Theodore Andronikos. International Journal of Engineering Research and Applications
www.ijera.com
ISSN: 2248-9622, Vol. 10, Issue 11, (Series-I) November 2020, pp. 23-30

RESEARCH ARTICLE OPEN ACCESS

Complex Matrices for the Approximate Evaluation of Probabilistic Queries

Theodore Andronikos*
**(Department of Informatics, Ionian University, 7 Tsirigoti Square, Corfu, Greece*

ABSTRACT
This paper studies the evaluation of probabilistic SPARQL queries. The evaluation of such queries is, in principle, conceptually simple and straightforward. However, it does incur a time cost that must be taken into account. It thus expedient to devise methods that lower this cost. We propose and explain such a method, which makes use of complex matrices. The idea to resort to the complex numbers is inspired from the greatly expanding field of unconventional computing, and, in particular, from quantum computing. This novel proposal simplifies and speeds up calculation of products of probabilities. Therefore, it is particularly promising in these cases where the same answer set can be obtained either by employing exact computations or by employing suitable approximations, as is indeed the case in many probabilistic queries.
Keywords **-** RDF graph, SPARQL query, probabilistic SPARQL query, complex matrices

---

Date of Submission: 28-10-2020 Date of Acceptance: 09-11-2020

I. INTRODUCTION

During the last two decades numerous researches have focused their attention to all things related to the Web. This tremendous effort has produced state of the art technologies and has literally embedded the World Wide Web into everyone's life. The resounding success of the whole endeavor can be, at least in part, attributed to the adherence to well-designed standards. Linked Open Data [1] provides guidelines regarding the storage and communication of data on the Web. The Resource Description Framework (RDF) is the most prominent standard governing the storage of information, whereas SPARQL deals with retrieving this information, i.e., querying the data.

RDF promotes the use of directed graphs [2] as convenient and effective structures for keeping information. The data contained in the RDF graph obey an explicit syntactic pattern: *subject* $\rightarrow$ *predicate* $\rightarrow$ *object*. The idea behind this is simple and functional: the predicate relates the subject with the object. The rules by which one can query such datasets are stipulated by SPARQL [3]. This work is concerned with probabilistic Regular Path Queries, which have the potential to specify a path of adjacent nodes in the underlying graph. This is achieved through *transitive* predicates. To see how this works, let us start by picturing predicates as labels on the directed edges of the RDF graph. In this setting, a predicate R is defined to be *transitive* if two triples (x, R, y) and (y, R, z) lead to the inference of the triple (x, R, z). Hence, the existence of transitive predicates, makes possible the formation of paths resembling those typically found in standard directed graphs. The paths start from an initial node, follow adjacent directed edges labeled by R, and finally terminate at some other node. It is implicitly assumed that such a path expresses some meaningful property and that this property can be formulated by SPARQL.

Let us now shift the emphasis to the fact that in many situations that data contained in the dataset are approximations, estimations, beliefs, i.e., they lack certainty. The world is uncertain and, in many respects, probabilistic. Therefore, the data stored in an RDF graph may not be totally correct, but probably correct. In domains where uncertainty is prevalent, we may adopt a new perspective, namely that the triplets are assigned a nonnegative real value that indicates the probability of being accurate. It is expedient to assume that this number takes values in the interval [0, 1], as expected from a probability measure. Alternatively, one may well view this number as a weight or even as a degree of certainty. There are many domains of practical importance where probability and uncertainty arise. One important example worth mentioning would be biological data where connections among biological concepts have a probabilistic nature and where additionally the links themselves are statistically independent [4]. It is, thus, evident that tools and techniques must be developed to address the issue of uncertainty.

1.1 Related work

In this subsection, we shall refer to some notable works that are relevant to this paper. This connection should be understood either in the sense that they study theoretical aspects of the RDF and SPARQL formalisms, or that they propose mechanisms that enhance the capabilities of RDF and/or SPARQL. Zhang et al. in [6] studied path queries making use of context-free grammars, which are strictly more powerful than regular expressions. Their primary motivation was the restricted expressiveness of the latter, for example in the case of same generation-queries. They proposed an enhanced query language, aptly named cfSPARQL, which is SPARQL with context-free capability. cfSPARQL, being more expressive than vanilla SPARQL, allows for more interesting queries. Sistla et al. in [7] showed that queries can be understood in terms of automata, and, as a matter of fact, can be formulated using techniques from automata theory. A similar approach was followed in [8] and [9], where Büchi automata and ω-automata were associated to queries on Linked Data and Path Queries, respectively. In the same way, Wang et al. in [10] developed a methodology to answer queries based on automata.

