Knowledge-Based Systems 92 (2016) 112–126

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

On efficient conditioning of probabilistic relational databases

Hong Zhu [a], Caicai Zhang [a], Zhongsheng Cao [a,*], Ruiming Tang [b]

[a] School of Computer Science and Technology, Huazhong University of Science and Technology, 1037 Luoyu Road, Wuhan 430074, China
[b] Huawei Noah's Ark Lab, Hong Kong

ARTICLE INFO

Article history:
Received 11 February 2015
Revised 10 October 2015
Accepted 15 October 2015
Available online 26 October 2015

Keywords:
Probabilistic databases
Possible worlds
Conditioning
Functional dependency
Constraints

ABSTRACT

A probabilistic relational database is a probability distribution over a set of deterministic relational databases (namely, possible worlds). Efficient updating information in probabilistic databases is required in several applications, such as sensor networking and data cleaning. As a way to update a probabilistic database, conditioning refines the probability distribution of the possible worlds based on general knowledge, such as functional dependencies. The existing methods for conditioning are exponential over the number of variables in the probabilistic database for an arbitrary constraint. In this paper, a constraint-based conditioning framework is proposed, which solves the conditioning problem by considering only the variables in the given constraint. Then, we prove the correctness of our proposed approach and provide efficient algorithms for each step of the approach. Afterward, a pruning strategy that can significantly improve the efficiency of the constraint-based approach is proposed for the functional dependency constraints. Furthermore, for functional dependency constraints, a variable-elimination strategy that minimizes the number of generated variables can benefit the subsequent query processing. The experimental study shows that the constraint-based approach is more efficient than other approaches described in the literature. The effectiveness of the two optimization strategies for functional dependency constraints is also demonstrated in the experiment.

1. Introduction

Deterministic databases were invented to support applications that require precise data semantics, such as banking, payroll, and accounting. However, modern applications need to process uncertain data that are retrieved from diverse and autonomous sources [1], such as data cleaning [2,3], sensor networks [4,5], tracking moving objects [6,7], or healthcare information systems [8]. For example, in the context of data cleaning [9], many real-world applications accurately merge a number of duplicate records. Fully eliminating erroneous duplicates is still an exhausting human labor intensive process. Furthermore, full deduplication could result in the loss of valuable information. Probabilistic databases are an alternative approach to keeping all of the candidates (and their probabilities) that potentially refer to the same entity. Informally, a probabilistic database is a probability distribution over a set of deterministic databases (namely, possible worlds) [10].

Efficient updating in probabilistic databases is required in several applications. Continuous learning and surveying introduce significant new rules, such as statistical relational learning and domain knowledge from experts. Statistical relational learning might learn some rules as conditioning constraints, and then, the constraints can be used to update the probabilistic databases. Updating probabilistic databases based on new rules is important to a wide range of applications. For example, the authors of [11] agree that modern knowledge bases such as Yago [12], DeepDive [13], and Google's Knowledge Vault [14] are constructed from large corpora of text by using some form of supervised information extraction. The extracted data usually starts as a large probabilistic database; then, its accuracy is improved by adding domain knowledge expressed as constraints. In the context of data cleaning, as stated by the authors of [15], "it is only natural to start with a probabilistic database and clean it – reduce uncertainty – by adding constraints or additional information". Several rules are specified to remove impossible value assignments of the data. In the context of sensor networks [16], sensor readings are usually represented using correlated probabilistic models. Many applications benefit from updating correlated sensor data, for example, obtaining the values of the other correlated sensors by figuring out the up-to-date value of one sensor. Similar applications exist in data integration [17].

Conditioning a probabilistic database with a constraint is performed by removing all of the possible worlds that do not satisfy the given constraint. Then, the probability of each remaining possible world is refined according to the conditional probability of the possible worlds when the constraint is true. The conditioning problem is to find such a probabilistic database that assigns the possible

* Corresponding author. Tel.: +86 13807192568.
E-mail addresses: zhuhong@hust.edu.cn (H. Zhu), caicaizhng@gmail.com (C. Zhang), caozhongsheng@163.com (Z. Cao), tangruiming@huawei.com (R. Tang).

http://dx.doi.org/10.1016/j.knosys.2015.10.017
0950-7051/

Table 1
A probabilistic database $\mathbb{D}$.

