root nodes $\mathtt{Thing}$ and $\mathtt{Thing} \times \mathtt{Thing}$, respectively.
2. For all $x, y, z \in \mathbf{C}$:
   (a) $\mathtt{dsub}_\mathtt{c}(x, y) \in \mathbf{W} \Rightarrow \mathtt{sub}_\mathtt{c}(x, y) \in \mathbf{W}$;
   (b) $\mathtt{sub}_\mathtt{c}(x, y) \in \mathbf{W} \wedge \mathtt{sub}_\mathtt{c}(y, z) \in \mathbf{W} \Rightarrow \mathtt{sub}_\mathtt{c}(x, z) \in \mathbf{W}$;
   (c) $\forall \mathtt{A} \in \mathbf{A}(x)$ there exists exactly one $\mathtt{V} \in \mathbf{Val}(\mathtt{A})$ with $\mathtt{attr}_\mathtt{c}(x, \mathtt{V}, \mathtt{A}) \in \mathbf{W}$.
3. For all $x, y, z \in \mathbf{R}$:
   (a) $\mathtt{dsub}_\mathtt{r}(x, y) \in \mathbf{W} \Rightarrow \mathtt{sub}_\mathtt{r}(x, y) \in \mathbf{W}$;
   (b) $\mathtt{sub}_\mathtt{r}(x, y) \in \mathbf{W} \wedge \mathtt{sub}_\mathtt{r}(y, z) \in \mathbf{W} \Rightarrow \mathtt{sub}_\mathtt{r}(x, z) \in \mathbf{W}$;
   (c) $\forall \mathtt{A} \in \mathbf{A}(x)$ there exists exactly one $\mathtt{V} \in \mathbf{Val}(\mathtt{A})$ with $\mathtt{attr}_\mathtt{r}(x, \mathtt{V}, \mathtt{A}) \in \mathbf{W}$.
4. For all $x \in \mathbf{O}$:
   (a) $\exists \mathtt{C} \in \mathbf{C} : \mathtt{dins}_\mathtt{c}(x, \mathtt{C}) \in \mathbf{W}$;
   (b) $\forall \mathtt{C} \in \mathbf{C} : \mathtt{dins}_\mathtt{c}(x, \mathtt{C}) \in \mathbf{W} \Rightarrow \mathtt{ins}_\mathtt{c}(x, \mathtt{C}) \in \mathbf{W}$;
   (c) $\forall \mathtt{C}, \mathtt{C}' : \mathtt{ins}_\mathtt{c}(x, \mathtt{C}) \in \mathbf{W} \wedge \mathtt{sub}_\mathtt{c}(\mathtt{C}, \mathtt{C}') \in \mathbf{W} \Rightarrow \mathtt{ins}_\mathtt{c}(x, \mathtt{C}') \in \mathbf{W}$;
   (d) $\forall \mathtt{A} \in \mathbf{A}(x)$ there exists exactly one $\mathtt{V} \in \mathbf{Val}(\mathtt{A})$ with $\mathtt{attr}_\mathtt{o}(x, \mathtt{V}, \mathtt{A}) \in \mathbf{W}$.
5. For all $x, y \in \mathbf{O}$:
   (a) $\exists \mathtt{R} \in \mathbf{R} : \mathtt{dins}_\mathtt{r}(x, y, \mathtt{R}) \in \mathbf{W}$;
   (b) $\forall \mathtt{R} : \mathtt{dins}_\mathtt{r}(x, y, \mathtt{R}) \in \mathbf{W} \Rightarrow \mathtt{ins}_\mathtt{r}(x, y, \mathtt{R}) \in \mathbf{W}$;
   (c) $\forall \mathtt{R}, \mathtt{R}' : \mathtt{ins}_\mathtt{r}(x, y, \mathtt{R}) \in \mathbf{W} \wedge \mathtt{sub}_\mathtt{r}(\mathtt{R}, \mathtt{R}') \in \mathbf{W} \Rightarrow \mathtt{ins}_\mathtt{r}(x, y, \mathtt{R}') \in \mathbf{W}$;
   (d) $\forall \mathtt{A} \in \mathbf{A}(x, y)$ there exists exactly one $\mathtt{V} \in \mathbf{Val}(\mathtt{A})$ with $\mathtt{attr}_\mathtt{oo}(x, y, \mathtt{V}, \mathtt{A}) \in \mathbf{W}$.

Figure 2: Axiomatization of possible worlds.

The chosen syntax facilitates a seamless knowledge transfer from existing logical KB formalisms. We define the syntax and semantics of the novel TPKB, its parameterization, and prove tractability results.

