pSPARQL: A Querying Language for Probabilistic RDF (Extended Abstract)

Hong Fang[1] and Xiaowang Zhang[2,3,4,⋆]

[1] College of Arts and Sciences, Shanghai Polytechnic University, Shanghai 201209, China
[2] School of Computer Science and Technology, Tianjin University, Tianjin 300350, China
[3] Tianjin Key Laboratory of Cognitive Computing and Application, Tianjin 300350, China
[4] Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, Nanjing 211189, China

Abstract. In this paper, we present a querying language for probabilistic RDF databases, where each triple has a probability, called pSRARQL, built on SPARQL, recommended by W3C as a querying language for RDF databases. Firstly, we present the syntax and semantics of pSPARQL. Secondly, we define the query problem of pSPARQL corresponding to probabilities of solutions. Finally, we show that the query evaluation of general pSPARQL patterns is PSPACE-complete.

1 Introduction

Resource Description Framework (RDF)[5] is the standard data model in the Semantic Web. In our real world, RDF data possibly contains some uncertainty data due to the diversity of data sources such as YAGO[6]. For instance, some RDF data is generated from raw data via knowledge extraction and machine learning. However, RDF model itself provides little support for uncertain data [3]. There are some approaches to querying over probabilistic RDF [1,2,4,3]. Though those approaches can query probabilistic RDF, they cannot support S-PARQL [7] which, recommended by W3C, has become the standard language for querying RDF data since 2008. Indeed, SPARQL has been applied to query probabilistic ontologies [6].

In this paper, we present an extended querying language (called *pSPARQL: probabilistic SPARQL*) for probabilistic RDF databases with supporting SPARQL. As an important result of this paper, we show that the query evaluation of general pSPARQL patterns is PSPACE-complete, which has the same complexity of SPARQL.

[⋆] Corresponding author: xiaowangzhang@tju.edu.cn

[5] `https://www.w3.org/TR/rdf11-primer/`

[6] `http://www.mpi-inf.mpg.de/yago./`

[7] `https://www.w3.org/TR/rdf-sparql-query/`

2 Probabilistic RDF

Let I, B, and L be infinite sets of *IRIs*, *blank nodes* and *literals*, respectively. These three sets are pairwise disjoint. We denote the union $I \cup B \cup L$ by U, and elements of $I \cup L$ will be referred to as *constants*.

A triple $(s, p, o) \in (I \cup B) \times I \times (I \cup B \cup L)$ is called an *RDF triple*. An *RDF graph* is a finite set of RDF triples.

A *probabilistic RDF* R is a pair (G, ρ) where G is an RDF graph and ρ is a total function from $G \to [0, 1]$. Intuitively speaking, ρ is a probability function mapping each triple to a probability.

For instance, let $R = (G, \rho)$ be a probabilistic RDF with $G = \{t_1, t_2, t_3\}$ and ρ is a function from $G \to [0, 1]$ defined in the following table.

No	**Triple**	ρ
t_1	(*John*, *sufferedFrom*, *Schizophrenia*)	0.32
t_2	(*John*, *sufferedFrom*, *MentalDisorder*)	0.84
t_3	(*John*, *Treatedby*, *Psychiatrist*)	0.95

3 pSPARQL

In this section, we introduce a probabilistic SPARQL (for short, pSPARQL).

Patterns The syntax of pSPARQL is slightly different from the syntax of SPARQL [5].

Assume furthermore an infinite countable set V of *variables*, disjoint from U. It is a SPARQL convention to prefix each variable with a question mark.

Patterns are now inductively defined as follows.

- Any triple from $(I \cup L \cup V) \times (I \cup V) \times (I \cup L \cup V)$ is a pattern (called a *triple pattern*.
- If P_1 and P_2 are patterns, then so are the following: P_1 UNION P_2, P_1 AND P_2, and P_1 DIFF P_2.
- If P is a pattern and C is a constraint, then P FILTER C is a pattern. Here, a *constraint* is a boolean combination ($C_1 \wedge C_2$, $C_1 \vee C_2$, or $\neg C$) of *atomic constraints* with one of the three following forms: $\mathrm{bound}(?x)$, $?x = ?y$, and $?x = c$ with $?x, ?y \in V$ and $c \in U$.

Note that, in pSRARQL, we leave out the OPTIONAL operator while we add the DIFF operator in SPARQL 1.1. Indeed, the treatment is allowed since OPTIONAL can be expressed by AND, UNION, and DIFF as follows:

$$P \text{ OPT } Q = (P \text{ AND } Q) \text{ UNION } (P \text{ DIFF } Q). \tag{1}$$

Semantics The semantics of pSPARQL patterns is defined in terms of sets of pairs of the form (μ, p) (called a *solution*) where μ is simply a total function $\mu\colon S \to U$ on some finite set S of variables and $p \in [0,1]$. We denote the domain S of μ by $\mathrm{dom}(\mu)$. Note that (μ, p) is meaningless if $p = 0$. For simplification, we mainly consider $p \neq 0$ in the following.

Now given an RDF graph $R = (G, \rho)$ and a pattern P, we define the semantics of P on R, denoted by $[\![P]\!]_R$, as a set of mappings, in the following manner.