The algebraic characterization of SPARQL queries was undertaken by Schmidt et al. in [11]. The classified queries into equivalence classes and established connections with complexity theory. A different attempt at query classification was conducted in [12], where another, quite distinct, notion of equivalence between SPARQL queries was defined and studied. The possibility of adding additional information (of a temporal nature) was also discussed in [14].

The importance of incorporating probabilities in the RDF datasets and evaluating probabilistic queries has, by now, gained widespread acceptance in the research community. An influential position paper by Reynolds [16] stressed the practical usefulness of dealing with uncertain data and gave pointers about possible solutions that can be employed to overcome difficulties regarding their implementation. Huang and Liu in [4] developed an approach, which, by suitable extending SPARQL, would allow the evaluation of queries on probabilistic databases. They also presented an approximation algorithm that is able to evaluate efficiently path expressions. Likewise, the potential to query data characterized by uncertainty has been studied by many researchers. Hua and Pei in [13] examined probabilistic path queries using a method inspired from dynamic programming. Lian et al. in [15] incorporated probability values to typical datasets, converting them into probabilistic datasets. They also examined certain algorithmic issues and proposed two pruning algorithms. Schoenfisch in [17] examined probabilistic ontologies, as well as the computation of queries on such ontologies. Krompass et al. in [19] studied more general probabilistic databases. For further insight the interested reader is also referred to [19] and references therein.

The query language pSPARQL was proposed in [20] by Fang and Zhang, and was further elaborated by Fang in [21]. This language is an important effort towards the ultimate goal of probabilistic query evaluation. It subsumes vanilla SPARQL and contains additional constructs to handle probabilistic queries. An approach that focuses on the evaluation of probabilistic queries through the use of probabilistic automata is outlined in [22] and in [23]. The current paper is an extension of these works, but with two major differences. First, there is a shift in the perspective from exact evaluation to approximate evaluation. We argue that this is acceptable in a situation involving products of probabilities, as will be seen in the following sections of this paper. Second, and, perhaps, more importantly, we do not use real numbers, but turn to complex numbers. This is inspired from another recent prevalent trend, that of unconventional computing. One aspect of unconventional computing is quantum-inspired computation. Mimicking techniques and concepts from quantum computing can provide new insights into classical problems (see [27], [28] and [29] for an application in optimization problems). Often, replacing a matrix of real numbers by a matrix of complex numbers satisfying some extra property can give better results, as verified experimentally (see for instance [24], [25] and [26] for the use of unitary matrices instead of typical real matrices, as well as extensive comparative test results).

1.2 Contribution

The contribution of this paper lies on its novelty. It demonstrates convincingly that probabilities can be approached via complex numbers, and in particular numbers of the form $e^{i\theta}$, where $\theta \in \mathbb{R}$, which of course immediately implies that $\left|e^{i\theta}\right| = 1$. This facilitates the calculation of products of probabilities, but the end result has to be carefully interpreted. The result would not be exact, only approximate. Still its usefulness cannot be overlooked, because in a setting of uncertainty probabilities close to 0 mean that the assertion is virtually improbable and can be dismissed. Hence, by fixing a threshold, something that is standard practice in probabilistic analysis, we can use the

approximate results for valid, with respect to the agreed threshold, assertions.

II. MOTIVATION

In this section, we shall present an extensive example to motivate the reader and explain the idea behind this work.

Example 1. Suppose we are given the RDF graph shown in Figure 1a. The predicate R is a transitive predicate and the numbers are probabilities.

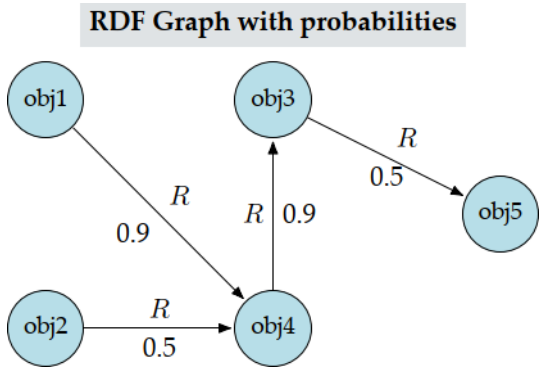

Figure 1a. The RDF graph above shows how objects are related through the transitive predicate R and the degree of certainty associated to each such relation.

```
SELECT ?x
WHERE {
   Obj1 R+ ?x .
}
```

Figure 1b. The above SPARQL query returns the objects that are related to obj1 through an R-path.