Syntax

The domain of a PKB $\mathcal{K}$ is a set of symbols $\mathbf{K}$ defined as the union of a set of object names $\mathbf{O}$, a set of class names $\mathbf{C}$, a set of attribute names $\mathbf{A}$, and a set of binary relation names $\mathbf{R}$. Object names $\mathbf{O}$ represent individuals such as `AlbertEinstein` and `MountEverest`. Class names $\mathbf{C}$ represent sets of objects such as `Person` and `Mountain`. There is a unique class name `Thing` that represents the set of all objects. Binary relation names $\mathbf{R}$ represent subsets of $\mathtt{Thing} \times \mathtt{Thing}$, that is, classes of pairs of objects, such as `studentOf` and `actedIn`. Attribute names $\mathbf{A}$ represent properties of objects such as `birthYear`, properties of classes such as `medianAge`, and properties of relations such as `surfaceForm`. In addition, we write $\mathbf{A}(x)$ to denote the set of attributes of entity x and $\mathbf{A}(x, y)$ to denote the set of attributes of object pair $x, y \in \mathbf{O}$. Moreover, for every $\mathtt{A}$, we assume that the set of possible values $\mathbf{Val}(\mathtt{A})$ is finite. The difference between attributes and relations is that the former relate entities to values of a particular data type such as the integers or the set of all strings while the latter relate objects to objects. There are 8 predicate symbols $\mathtt{ins}_\mathtt{c}(x, y), \mathtt{ins}_\mathtt{r}(x, y, z)$, $\mathtt{attr}_\mathtt{o}(x, y, z)$, $\mathtt{attr}_\mathtt{oo}(x, y, z, z')$, $\mathtt{attr}_\mathtt{c}(x, y, z)$, $\mathtt{attr}_\mathtt{r}(x, y, z)$, $\mathtt{sub}_\mathtt{c}(x, y)$, and $\mathtt{sub}_\mathtt{r}(x, y)$ representing class, relation, attribute, subclass, and subrelation assertions. Note that classes and relations are reified allowing us to express arbitrary relations between object pairs. Table 1 depicts some example expressions formed with these predicates. There are 4 additional predicates $\mathtt{dsub}_\mathtt{c}(x, y), \mathtt{dsub}_\mathtt{r}(x, y), \mathtt{dins}_\mathtt{c}(x, y)$, and $\mathtt{dins}_\mathtt{r}(x, y, z)$ whose sole purpose is to model *prior* conditional probabilities for subclass, subrelation, and instance assertions. A grounding of a predicate is obtained by replacing each variable with the appropriate symbols in $\mathbf{K}$. The Herbrand base of $\mathbf{K}$ is the set of all ground predicates. Each subset of the Herbrand base is a Herbrand interpretation specifying which ground predicates are true. We make the closed (active) domain assumption meaning that quantifiers range over elements in $\mathbf{K}$ (Abiteboul, Hull, and Vianu 1995).

Semantics

Every TPKB's joint distribution is fully characterized by two components. First, a set of first-order formulas $\mathbf{T}$ that axiomatize the Herbrand models (possible worlds) of the TPKB. A Herbrand model $\mathbf{W}$ is a Herbrand interpretation that satisfies all formulas in $\mathbf{T}$. We use the words Herbrand model and possible world interchangeably. Second, a set of *prior* conditional probabilities for knowledge base assertions. Prior probabilities *might differ* from the posterior probabilities obtained via inference as the axiomatization introduces complex dependencies between assertions.

There are several possible ways to specify an axiomatization of possible worlds and a parameterization in form of prior probabilities. We focus on one particular combination that results in tractable probabilistic inference. First, we characterize a possible world $\mathbf{W}$ with the axioms in Figure 2. These axioms make every possible world corresponds to a fully materialized description logic based ontology with classes, roles (binary predicates), individuals (objects), and concrete domains (attributes) (Baader et al. 2003).