- If P is a triple pattern (v_1, v_2, v_3), then

$$\begin{aligned}[\![P]\!]_R := \{(\mu, p)\colon \{v_1, v_2, v_3\} \cap V \to U \mid \\ (\mu(v_1), \mu(v_2), \mu(v_3)) \in G \text{ and} \\ p = \max_{(\mu(v_1),\mu(v_2),\mu(v_3))\in G} \{\rho((\mu(v_1), \mu(v_2), \mu(v_3)))\}\}.\end{aligned}$$

  Here, for any mapping μ and any constant $c \in I \cup L$, we agree that $\mu(c)$ equals c itself. In other words, mappings are extended to constants according to the identity mapping.
- If P is of the form P_1 UNION P_2, then

$$\begin{aligned}[\![P]\!]_R := \{(\mu, p) \mid (\mu, p_1) \in [\![P_1]\!]_R \text{ or } (\mu, p_2) \in [\![P_2]\!]_R \\ \text{and } p = \max\{p_1, p_2\}\}.\end{aligned}$$

- If P is of the form P_1 AND P_2, then $[\![P]\!]_R := [\![P_1]\!]_R \bowtie [\![P_2]\!]_R$, where, for any two sets of solutions Ω_1 and Ω_2, we define

$$\begin{aligned}\Omega_1 \bowtie \Omega = \{(\mu_1 \cup \mu_2, p) \mid (\mu_1, p_1) \in \Omega_1,\ (\mu_2, p_2) \in \Omega_2,\ \mu_1 \sim \mu_2 \\ \text{and } p = p_1 \cdot p_2\}.\end{aligned}$$

  Here, two mappings μ_1 and μ_2 are called *compatible*, denoted by $\mu_1 \sim \mu_2$, if they agree on the intersection of their domains, i.e., if for every variable $?x \in \mathrm{dom}(\mu_1) \cap \mathrm{dom}(\mu_2)$, we have $\mu_1(?x) = \mu_2(?x)$. And p is the product of p_1 and p_2.
- If P is of the form P_1 DIFF P_2, then $[\![P]\!]_R := [\![P_1]\!]_R \smallsetminus [\![P_2]\!]_R$, where, for any two sets of mappings Ω_1 and Ω_2, we define

$$\Omega_1 \smallsetminus \Omega_2 = \{(\mu_1, p) \in \Omega_1 \mid \neg\exists(\mu_2, p_2) \in \Omega_2 \text{ s.t. } \mu_1 \sim \mu_2\}.$$

- Finally, if P is of the form P_1 FILTER C, then $[\![P]\!]_G := \{(\mu, p) \in [\![P_1]\!]_G \mid \mu(C) = \mathit{true}\}$. Here, for any mapping μ and constraint C, the evaluation of C on μ, denoted by $\mu(C)$, is defined as normal [5].

The following proposition shows that the semantics of pSPARQL is well-defined.

Proposition 1. *For any pSPARQL pattern P, for any probabilistic RDF R, for any solutions $(\mu, p) \in [\![P]\!]_R$, we have $p \in [0, 1]$.*

4 Querying evaluation

A *basic SPARQL query* is an expression of the form $\mathrm{SELECT}_S(P)$ where S is a finite set of variables and P is a pattern. Semantically, given an RDF graph G, we define $[\![\mathrm{SELECT}_S(P)]\!]_G = \{(\mu|_{\mathrm{dom}(\mu)\cap S}, p) \mid (\mu, p) \in [\![P]\!]_G\}$, where we use the common notation $f|_X$ for the restriction of a function f to a subset X of its domain.

Given a probabilistic RDF R, a pSPARQL pattern P, a mapping , a probability $p \in [0, 1]$, the query evaluation problem is to determine whether there exists some probability $p' \in [0, 1]$ with $p' \geq p$ such that $(\mu, p') \in [\![P]\!]_R$.

Proposition 2. *The query evaluation of general pSPARQL patterns is PSPACE-complete.*

Acknowledgments

This work is supported by the program of Applied Mathematics Discipline of Shanghai Polytechnic University (XXKPY1604) and the open funding project of Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education.

References

1. Dalvi, N. & Suciu, D. (2004). Efficient query evaluation on probabilistic databases. In: *Proceedings of the Thirtieth International Conference on Very Large Data Bases (VLDB 2004)*, pp. 864–875.
2. Huang, H. & Liu, C. (2009). Query evaluation on probabilistic RDF databases. In: *Proceedings of the 10th International Conference on Web Information Systems Engineering (WISE 2009)*, pp. 307–320.
3. Kementsietsidis, A., Pema, E., & Tan, W. (2014). Query answering over incomplete and uncertain RDF. In: *Proceedings of the 18th International Workshop on Web and Databases (WebDB 2014)*.
4. Lian, X. & Chen, L. (2011). Efficient query answering in probabilistic RDF graphs. In: *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Date (SIGMOD 2011)*, pp.157–168.
5. Pérez, J., Arenas, M., & Gutierrez, C. (2009). Semantics and complexity of SPARQL. *ACM Transactions on Database Systems*, 34(3):article 16, 2009.
6. Schoenfisch, J. (2014). Querying probabilistic ontologies with SPARQL. In: *Proceedings of the 44th Jahrestagung der Gesellschaft fr Informatik-Komplexität meistern (GI-Jahrestagung 2014)*, pp.2245–2256.