The triple $(obj1, R, obj4)$ is assigned probability 0.9, which means that it is almost certain that the relation R holds between obj1 and obj4. On the other hand, the triple $(obj2, R, obj4)$ is associated with probability 0.5, which indicates that relation R may or may not hold between obj2 and obj4 with equal probabilities. The query in Figure 1b uses the syntax of SPARQL 1.1 that makes it possible to express path properties. The meaning of the symbol special symbol + is that there exists an R-path (that is one or more edges labeled by R) between obj1 and every listed object.

Let us now see how this query can be computed using matrices and (column) vectors. Consider the following matrix M_R and vector q_1.

$$M_R = \begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.9 & 0.0 \\ 0.9 & 0.5 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.5 & 0.0 & 0.0 \end{bmatrix} \text{ and } q_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

The matrix M_R represents all the information contained in the RDF graph about predicate R and q_1 is a representation of obj1. The element m_{st} of M_R stores the probability value for the triplet (obj_t, R, obj_s). A value of zero indicates the absence of such a triplet. In this matrix - vector representation we can compute the products $M_R q_1$, $M_R^2 q_1$ and $M_R^3 q_1$, which will give us the following results.

$$M_R q_1 = \begin{bmatrix} 0.0 \\ 0.0 \\ 0.0 \\ 0.9 \\ 0.0 \end{bmatrix}, M_R^2 q_1 = \begin{bmatrix} 0.0 \\ 0.0 \\ 0.81 \\ 0.0 \\ 0.0 \end{bmatrix}, M_R^3 q_1 = \begin{bmatrix} 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.405 \end{bmatrix}$$

A product of the form $M_R^k, k \geq 2$, contains the pairs of objects that related via an R-path of length k. The first product $M_R q_1$ provides no more information than that already shown in the RDF graph. Things get interesting when we look at $M_R^2 q_1$ and $M_R^3 q_1$. These vectors contain the probabilities that obj1 is related to other objects through R-paths of length 2 and 3, respectively. As before, the value zero signifies the absence of such a path, i.e., that there is no connection though predicate R. The actual numbers, reveal the existence of two inferred triplets $(obj1, R, obj3)$ and $(obj1, R, obj5)$, respectively. The former has probability 0.81, which makes the corresponding assertion very likely, while the latter has probability 0.405, which makes the corresponding assertion ambivalent. So, by taking all the above results into account, we can be fairly certain that the inferred assertion $(obj1, R, obj3)$ is true, but cautious about the inferred assertion $(obj1, R, obj5)$. To develop better intuition about the situation, we can repeat the calculations for vector q_2 corresponding to obj2, where $q_2^T = [0 \quad 1 \quad 0 \quad 0 \quad 0]$.

$$M_R q_2 = \begin{bmatrix} 0.0 \\ 0.0 \\ 0.0 \\ 0.5 \\ 0.0 \end{bmatrix}, M_R^2 q_2 = \begin{bmatrix} 0.0 \\ 0.0 \\ 0.45 \\ 0.0 \\ 0.0 \end{bmatrix}, M_R^3 q_2 = \begin{bmatrix} 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.225 \end{bmatrix}$$

Now we observe that the degree of certainty regarding the inferred assertion $(obj2, R, obj5)$ drops to 0.225, i.e., we can be rather sure that is does not hold. These situations always come up in probabilistic scenarios. In order to address them practically and efficiently, it is

common practice to fix a cut-off threshold, or cut-point (see also [30] and [31]). Then, every assertion or fact with probability value below the cut-point is assumed to be false, and may be ignored in any further computation. If we assume for a moment that the cut-point is 0.5, then for normalization purposes all inferred assertion with probability value < 0.5 should be dismissed, i.e., obj1 is not related to obj5, and, similarly, obj2 is related neither to obj3 nor to obj5. The corresponding vector components can be taken to be zero. However, increased accuracy may dictate a smaller threshold, e.g., 0.3. Then only those assertions with probability < 0.3 are neglected. This means that $(obj1, R, obj5)$ and $(obj2, R, obj3)$ are now kept and used for the rest of the computation, and only the inferred assertion $(obj2, R, obj5)$ is dismissed.