Second, we specify *prior* conditional probabilities for assertions of the knowledge base. For every $\mathtt{C} \in \mathbf{C} \setminus \{\mathtt{Thing}\}$, $P(\mathtt{dsub_c}(\mathtt{C}, \mathtt{C}'))$ is the *prior* probability of an object being an instance of class $\mathtt{C}'$ given that it is an instance of class $\mathtt{C}$. Since we specify conditional distributions, we must have that $\sum_{\mathtt{C}' \in \mathbf{C}} P(\mathtt{dsub_c}(\mathtt{C}, \mathtt{C}')) = 1$, for every $\mathtt{C} \in \mathbf{C}$, and we write $P(\mathtt{dsub_c}(\mathtt{C}, \cdot))$ for this conditional distribution. For example, the probabilistic KB depicted in Figure 1 has the prior probabilities $P(\mathtt{dsub_c}(\mathtt{Person}, \mathtt{Organization})) = 0.1$, $P(\mathtt{dsub_c}(\mathtt{Person}, \mathtt{Agent})) = 0.9$, and $P(\mathtt{dsub_c}(\mathtt{Person}, \mathtt{C}')) = 0.0$ for all other $\mathtt{C}' \in \mathbf{C}$. In this example, the *prior* probability of $\mathtt{dsub_c}(\mathtt{Person}, \mathtt{Thing})$ is 0.0 but the *posterior* probability of $\mathtt{sub_c}(\mathtt{Person}, \mathtt{Thing})$ is 1.0 which is due to the axiomatization of a possible world in Figure 2. We have to impose a mild restriction on the *prior* probabilities to make them compatible with the axiomatization of possible worlds. We assume that we can partition the classes $\mathbf{C}$ into sets $\mathbf{C}_1, ..., \mathbf{C}_k$ with $\mathbf{C}_1 = \{\mathtt{Thing}\}$ such that for all $1 \leq i \leq j \leq k$, if $\mathtt{C}_i \in \mathbf{C}_i$ and $\mathtt{C}_j \in \mathbf{C}_j$ then $P(\mathtt{dsub_c}(\mathtt{C}_i, \mathtt{C}_j)) = 0$. Stratification ensures that, in every possible world, classes in stratum $\mathbf{C}_i$ can only be subclasses of classes in strata $\mathbf{C}_j$ with $j < i$ which, in turn, ensures that the $\mathtt{dsub_c}$ assertions form a tree. We refer to this restriction on the parameterization as *stratification* and to k as the stratification size. For every $\mathtt{R} \in \mathbf{R} \setminus \{\mathtt{Thing} \times \mathtt{Thing}\}$, $P(\mathtt{dsub_r}(\mathtt{R}, \cdot))$ is the prior conditional distribution for relations with stratification restrictions identical to those made for classes.

For every $\mathtt{O} \in \mathbf{O}$, $P(\mathtt{dins_c}(\mathtt{O}, \mathtt{C}_1) \wedge ... \wedge \mathtt{dins_c}(\mathtt{O}, \mathtt{C}_n))$ is the *prior* probability of object $\mathtt{O}$ being an instance of exactly the classes $\mathtt{C}_1, .., \mathtt{C}_n$. We must have that $\sum_{\mathbf{C}' \subseteq \mathbf{C}} P(\bigwedge_{\mathtt{C}' \in \mathbf{C}'} \mathtt{dins_c}(\mathtt{O}, \mathtt{C}')) = 1$ for every $\mathtt{O} \in \mathbf{O}$. For example, the probabilistic KB depicted in Figure 1 has the prior probabilities $P(\mathtt{dins_c}(\mathtt{Arnold}, \mathtt{Actor})) = \mathtt{0.1}$, $\mathtt{P}(\mathtt{dins_c}(\mathtt{Arnold}, \mathtt{Politician}) \wedge \mathtt{dins_c}(\mathtt{Arnold}, \mathtt{Actor})) = \mathtt{0.8}, \mathtt{P}(\mathtt{dins_c}(\mathtt{Arnold}, \mathtt{Politician})) = 0.1$, and $P(\bigwedge_{\mathtt{C}' \in \mathbf{C}'} \mathtt{dins_c}(\mathtt{Arnold}, \mathtt{C}')) = 0$ for all other $\mathbf{C}' \subseteq \mathbf{C}$. In order to ensure tractability, we have to impose two restrictions. First, we have to ensure that every object is an instance of at least one class. Hence, there must exists a $\mathbf{C}' \subseteq \mathbf{C}$ with $P(\bigwedge_{\mathtt{C}' \in \mathbf{C}'} \mathtt{dins_c}(\mathtt{O}, \mathtt{C}')) > 0$ for all $\mathtt{O} \in \mathbf{O}$. Moreover, we have to bound the number of sets of classes an object can be an instance of. That is, we assume there exists a constant $c \in \mathbb{N}$ which is bounded polynomially in the size of the PKB and, for every $\mathtt{O} \in \mathbf{O}$, we have that $|\{\mathbf{C}' \subseteq \mathbf{C} : P(\bigwedge_{\mathtt{C}' \in \mathbf{C}'} \mathtt{dins_c}(\mathtt{O}, \mathtt{C}')) > 0\}| < c$. For every pair of objects $\mathtt{O}, \mathtt{O}' \in \mathbf{O}$, $P(\mathtt{ins_r}(\mathtt{O}, \mathtt{O}', \mathtt{X}))$ specifies the analogous prior probabilities for object pairs with restrictions identical to those made for objects.

