Data & Knowledge Engineering ∎ (∎∎∎∎) ∎∎∎–∎∎∎

Contents lists available at ScienceDirect

Data & Knowledge Engineering

journal homepage: www.elsevier.com/locate/datak

Probabilistic object deputy model for uncertain data and lineage management

Liang Wang [b,c,d], Liwei Wang [a,*], Zhiyong Peng [b,c,**]

[a] International School of Software, Wuhan University, Wuhan, China
[b] State Key Laboratory of Software Engineering, China
[c] Computer School, Wuhan University, Wuhan, China
[d] Center of Computer, Tongji Hospital, Tongji Medical College, Huazhong University of Science and Technology, China

ARTICLE INFO

Article history:
Received 2 March 2017
Received in revised form
2 March 2017
Accepted 2 March 2017

Keywords:
Uncertain data
Data modeling
Lineage
Probability computation

ABSTRACT

Lineage is important in uncertain data management since it can be used for finding out which part of data contributes to a result and computing the probability of the result. Nonetheless, the existing works consider an uncertain tuple as a set of tuples that can be stored in a relational table. Lineage can derive each tuple in the table, with which one can only find out the tuples rather than specific attributes that contribute to the result. If uncertain tuples have multiple uncertain attributes, for a result tuple with low probability, users cannot know which attribute is the main cause of it. In this paper, we propose an approach to model uncertain data. Compared with the alternative way based on the relational model, our model achieves a low maintenance cost and avoids a large number of redundant storage and join operations. Based on our model, some operations are defined for querying data, generating lineage, computing probability and derivation of results. Further, we discuss how to correctly compute probability with lineage and an algorithm is proposed to transform lineage for correct probability computation. We also discuss how to realize result derivation with the lineage. Experiments show the advantages of the proposed model on uncertain data management.

1. Introduction

Probabilistic databases have received considerable interest in recent years, due to their relevance to many applications like data cleaning and integration, information extraction, scientific and sensor data management, and others [1]. Each uncertain value in probabilistic databases has an existence probability to claim its confidence. This uncertainty may propagate during data operations and cause results with low confidence. Hence, data lineage was proposed to identify the derivation of a data item for better confidence understanding [2].

Generally, lineage is defined according to data model and operations on data. Traditional uncertain data management adopted a two-layer approach to manage uncertain data: an underlying logical model and a working model. The logical model [2,3] represents uncertain data with *probabilistic or-set-? tables model* [4] to simultaneously represent tuple-level and attribute-level uncertainty. The working model is defined based on the relational model and an *uncertain tuple* in

* Corresponding author.
** Corresponding author at: State Key Laboratory of Software Engineering, China.
E-mail addresses: nywl@whu.edu.cn (L. Wang), liwei.wang@whu.edu.cn (L. Wang), peng@whu.edu.cn (Z. Peng).

http://dx.doi.org/10.1016/j.datak.2017.03.005
0169-023X/© 2017 Published by Elsevier B.V.

probabilistic or-set-? tables model was stored as a set of tuples. It models uncertainties by assigning each tuple a probability value to assert its existence probability and we call such tuples *possible tuples*. This working model works well when each tuple has only one uncertain attribute. In this situation, every possible tuple is identified by an *ID*, through which we can locate a specific uncertain attribute. However, if tuples contain multiple uncertain attributes, the working model seems to be unsatisfactory. We illustrate the problem in Example 1.

Example 1. Fig. 1(a) exemplifies *probabilistic or-set-? tables model*. For a location, multiple sensors monitor its temperature and humidity. Since different sensors may obtain different data for the same location, the attributes *temperature* and *humidity* are uncertain in the table *Sensor*. An uncertain attribute of a tuple has multiple values and each value has a probability to claim its confidence. The attribute *t_p* is the existence probability of the tuple that at least one of the sensors monitoring the same location works well. We assume that all the probabilities of uncertain attributes have been obtained and the attributes *temperature* and *humidity* are independent. Fig. 1(b) shows how to store the uncertain relation *Sensor* in relational databases. The uncertain tuple with $ID=1$ in Fig. 1(a) has been mapped to four possible tuples identified by attributes *ID* and *ID′*. In Fig. 1(c), for the result tuple (l1, t1, h2) with the demonstrated query, its lineage can be denoted as "Sensor.1.2", which means this result is produced by a possible tuple in *Sensor* whose *ID* is 1 and *ID′* is 2.