Up to this point, the whole approach was conventional. Let us now turn to the unconventional and introduce the complex matrix C_R.

$$C_R = \begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & e^{0.45i} & 0.0 \\ e^{0.45i} & e^{1.04i} & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & e^{1.04i} & 0.0 & 0.0 \end{bmatrix},$$

where, $\begin{matrix} 0.45 = \cos^{-1}(0.9) \\ 1.04 = \cos^{-1}(0.5) \end{matrix}$.

This matrix can be constructed from the RDF graph of Figure 1a, as follows. First, we recall the trivial fact that probabilities are real numbers in the interval $[0, 1]$. As such, a probability value can be considered as the $\cos()$ of an angle θ, that is $p = \cos(\theta)$, which allows us to compute the angle θ as $\theta = \cos^{-1}(p)$. Having retrieved the angle θ, we may go one step further and enter the complex realm by using the formula $e^{i\theta} = \cos\theta + i\sin\theta$. The resulting matrix C_R is equivalent to M_R, only now the probabilities have become angles (in radians). In this way, probabilities 0.9 and 0.5 correspond to $0.45\ rad$ and $1.04\ rad$, respectively. We may now calculate the products $C_R q_1$ and $C_R q_2$.

$$C_R q_1 = \begin{bmatrix} 0.0 \\ 0.0 \\ 0.0 \\ e^{0.45i} \\ 0.0 \end{bmatrix}, and\ C_R q_2 = \begin{bmatrix} 0.0 \\ 0.0 \\ 0.0 \\ e^{1.04i} \\ 0.0 \end{bmatrix}$$

As expected $C_R q_1$ and $C_R q_2$ are just the first and second column of C_R and contain those objects explicitly related to obj1 and obj2, along with the complex numbers $e^{0.45i}$ and $e^{1.04i}$. The probabilities that the triplets $(obj1, R, obj4)$ and $(obj2, R, obj4)$ are valid are given by $\Re(e^{0.45i}) = \cos(0.45) = 0.90$ and $\Re(e^{1.04i}) = \cos(1.04) = 0.50$ respectively. Let us now compute the matrices $C_R^2 q_1$ and $C_R^2 q_2$.

$$C_R^2 q_1 = \begin{bmatrix} 0.0 \\ 0.0 \\ e^{0.90i} \\ 0.0 \\ 0.0 \end{bmatrix}, and\ C_R^2 q_2 = \begin{bmatrix} 0.0 \\ 0.0 \\ e^{1.49i} \\ 0.0 \\ 0.0 \end{bmatrix}$$

The matrix $C_R^2 q_1$ represents the triple $(obj1, R, obj4)$ associated with the complex number $e^{0.90i}$. Two things must be emphasized in this case. First, computing this number involves just one addition instead of one multiplication required by the computation of $M_R^2 q_1$. The advantage of using additions over multiplications is very important, as, in a practical setting, this will scale in proportion to the size of the RDF graph. However, care is needed in the interpretation of the result. A small error arises in the result of adding the angles instead. More precisely, the probability implied by $e^{0.90i}$ is $\Re(e^{0.90i}) = \cos(0.90) = 0.62$, instead of the correct value 0.81. This can be handled in two ways.

One is to fix a cut-point. In the conventional setting the cut-point is a small positive number, e.g., 0.2, 0.3, even 0.5, whose interpretation is that a probability below that threshold means that the inferred fact probably does not hold, and, hence, should be disregarded. In our new complex setting the cut-point is the exponent of $e^{i\theta}$. Specifically, we may fix a cut-point θ_0 (in radians) such that whenever we encounter a complex number $e^{i\theta}$ with $\theta > \theta_0$, we understand that it represents a very small likelihood and the associated triple can be ignored. For example, by setting θ_0 equal to 1.36, or 1.26 or even 1.04 we can simulate the cut-points 0.2, 0.3, and 0.5, respectively. Choosing a cut-point that is suitable for a particular dataset then becomes a matter of deciding which is the acceptable degree of precision for that dataset. If we assume that 0.5 is chosen as an appropriate cut-point, then the corresponding $\theta_0 = 1.04$. Therefore, for the case at hand, the particular $\theta = 0.90$ is less than the cut-point, which implies that the triple $(obj1, R, obj4)$ is considered a valid inference and is accepted.