For every $\mathtt{O} \in \mathbf{O}$ and every $\mathtt{A} \in \mathbf{A}(\mathtt{O})$, $P(\mathtt{attr_o}(\mathtt{O}, X, \mathtt{A}))$ is the probability of object $\mathtt{O}$ having value $X \in \mathbf{Val}(\mathtt{A})$ for attribute $\mathtt{A}$. For example, for the PKB depicted in Figure 1, we have $P(\mathtt{attr_o}(\mathtt{Arnold}, \text{“Arni”}, \mathtt{surfaceForm})) = 0.1$, $P(\mathtt{attr_o}(\mathtt{Arnold}, \text{“Arnold”}, \mathtt{surfaceForm})) = 0.7$, and $P(\mathtt{attr_o}(\mathtt{Arnold}, \text{“Schwarzenegger”}, \mathtt{surfaceForm})) = 0.2$. The families of prior conditional distributions $P(\mathtt{attr_{oo}}(\mathtt{O}, \mathtt{O}', \cdot, \mathtt{A})), P(\mathtt{attr_c}(\mathtt{C}, \cdot, \mathtt{A}))$, and $P(\mathtt{attr_r}(\mathtt{R}, \cdot, \mathtt{A}))$ are the analogous conditional distributions for object pairs, classes, and relations.

The probability of a possible world now factorizes into the probability of the class hierarchy, the probability of the relation hierarchy, and the probability of instance and attribute assertions given these hierarchies:

$$P(\mathbf{W}) = \prod_{\mathtt{dsub_c}(\mathtt{C},\mathtt{C}') \in \mathbf{W}} P(\mathtt{dsub_c}(\mathtt{C}, \mathtt{C}')) \prod_{\mathtt{attr_c}(\mathtt{C},\mathtt{V},\mathtt{A}) \in \mathbf{W}} P(\mathtt{attr_c}(\mathtt{C}, \mathtt{V}, \mathtt{A})) \times$$

$$\prod_{\mathtt{dsub_r}(\mathtt{R},\mathtt{R}') \in \mathbf{W}} P(\mathtt{dsub_r}(\mathtt{R}, \mathtt{R}')) \prod_{\mathtt{attr_r}(\mathtt{R},\mathtt{V},\mathtt{A}) \in \mathbf{W}} P(\mathtt{attr_r}(\mathtt{R}, \mathtt{V}, \mathtt{A})) \times$$

$$\prod_{\mathtt{O} \in \mathbf{O}} \left[P(\bigwedge_{\mathtt{C} \in \mathbf{C}_\mathtt{O}} \mathtt{dins_c}(\mathtt{O}, \mathtt{C})) \prod_{\mathtt{attr_o}(\mathtt{O},\mathtt{V},\mathtt{A}) \in \mathbf{W}} P(\mathtt{attr_o}(\mathtt{O}, \mathtt{V}, \mathtt{A})) \right] \times$$

$$\prod_{\mathtt{O},\mathtt{O}' \in \mathbf{O}} \left[P(\bigwedge_{\mathtt{R} \in \mathbf{R}_{\mathtt{O},\mathtt{O}'}} \mathtt{dins_r}(\mathtt{O}, \mathtt{O}', \mathtt{R})) \prod_{\mathtt{attr_{oo}}(\mathtt{O},\mathtt{O}',\mathtt{V},\mathtt{A}) \in \mathbf{W}} P(\mathtt{attr_{oo}}(\mathtt{O}, \mathtt{O}', \mathtt{V}, \mathtt{A})) \right],$$

where $\mathbf{C}_\mathtt{O} = \{\mathtt{C} \in \mathbf{C} : \mathtt{dins_c}(\mathtt{O}, \mathtt{C}) \in \mathbf{W}\}$ and $\mathbf{R}_{\mathtt{O},\mathtt{O}'} = \{\mathtt{R} \in \mathbf{R} : \mathtt{dins_r}(\mathtt{O}, \mathtt{O}', \mathtt{R}) \in \mathbf{W}\}$. We can now state the following theorem.

Theorem 1. *Let $\mathcal{K}$ be a PKB, let $\mathcal{W}$ be the set of possible worlds for $\mathcal{K}$, and let P be the distribution over possible worlds. Then,* $\sum_{\mathbf{W} \in \mathcal{W}} P(\mathbf{W}) = 1.0$.

Probabilistic Reasoning

The joint distribution over possible worlds is a product of conditional distributions. The conditional distributions are representable with block-independent-disjoint (BID) tables of a probabilistic database. BID tables are table where the tuples can be partitioned into blocks such that tuples within the same block are mutually exclusive and tuples in different blocks are independent (Suciu et al. 2011). This opens the door to the application of known tractability results from the probabilistic database literature. However, for the types of queries typically performed with a PKB, the known results do not generally apply and we have to prove the tractability of certain queries only partially relying on known results.