In this example, if the probability of a result tuple is low, it means the tuple has a low confidence. The lineage of the result tuple (l1, t1, h2) can only claim that the result is generated by the possible tuple (1, 2, l1, t1, h2). If users want to know which element (temperature or humidity) is the main cause of the low confidence for this result tuple, the ideal answer is "the value h2 of the attribute *humidity* in the uncertain tuple with $ID=1$ leads to the low confidence". However, the existing method can only return a possible tuple as an answer, since the working model of existing methods have transformed attribute-level uncertainty into tuple-level uncertainty. That is, such lineage can be seen as tuple-level lineage and is not adequate in some situations.

To handle attribute-level uncertainty, we need to identify each value of the uncertain attributes in a tuple. When executing a query, it needs to simultaneously obtain identifications of attribute values and produce attribute-level lineages for result tuples. It seems to be reasonable to store attribute values and their identifications together. An alternative way treats each attribute value of an uncertain tuple as a tuple associated with identification and probability value. It is shown in Fig. 1(d) and through join operations we can get possible tuples. Hence it reduces to tuple-level lineage and can be solved by the existing methods. Unfortunately, if we want to obtain a possible tuple, we have to execute the join operation multiple times, which is a time consuming operation. To accelerate query, we can create a materialized view for possible tuples. But the maintenance cost for it is huge. Besides, with the number of uncertain attributes in the same uncertain tuple increasing, the storage space for possible tuples exponentially increases.

In this paper, we adopt Object Deputy Model (ODM) [5] to solve the mentioned problems. ODM has been proved to be more efficient in storage and bilateral links between objects achieve low query and maintenance costs. The properties are natural supports for the solution of storage and computational inefficiency caused by the relational database. Further, due to the fact that ODM is designed for certain data, we extend it to the Probabilistic Object Deputy Model (PODM) to represent data uncertainty. Then, we define a comprehensive set of operations that can get query results while generating data lineages. At last, based on the lineage, we define a probability computation operation for results and propose an algorithm to transform lineage for correct probability computation.

The remaining sections are organized as follows. We review the related work in Section 2 and introduce our working model for uncertain data management in Section 3, following which we elaborate operations on uncertain data and lineage generated by different operations in Section 4. We further discuss how to guarantee correctness during the probability computation in Section 5 and how to derivation from the result lineage in Section 6. A number of experiments are conducted in Section 7 before concluding the paper in Section 8.

2. Related works

In [6], the lineage of an output record was defined as identifying a subset of input records relevant to the output record. In [7], three kinds of lineage (also called provenance) in databases was discussed. "Why" provenance gives the reason the data was generated, say, in the form of a proof tree that locates source data items contributing to its creation. "How" provenance further gives some information about how an output tuple is derived according to the query. "Where" provenance provides an enumeration of the source data items that were actually copied over or transformed to create this data item. The "where" provenance describes the relationships between input and output attributes in tuples, while "why" and "how" provenance describes the relationships between input and output tuples. All the above works mainly focus on lineages in certain data. In uncertain data management, most existing works focus on three aspects about lineage: representation, storage and probability computation.

Trio [2] is the first database simultaneously considering uncertainty and lineage. It assumed all base tuples are independent. In [8], for correlated base tuples, it also utilized lineages to record how to produce result tuples from base tuples. Additionally, it utilized a forest of junction trees to represent correlation between tuples. Combining the junction tree and lineage can correctly compute the probability of a result tuple. Besides the junction tree, in references [9–12], they utilized Bayesian network to model correlations. Our method, which is different from theirs, combines the "where" and "how"

Please cite this article as: L. Wang, et al., (2017), http://dx.doi.org/10.1016/j.datak.2017.03.005