The other way is to correct the numerical results using a very simple formula containing a correction coefficient and the information provided by the sine of the angle. We propose the formula $\cos\theta + c\sin\theta$, where θ is the angle in question and c is the correction coefficient, as a suitable approximation. The philosophy behind the uncertainty setting is to cope with inexact or approximate information. In such a situation, approximating the degree of uncertainty is the only

reasonable approach. Of course, the correction coefficient must be chosen carefully. Due to the non-linearity of the trigonometric functions, it is impossible to use one correction coefficient for the entire interval $[0, 1]$. The correction coefficient suitable for a specific subinterval must be validated experimentally. Table 1 below shows that to get a good approximation for the probability subinterval $[0.8, 0.9]$ we may use the formula $\cos\theta + c\sin\theta$ with $c = 0.2$. This formula gives for $\theta = 0.90$ the approximate value 0.77, which is very close to the accurate value 0.81. ▲

Table 1. The coefficient for the interval [0.8, 0.9].

Correction coefficient c:					0.2		
p_1	p_2	θ_1	θ_2	$\theta_1+\theta_2$	p_1p_2	$\cos(\theta_1+\theta_2)$	$\cos(\theta_1+\theta_2)+$ $c\ \sin(\theta_1+\theta_2)$
0.9	0.9	0.451	0.451	0.902	0.81	0.62	0.777
0.9	0.95	0.451	0.318	0.769	0.855	0.719	0.858
0.9	1	0.451	0	0.451	0.9	0.9	0.987

III. MAIN RESULTS

In this section, we shall present the ideas outlined in Example 1 in a general and formal way.

Definition 1. Let G be an RDF graph fragment containing a transitive predicate R, and let $1, 2, \ldots, n$, be an arbitrary enumeration of its nodes. We associate to G the $n \times n$ matrix C_R whose elements $c_{st}, 1 \le s, t \le n$, are defined as follows:

$$c_{st} = \begin{cases} 0.0 & if\ (t,R,s)\ does\ not\ exist, \\ e^{i\theta} & if\ (t,R,s)\ exists\ with\ probability\ p = \cos\theta \end{cases}.$$

Definition 2. The cut-point θ_0 is a positive real number, understood to express radians, such that any complex number of the form $e^{i\theta}$ with $\theta > \theta_0$, is simply replaced by 0.0 (zero).

Remark 1. The choice of the cut-point is of great importance because it has the potential to significantly facilitate computations by providing a strict criterion for consistently omitting very small probabilities. This policy has as an immediate and concrete practical impact on the measurable speed-up of the evaluation of probabilistic SPARQL queries. Obviously, datasets of different nature, or increased accuracy requirements, must be taken into account when determining the cut-point. An aggressive approach for increased performance could be to set the cut-point $\theta_0 = 1.04$, corresponding to probability 0.5. Such an approach might also modify the original complex matrix C_R. This is due to the fact that in the above Definition 1, the case that the triple does not exist includes the case where the probability attributed to a triple is below the adopted cut-point. Then, even when the matrix coefficient c_{st} is the nonzero complex number $e^{i\theta}$ (expressing the initial degree of confidence about the corresponding triple), if it happens that $\theta > \theta_0$, then $e^{i\theta}$ must be replaced by 0.0.

In the process of answering queries, starting from the known given matrix C_R, it is inevitable that some matrix power $C_R^k, k \ge 2$, will be computed. The coefficients $c_{st}^k, 1 \le s, t \le n$, of C_R^k have the obvious interpretation. If $c_{st}^k = e^{i\theta}$, then with approximate probability $p = \cos\theta$ there exists an inferred triple (t, R, s) through an R-path of length k or, equivalently, with approximate probability $p = \cos\theta$ there exists an R-path of length k from node t to node s. If $c_{st}^k = 0$, then there exists no inferred triple (t, R, s) through an R-path of length k or, equivalently, there exists no R-path of length k from node t to node s. Remark 1 also applies in this case, i.e., if it happens that $\theta > \theta_0$, where θ_0 is the cut-point, then $e^{i\theta}$ must be replaced by 0.0, something that will add one more zero to the matrix C_R^k. Alternatively, it is worth considering the scenario where $\theta < \theta_0$, which means that θ represents a probability too large to be ignored. In such a case, it is expedient to address the relative error. We caution the reader here that there is no error in the coefficients of C_R, since they express explicit facts taken from the RDF graph itself. The errors occur when we compute the powers C_R^k, where $k \ge 2$. As we have explained in detail in Example 1, the following very simple formula (1) can be used for this purpose.