We first define the query language and semantics. A formula is given by the following grammar:

$$Q := u = v \mid R(\overline{x}) \mid \exists x.Q_1 \mid Q_1 \wedge Q_2 \mid Q_1 \vee Q_2$$

where $R(\overline{x})$ is one of the TPKB's predicate symbols with variables and/or constants from $\mathbf{K}$. A union of conjunctive

$Q_1(x)$	:=	$\mathtt{attr_o}(x, \mathtt{Val}_1, \mathtt{A}_1), ..., \mathtt{attr_o}(x, \mathtt{Val}_n, \mathtt{A}_n)$
$Q_2(x)$	:=	$\mathtt{ins_c}(x, \mathtt{C}_1), ..., \mathtt{ins_c}(x, \mathtt{C}_n)$
$Q_3(x, y)$	:=	$\mathtt{ins_c}(x, y), Q_1(x), Q_2(x)$
$Q_4(x)$	:=	$\mathtt{sub_c}(x, \mathtt{C}_1), ..., \mathtt{sub_c}(x, \mathtt{C}_n)$
$Q_5(x)$	:=	$Q_1(x), Q_2(x)$
$Q_6(x, y)$	:=	$\mathtt{ins_r}(x, y, \mathtt{R}_1), ..., \mathtt{ins_r}(x, y, \mathtt{R}_n)$
$Q_7(x, y, z)$	:=	$\mathtt{ins_r}(x, y, z), Q_6(x, y)$
$Q_8(x, y)$	:=	$\mathtt{sub_r}(x, y, \mathtt{R}_1), ..., \mathtt{sub_r}(x, y, \mathtt{R}_n)$
$Q_9(x, y)$	:=	$Q_6(x, y), Q_5(x), Q_5(y)$
$Q_{10}(x, y)$	:=	$Q_7(x, y), Q_5(x), Q_5(y)$

Table 2: A collection of tractable query families.

queries (UCQ) has the form $Q(\overline{x})$ where $\overline{x}$ are the free variables of formula Q. A Boolean query is a query without free variables. We write $Q[\overline{a}/\overline{x}]$ to refer to the query expression Q where the free variables $\overline{x}$ are substituted with constants $\overline{a}$. When a query has free variables $\overline{x}$, then for possible world $\mathbf{W}$, $Q(\mathbf{W}) = \{\overline{a} : \mathbf{W} \models Q[\overline{a}/\overline{x}]\}$ where $W \models Q$ means Q is true in $\mathbf{W}$. We refer to the elements $Q(\mathbf{W})$ as tuples. Given a PKB $\mathcal{K}$ with possible worlds $\mathcal{W}$, the *marginal probability* of tuple t for a query Q is

$$P(t \in Q) = \sum_{\mathbf{W} \in \mathcal{W} : t \in Q(\mathbf{W})} P(\mathbf{W}).$$

We construct mappings from queries over the PKB to queries over a probabilistic database and exploit the hierarchical structure of possible worlds to derive tractability results and efficient query evaluation plans. Note that we can compute probabilities *conditioned* on attribute values by normalizing the query probabilities with the probability of the tractable Boolean query $\exists x.Q_1(x)$.

Consider the families of queries in Table 2. Queries of type Q_1 return, for a given set of attribute values, a list of objects with their marginal probabilities. For instance, the query "All entities born in 1950 and with surface form 'Einstein'" can be expressed. Queries of type Q_2, Q_3, and Q_4 are queries one typical asks against a logical KB. For instance, the query "All classes/instances that are subclasses/instances of Politician and Actor" can be expressed. Queries of type Q_3 return a list of object class tuples given a set of attribute values and instance restrictions. For instance, the query "All classes whose instances were born after 1950 and are instances of the class Person." Queries of type Q_9 return a list of object pairs that are related via a set of relations $\mathtt{R}_i$ given a set of attribute values and instance restrictions. For instance, the query "All object pairs (x, y) that are in a studentOf relation and where x was born in 1950 and y has surface form 'Bach'" is a member of this query family. Query Q_{10} returns a list of object pairs and relations that could hold between objects x and y given a set of attribute values and instance restrictions for the two objects such as "All relations that could hold between a person and an organization where the person's surface form is 'Obama' and the organization was founded after 2009."

Theorem 2. *Query types $Q_1, ..., Q_{10}$ are tractable.*

The tractability of most queries does not follow from the probabilistic database literature and the proofs in the appendix are quite involved. This is because the posterior probabilities for the $\mathtt{ins_c}$, $\mathtt{ins_r}$, $\mathtt{sub_c}$, and $\mathtt{sub_r}$ assertions are not identical to the prior probabilities. Moreover, there are only few tractability results for conjunctive queries with self-joins. The TPKB's hierarchical structure of the possible worlds and the stratification of the parameterization render the queries tractable. Table 2 lists classes of queries that are important for applications such as entity resolution and data integration, however, there are other classes of tractable queries and this is often easily verifiable.

Modeling and Learning

The most sophisticated PKB is of limited use if learning its parameters from existing data sets is not possible. Users of the TPKB only need to worry about learning (some of) the prior conditional distributions. *Both* structure and parameter learning is accomplished by estimating these parameters *independently* because the likelihood decomposes into a product of the local conditional distributions of the joint distribution. It is possible to incrementally add attribute and object distributions without having to relearn the entire PKB. Especially if the parameter values are learned by processing very large data sets, this property of incremental learning is crucial. Since there are no latent variables and all parameters to be estimated are those of conditional distributions, we can apply closed-form maximum-likelihood learning as long as the chosen parameterization of the random variables allows this. For instance, this is always possible if we use multinomials to model the distributions.

Let $\mathbf{W}_1, ..., \mathbf{W}_N$ be sets of assertions of the PKB's predicates. For instance, we estimate the parameter $P(\mathtt{dsub_c}(\mathtt{C}, \mathtt{C'}))$ of the multinational distribution with

$$\frac{1}{|N||\mathbf{O}|} \sum_{i=1}^{N} \sum_{\mathtt{O} \in \mathbf{O}} \frac{[[\mathtt{ins_c}(\mathtt{O}, \mathtt{C'}) \in \mathbf{W}_i \text{ and } \mathtt{ins_c}(\mathtt{O}, \mathtt{C}) \in \mathbf{W}_i]]}{[[\mathtt{ins_c}(\mathtt{O}, \mathtt{C}) \in \mathbf{W}_i]]},$$

where $[[\cdot]]$ is the indicator function. Similarly, we can estimate the parameters for the other distributions. The only additional constraint is that the parameterization is stratified and that every object (object pair) is an instance of at least one class (relation) in a possible world.

We also learn histograms to estimate the densities of numerical attribute values on the level of classes and relations. These histograms can be leveraged for *default reasoning* when data on the level of objects is missing. For instance, consider the attribute birthYear. There are numerous persons without an assertion for this property in any of the information extraction projects. However, the TPKB can still compute the probability for those objects by using the distribution of birthYear on the class level. For instance, Arnold who is an instance of the class Actor might not have an assertion for the attribute birthYear. In this case, we use the probability of an actor being born in 1950 as a surrogate for $P(\mathtt{attr_o}(\mathtt{Arnold}, 1950, \mathtt{birthDate}))$.

Experiments

For the experiments, we learned a TPKB's direct subclass and instance distributions (both for objects and object pairs)

	Precision@1			Recall@1		
NELL relation	WL	TPKB1	TPKB2	WL	TPKB1	TPKB2
`ActorStarredInMovie`	0.81	0.85	0.95	0.82	0.81	0.31
`AgentcollaboratesWithAgent`	0.82	0.83	0.95	0.86	0.84	0.20
`AnimalIsTypeofAnimal`	0.86	0.86	1.00	0.86	0.85	0.00
`AthleteLedSportsTeam`	0.89	0.91	0.92	0.86	0.79	0.37
`BankBankInCountry`	0.82	0.87	0.92	0.76	0.69	0.10
`CityLocatedInState`	0.80	0.85	0.96	0.81	0.80	0.64
`BookWriter`	0.82	0.83	0.92	0.83	0.82	0.73
`CompanyAlsoKnownAs`	0.71	0.71	1.00	0.58	0.28	0.00
`PersonLeadsOrganization`	0.79	0.81	0.93	0.75	0.57	0.11
`TeamPlaysAgainstTeam`	0.81	0.81	1.00	0.81	0.73	0.00
`WeaponMadeInCountry`	0.88	0.91	1.00	0.88	0.84	0.00
`LakeInState`	0.90	0.91	1.00	0.92	0.90	0.84

System	Precision	Recall
PARIS	91.9	73.8
TPKB1@1	85.3	72.4
TPKB2@1	89.3	69.3
TPKB3@1	92.8	67.1
TPKB1@2	-	73.1
TPKB2@2	-	74.0
TPKB3@2	-	74.2

Table 3: Results for entity resolution experiments (left) and entity linking experiments (right). WL is the WikiLink baseline.

using the ontology and the data from DBPEDIA (Lehmann et al. 2014; Bizer, Heath, and Berners-Lee 2009) and we manually created a shallow relation hierarchy. For several attributes on the class and object level, such as `birthDate`, `elevation`, `geocoordinates`, etc. we also used data in DBPEDIA. To learn these attributes on the object level, we assumed a uniform distribution if, for one object, more than one value is given for a particular attribute. For instance, if `Arnold` has values 1947 and 1946 for attribute `birthDate` then, following the maximum-likelihood principle, we assume that both values have probability 0.5. For the class level, we pooled attribute values of the instances of each class and used histograms to model these distributions. For the object attribute `surfaceForm` we used the WIKIPREP tool (Gabrilovich and Markovitch 2006; 2007) to compute the conditional distributions using Wikipedia's link structure. Based on object attributes it is possible to learn the parameters of attributes for object pairs. For instance, we introduced the attribute `diffBirthYear` which models a distributions over the absolute value of birth year differences. For instance, if `Arnold` was born in 1947 or 1946 and `Einstein` in 1879, then the value for this attribute would be 68 and 67 with probability 0.5 each. We also pool attributes for distributions on the level of relations. The number of parameters of the resulting TPKB exceeds 1 billion and we model more than 1 million objects and object pairs. We ran the respective conjunctive queries using a MYSQL database and *stored functions* for the computations of the extensional queries. Each query used for the experiments could be answered in less than one second.