$$p \approx \cos\theta + c\sin\theta \qquad (1)$$

It is more accurate to approximate the probability that corresponds to the triple (t, R, s) by formula (1). The complex number $e^{i\theta} = \cos\theta + i\sin\theta$ contains the information regarding $\sin\theta$, so it only remains to estimate the correction coefficient c. The non-linearity of the $\cos()$ and $\sin()$ functions preclude the possibility of one correction coefficient for all cases. Instead, the probability interval $[0, 1]$ must be partitioned into smaller subintervals. This can be achieved through extensive experimental tests. Tables 2a - 2d give the suggested correction coefficients for the subintervals $[0.6, 0.7]$, $[0.5, 0.6]$, $[0.4, 0.5]$, and $[0.2, 0.4]$, respectively. The Tables also contain indicative examples of probability values within the respective interval, along with the value of $\cos()$ before the correction of formula (1) is applied.

Table 2a. The coefficient for the interval [0.6, 0.7].

Correction coefficient c:					0.35		
p_1	p_2	θ_1	θ_2	$\theta_1+\theta_2$	p_1p_2	$\cos(\theta_1+\theta_2)$	$\cos(\theta_1+\theta_2)+$ $c\ \sin(\theta_1+\theta_2)$
0.8	0.8	0.644	0.644	1.288	0.64	0.279	0.615
0.8	0.85	0.644	0.555	1.199	0.68	0.363	0.689
0.8	0.9	0.644	0.451	1.095	0.72	0.458	0.769

Table 2b. The coefficient for the interval [0.5, 0.6].

Correction coefficient c:					0.42		
p_1	p_2	θ_1	θ_2	$\theta_1+\theta_2$	p_1p_2	$\cos(\theta_1+\theta_2)$	$\cos(\theta_1+\theta_2)+$ $c\ \sin(\theta_1+\theta_2)$
0.8	0.65	0.644	0.863	1.507	0.52	0.064	0.483
0.8	0.7	0.644	0.795	1.439	0.56	0.131	0.548
0.8	0.75	0.644	0.723	1.367	0.6	0.202	0.614

Table 2c. The coefficient for the interval [0.4, 0.5].

Correction coefficient c:					0.48		
p_1	p_2	θ_1	θ_2	$\theta_1+\theta_2$	p_1p_2	$\cos(\theta_1+\theta_2)$	$\cos(\theta_1+\theta_2)+$ $c\ \sin(\theta_1+\theta_2)$
0.8	0.5	0.644	1.047	1.691	0.4	-0.12	0.357
0.8	0.525	0.644	1.018	1.662	0.42	-0.091	0.387
0.8	0.55	0.644	0.988	1.632	0.44	-0.061	0.418
0.8	0.575	0.644	0.958	1.602	0.46	-0.031	0.449
0.8	0.6	0.644	0.927	1.571	0.48	-0	0.48
0.8	0.625	0.644	0.896	1.54	0.5	0.031	0.511

Table 2d. The coefficient for the interval [0.2, 0.4].

Correction coefficient c:					0.65		
p_1	p_2	θ_1	θ_2	$\theta_1+\theta_2$	p_1p_2	$\cos(\theta_1+\theta_2)$	$\cos(\theta_1+\theta_2)+$ $c\ \sin(\theta_1+\theta_2)$
0.7	0.3	0.795	1.266	2.061	0.21	-0.471	0.103
0.7	0.325	0.795	1.24	2.035	0.228	-0.448	0.134
0.7	0.375	0.795	1.186	1.981	0.263	-0.399	0.197
0.7	0.425	0.795	1.132	1.927	0.298	-0.349	0.26
0.7	0.475	0.795	1.076	1.871	0.333	-0.296	0.325
0.7	0.525	0.795	1.018	1.813	0.368	-0.24	0.391
0.7	0.575	0.795	0.958	1.753	0.403	-0.181	0.458

To biggest benefit of the proposed approach is the fact that the computation of the matrix powers $C_R^k, k \geq 2$, involves only the operation of addition (of angles) and foregoes the operation of multiplication of real numbers that the conventional approach would entail. Using additions instead of multiplications is always preferable, as the compound computational cost scales down significantly with the size of the problem at hand. Hence, this method is undoubtedly better than the conventional. Of course, there is a trade-off, which in this case takes the form of arithmetic errors. There are two ways to remedy this situation. One is the systematic use of a cut-point, a well-established technique in probabilistic scenarios, that can safely and rapidly dismiss inferred fact of small probability, that is facts that are highly unlikely to hold. Another way to deal with this, especially suited to cases where increased accuracy is required, is the application of the correction formula (1). This formula has been validated experimentally through considerable testing, and some characteristics examples are shown in Table1 and Tables 2a - 2d.

IV. CONCLUSION

This paper advocates the evaluation of probabilistic SPARQL queries via unconventional means, namely the use of complex matrices. The extensive example we have presented in section 2 describes in detail the proposed methodology. This novel approach has an unquestionable advantage, and that is that the computation proceeds via additions instead of multiplications. Thus, the potential to scale down the computational cost is real and pragmatic. Finally, we have suggested two ways to handle efficiently the numerical errors that will come-up, either by utilizing a cut-point or my using an approximation formula, which has been thoroughly tested experimentally.

REFERENCES

[1] LOD Project, 2014. Linking Open (LOD) Data Project, http://esw.w3.org/topic/SweoIG/TaskForces/CommunityProjects/LinkingOpenData.

[2] Resource Description Framework (RDF), https://www.w3.org/TR/2015/NOTE-rdfa-primer-20150317/.

[3] SPARQL 1.1 Query Language. Tech. rep., W3C (2013), https://www.w3.org/TR/2013/REC-sparql11-query-20130321/.

[4] Huang, H., Liu, C.: Query evaluation on probabilistic RDF databases. In: *International Conference on Web Information Systems Engineering*, pp. 307–320, Springer (2009).

[5] Hartig, O.: An overview on execution strategies for Linked Data queries, *Datenbank-Spektrum 13(2)*, 89–99 (2013).

[6] Zhang, X., Feng, Z., Wang, X., Rao, G., Wu, W.: Context-free path queries on RDF graphs. In: *International Semantic Web Conference*, pp. 632–648, Springer (2016).

[7] Sistla, A.P., Hu, T., Chowdhry, V.: Similarity based retrieval from sequence databases using automata as queries. In: *Proceedings of the eleventh international conference on Information and knowledge management*, pp. 237–244, ACM (2002).

[8] Giannakis K., Andronikos T., "Querying Linked Data and Büchi Automata", *IEEE Proceedings of the 9th International Workshop on Semantic and Social Media*

Adaptation and Personalization (SMAP), 6-7 November, Corfu, Greece, pp. 110 - 114, 2014.

[9] Giannakis K., Theocharopoulou G., Papalitsas C., Andronikos T., Vlamos P., "Associating ω-automata to Path Queries on Webs of Linked Data", *Engineering Applications of Artificial Intelligence, Elsevier, Volume 51*, May 2016, pages 115-123.

[10] Wang, X., Ling, J., Wang, J., Wang, K., Feng, Z.: Answering provenance-aware regular path queries on RDF graphs using an automata-based algorithm. In: *Proceedings of the 23rd International Conference on World Wide Web*, pp. 395–396, ACM (2014).

[11] Schmidt M., Meier M., Lausen G.: Foundations of SPARQL Query Optimization. In: *Proceedings of the 13th International Conference on Database Theory (ICDT '10)*, pp. 4–33, Lausanne, Switzerland, 2010.

[12] Andronikos, T., "Classification of SPARQL queries into equivalence classes of relevant queries", *International Journal of Advanced Research in Computer Science, December 2017, Volume 8, No. 9*, pages 152-159.

[13] Hua, M., Pei, J.: Probabilistic path queries in road networks: traffic uncertainty aware path selection. In: *Proceedings of the 13th International Conference on Extending Database Technology*, pp. 347–358, ACM (2010).

[14] Andronikos T., Stefanidakis M., Papadakis I., "Adding Temporal Dimension to Ontologies via OWL Reification", *Proceedings of the 13th Panhellenic Conference on Informatics - PCI 2009 Conference*, 10-12 September, Corfu, Greece, IEEE Computer Society, pp. 19-22, 2009.

[15] Lian, X., Chen, L., Wang, G.: Quality-aware subgraph matching over inconsistent probabilistic graph databases. *IEEE Transactions on Knowledge and Data Engineering 28(6)*, 1560–1574 (2016).