We evaluated the learned TPKB on two important problem classes which we discuss in the following.

Entity Linking For the entity linking experiments we used an existing gold standard for aligning NELL triples (Carlson et al. 2010) to DBPEDIA (Dutta et al. 2013). We run two different queries to compute, for each triple, subject and object alignments using the same experimental set-up as previous work (Dutta et al. 2013). For each of the NELL predicates in Table 3(left) we manually aligned the domain and range with classes in the TPKB if possible. For instance, for the first relation, we know that subjects are instances of class `Actor` and objects are instances of class `Film`. The queries are now $Q_5(x)$ where we used only the attribute `surfaceForm` and the aligned classes (TPKB1); and query $Q_{10} := \mathtt{ins_r}(x, y, z), \mathtt{attr_o}(x, \mathtt{Val_1}, \mathtt{surfaceForm})$ $\mathtt{ins_c}(x, \mathtt{C}), \mathtt{attr_o}(y, \mathtt{Val_2}, \mathtt{surfaceForm}), \mathtt{ins_c}(y, \mathtt{C})$ where we only use answer tuples with $z \neq \mathtt{Thing} \times \mathtt{Thing}$ (TPKB2). This query retrieves all object pairs that are instance of a relation other than $\mathtt{Thing} \times \mathtt{Thing}$ given attribute values and class memberships. The results are given in Table 3 (left) and compared with a baseline given in (Dutta et al. 2013). Precision@k and Recall@k is computed by retrieving the k most probable answer tuples.

Entity Resolution Entity resolution is the problem of determine whether two objects are equivalent. To evaluate the TPKB for entity linking we repeated the experiment of linking YAGO (Hoffart et al. 2013) to DBPEDIA conducted to evaluate the PARIS matching system (Suchanek, Abiteboul, and Senellart 2011). Both knowledge bases use Wikipedia identifiers for their objects which gives us a large set of gold standard pairs for evaluation purposes. We manually aligned a set of attributes (datatype properties) and classes between YAGO and the TPKB (DBPEDIA). We sampled 100000 objects in YAGO, retrieved the aligned attributes for each object (labels, numerical attributes, etc.) and ran, for each object, the query Q_1 only with attribute `surfaceForm` (TPKB1); and all other attributes for which a manual alignment existed (TPKB2). Moreover, we ran the query $Q_5(x)$ when an alignment between at least two classes existed (TPKB3). Table 3 shows that TPKBs are able to accurately link entities and compare favorably with specialized algorithms.

The TPKB is both efficient and performs comparable to existing problem specific and less versatile approaches.

Future Work

Directions for future work include more expressive variants of the presented TPKB framework, more applications, and learning TPKBs from multiple IE projects.

References

Abiteboul, S.; Hull, R.; and Vianu, V. 1995. *Foundations of Databases*. Addison-Wesley.

Baader, F.; Calvanese, D.; McGuinness, D. L.; Nardi, D.; and Patel-Schneider, P. F., eds. 2003. *The Description Logic Handbook*. Cambridge University Press.

Bizer, C.; Heath, T.; and Berners-Lee, T. 2009. Linked data - the story so far. *Int. J. Semantic Web Inf. Syst.* 5(3):1–22.

Carlson, A.; Betteridge, J.; Kisiel, B.; Settles, B.; Jr., E. R. H.; and Mitchell, T. M. 2010. Toward an architecture for never-ending language learning. In Fox, M., and Poole, D., eds., *AAAI*. AAAI Press.

Domingos, P., and Webb, W. A. 2012. A tractable first-order probabilistic logic. In Hoffmann, J., and Selman, B., eds., *AAAI*. AAAI Press.

Dutta, A.; Meilicke, C.; Niepert, M.; and Ponzetto, S. P. 2013. Integrating open and closed information extraction: Challenges and first steps. In *Proceedings of NLP-DBPEDIA@ISWC*.

Etzioni, O.; Fader, A.; Christensen, J.; Soderland, S.; and Mausam. 2011. Open information extraction: The second generation. In Walsh, T., ed., *IJCAI*, 3–10. IJCAI/AAAI.

Gabrilovich, E., and Markovitch, S. 2006. Overcoming the brittleness bottleneck using wikipedia: Enhancing text categorization with encyclopedic knowledge. In *AAAI*, 1301–1306. AAAI Press.

Gabrilovich, E., and Markovitch, S. 2007. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In Veloso, M. M., ed., *IJCAI*, 1606–1611.

Gelman, A., and Hill, J. 2007. *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Analytical Methods for Social Research. Cambridge University Press.

Gutiérrez-Basulto, V.; Jung, J. C.; Lutz, C.; and Schröder, L. 2011. A closer look at the probabilistic description logic prob-el. In *Proceedings of AAAI*.

Hoffart, J.; Suchanek, F. M.; Berberich, K.; and Weikum, G. 2013. Yago2: A spatially and temporally enhanced knowledge base from wikipedia. *Artif. Intell.* 194:28–61.

Katz, S. 1987. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *Acoustics, Speech and Signal Processing* 35(3):400–401.

Kemp, C.; Tenenbaum, J. B.; Griffiths, T. L.; Yamada, T.; and Ueda, N. 2006. Learning systems of concepts with an infinite relational model. In *Proceedings of AAAI*, 381–388.

Kimmig, A.; Bach, S. H.; Broecheler, M.; Huang, B.; and Getoor, L. 2012. A short introduction to probabilistic soft logic. In *NIPS Workshop on Probabilistic Programming: Foundations and Applications*.

Lehmann, J.; Gerber, D.; Morsey, M.; and Ngomo, A.-C. N. 2012. Defacto - deep fact validation. In *International Semantic Web Conference (1)*, 312–327.

Lehmann, J.; Isele, R.; Jakob, M.; Jentzsch, A.; Kontokostas, D.; Mendes, P. N.; Hellmann, S.; Morsey, M.; van Kleef, P.; Auer, S.; and Bizer, C. 2014. DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal*.

Lukasiewicz, T. 2007. Probabilistic description logic programs. *Int. J. Appr. Reas.* 45.

McCallum, A.; Rosenfeld, R.; Mitchell, T. M.; and Ng, A. Y. 1998. Improving text classification by shrinkage in a hierarchy of classes. In *Proceedings of ICML*, 359–367.

Nakashole, N.; Sozio, M.; Suchanek, F. M.; and Theobald, M. 2012. Query-time reasoning in uncertain rdf knowledge bases with soft and hard rules. In Brambilla, M.; Ceri, S.; Furche, T.; and Gottlob, G., eds., *VLDS*, volume 884 of *CEUR Workshop Proceedings*, 15–20. CEUR-WS.org.

Nickel, M.; Tresp, V.; and Kriegel, H.-P. 2012. Factorizing yago: Scalable machine learning for linked data. In *Proceedings of WWW*, 271–280.

Niepert, M., and Van den Broeck, G. 2014. Tractability through exchangeability: A new perspective on efficient probabilistic inference. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI)*.

Niepert, M.; Noessner, J.; and Stuckenschmidt, H. 2011. Log-linear description logics. In *Proceedings of IJCAI*, 2153–2158.

Noessner, J., and Niepert, M. 2011. Elog: A probabilistic reasoner for owl el. In *Proceedings of the 5th Conference on Web Reasoning and Rule Systems (RR)*, 281–286.

Noessner, J.; Niepert, M.; and Stuckenschmidt, H. 2013. Rockit: Exploiting parallelism and symmetry for map inference in statistical relational models. In *Proceedings of the 27th Conference on Artificial Intelligence (AAAI)*.

Poole, D. 2008. The independent choice logic and beyond. In *Probabilistic inductive logic programming*. Springer-Verlag.

Raedt, L. D.; Kimmig, A.; and Toivonen, H. 2007. Problog: a probabilistic prolog and its application in link discovery. In *In Proceedings of 20th International Joint Conference on Artificial Intelligence*, 2468–2473.

Riedel, S.; Yao, L.; Marlin, B. M.; and McCallum, A. 2013. Relation extraction with matrix factorization and universal schemas. In *Proceedings of HLT-NAACL*.

Sato, T., and Kameya, Y. 1997. Prism: A language for symbolic-statistical modeling. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI)*, 1330–1339.

Suchanek, F. M.; Abiteboul, S.; and Senellart, P. 2011. Paris: Probabilistic alignment of relations, instances, and schema. *PVLDB* 5(3):157–168.

Suciu, D.; Olteanu, D.; Christopher, R.; and Koch, C. 2011. *Probabilistic Databases*. Morgan & Claypool Publishers, 1st edition.

Van den Broeck, G. 2011. On the completeness of first-order knowledge compilation for lifted probabilistic inference. In *Proceedings of NIPS*, 1386–1394.

Wang, D. Z.; Franklin, M. J.; Garofalakis, M. N.; and Hellerstein, J. M. 2010. Querying probabilistic information extraction. *PVLDB* 3(1):1057–1067.