[16] Reynolds, D.: Position paper: Uncertainty reasoning for Linked Data. In: *Workshop*, vol. 14 (2014).

[17] Schoenfisch, J.: Querying probabilistic ontologies with SPARQL, *GI-Edition/Proceedings 232*, 2245–2256 (2014).

[18] Krompaß, D., Nickel, M., Tresp, V.: Querying factorized probabilistic triple databases. In: *International Semantic Web Conference*, pp. 114–129. Springer (2014).

[19] Khan, A., Chen, L.: On uncertain graphs modeling and queries, *Proceedings of the VLDB Endowment 8(12)*, 2042–2043 (2015).

[20] H. Fang and X. Zhang, "pSPARQL: a querying language for probabilistic RDF (extended abstract)," *in Proceedings of the ISWC'16, Posters*, 2016.

[21] Fang, H. pSPARQL: A Querying Language for Probabilistic RDF Data Complexity, *Hindawi Limited*, 2019, 1-7.

[22] Andronikos T., Singh A., Giannakis K., Sioutas S., "Computing probabilistic queries in the presence of uncertainty via probabilistic automata", *Algorithmic Aspects of Cloud Computing, Third International Workshop, ALGOCLOUD 2017, Vienna, Austria, 5 September, 2017, Revised Selected Papers. Springer Theoretical Computer Science and General Issues, Volume 10739, pp. 106-122*, ISBN: 978-3-319-74874-0 (Print) 978-3-319-74875-7 (Online), 2018.

[23] Andronikos T., Singh A., Giannakis K., Sioutas S., "Computing probabilistic queries in the presence of uncertainty via probabilistic automata", Proceedings of the 3rd International Workshop on Algorithmic Aspects of Cloud Computing (ALGOCLOUD 2017), 4-8 September, Vienna, Austria, 2017.

[24] Papalitsas C., Giannakis K., Andronikos T., Theotokis D., Sifaleras A., "Initialization methods for the TSP with Time Windows using Variable Neighborhood Search", *6th International Conference on Information, Intelligence, Systems and Applications (IISA)*, 6-8 July, Corfu, Greece, 2015.

[25] Papalitsas C., Andronikos T., Karakostas P., "Studying the impact of perturbation methods on the efficiency of GVNS for the ATSP", *Proceedings of the 6th International Conference on Variable Neighborhood Search (ICVNS 2018)*, 4-7 October, Sithonia, Halkidiki, Greece, 2018.

[26] Papalitsas C., Andronikos T., Karakostas P., "Studying the impact of perturbation methods on the efficiency of GVNS for the ATSP", *6th International Conference, ICVNS 2018, Sithonia, Greece, October 4–7, 2018, Revised Selected Papers. Springer Theoretical Computer Science and General Issues, Volume 11328, pp. 287-302*, ISBN: 978-3-030-15842-2 (Print) 978-3-030-15843-9 (Online), 2019.

[27] Papalitsas C., Andronikos T., "Unconventional GVNS for Solving the Garbage Collection Problem with Time Windows", *(MDPI - Open Access Publishing), Technologies 2019, 7(3), 61*; https://doi.org/10.3390/technologies7030061.

[28] Papalitsas C., Karakostas P., Andronikos T., "A Performance Study of the Impact of

Different Perturbation Methods on the Efficiency of GVNS for Solving TSP", (*MDPI - Open Access Publishing), Applied System Innovation 2019, 2(4), 31*; https://doi.org/10.3390/asi2040031.

[29] Papalitsas C., Andronikos T., Giannakis K., Theocharopoulou G., Fanarioti S., "A QUBO Model for the Traveling Salesman Problem with Time Windows", (MDPI - Open Access Publishing), Algorithms 2019, 12(11), 224; https://doi.org/10.3390/a12110224.

[30] Paz, A.: Introduction to probabilistic automata, *Academic Press, Inc.*, Orlando, FL, USA (1971).

[31] Rabin, M.O.: Probabilistic automata, *Information and Control 6(3)*, 230–245 (1963).

[32] Cormen, T. H.; Leiserson, C. E.; Rivest, R. L. & Stein, C. Introduction to Algorithms, Third Edition, *The MIT Press*, 2009.

Theodore Andronikos*. "Complex Matrices for the Approximate Evaluation of Probabilistic Queries." *International Journal of Engineering Research and Applications (IJERA)*, vol.10 (10), 2020, pp 23-30.