(a) R_1

RID	Image ID	Box ID	Face	f
t_1	Image1	Box1	Mary	e_1
t_2	Image1	Box1	Lily	$\neg e_1$
t_3	Image1	Box2	Mary	e_2
t_4	Image2	Box1	Tom	e_3

(b) V_P

V	P
e_1	0.4
e_2	0.5
e_3	0.8

Table 2
Eight possible worlds of $\mathbb{D}$.

$w_i(e_1, e_2, e_3)$	R_1	$P(w_i)$
$w_0(0, 0, 0)$	$\{t_2\}$	0.06
$w_1(0, 0, 1)$	$\{t_2, t_4\}$	0.24
$w_2(0, 1, 0)$	$\{t_2, t_3\}$	0.06
$w_3(0, 1, 1)$	$\{t_2, t_3, t_4\}$	0.24
$w_4(1, 0, 0)$	$\{t_1\}$	0.04
$w_5(1, 0, 1)$	$\{t_1, t_4\}$	0.16
$w_6(1, 1, 0)$	$\{t_1, t_3\}$	0.04
$w_7(1, 1, 1)$	$\{t_1, t_3, t_4\}$	0.16

worlds that satisfy the given constraint with new probabilities. We use an example to illustrate the conditioning problem.

Example 1. Social networks such as Facebook and Flickr store digital photographs for users. Facebook provides the functionality of face detection, i.e., when a user moves the mouse over a person's face in her photograph, a box shows up to highlight the face. For each photograph, one or more bounding boxes represent faces, using face detection. The face recognition system recognizes whose face it is in each box. Let us consider that there is an automatic face recognition system for social networks. The face recognition information cannot be modeled by a traditional relational database because they are uncertain and could be correlated. We use a probabilistic database to represent such information.

The Boolean expression that is associated with a tuple, as shown in column f of Table 1(a), is interpreted as the condition that the tuple be correct. The probability that is associated with a Boolean variable, as shown in Table 1(b), is interpreted as the probability of the variable being true. The probabilities of the Boolean variables induce the probabilities of the tuples to be correct. For example, the probability that *Box 1* in *Image 1* is *Mary* is $P(e_1) = 0.4$, and the probability that it is *Lily* is $P(\neg e_1) = 0.6$.

Each assignment of variables in V_P represents one possible world of $\mathbb{D}$: this possible world contains all of the tuples whose associated formulae are satisfied by the given assignment. Table 2 shows eight possible worlds of R_1, where $w_i(e_1, e_2, e_3)$ is the ith possible world, and $P(w_i)$ is the corresponding probability of w_i. Although R_1 appears to be a traditional relational database, its semantics involve a set of possible worlds, which is different from a RDB. Any queries over probabilistic relational databases are semantically applied to each possible world [18].

Let us consider the additional knowledge that a person can be in a photograph at most once. This constraint implies that t_1 and t_3 cannot exist at the same time. Conditioning the probabilistic database with this constraint should have the effect of filtering out those possible worlds that do not satisfy this constraint. Here, w_6 and w_7 do not satisfy this constraint. The conditioned probabilistic database is obtained by removing these two possible worlds. The probabilities of the remaining possible worlds in the conditioned probabilistic database are conditional probabilities in the same sample space [15,19]. For example, the probability of w_3 in the conditioned probabilistic database is $P(w_3|C) = P(w_3 \wedge C)/P(C) = 0.24/0.8 = 0.3$, where $P(C)$ is the sum of the probabilities of those possible worlds that satisfy C.

The conditioning problem is to find a probabilistic database that represents the set of the valid possible worlds according to the given constraint, with their new probabilities.

The conditioning problem in probabilistic relational databases has been studied in [15,19,20]. Koch and Olteanu [15] show that conditioning is NP-Hard. They develop ws-trees to capture constraint information. However, it is not easy to construct a tree efficiently to compute the probability of the represented constraint and perform conditioning. In [19,20], Tang et al. identify tractable scenarios for which they devise polynomial time algorithms to perform conditioning. They focus on the special cases in which the tuples are independent, and the constraints considered are observation constraints, X-tuple constraints, and referential constraints. However, one of the challenges in conditioning probabilistic databases is that the existing methods are exponential over the number of variables in the probabilistic database for an arbitrary constraint as a result of enumerating possible worlds. For a probabilistic database that has n_R tuples and n_E variables involved in the formulae of tuples, the time complexity of the existing conditioning method is $O(n_R \cdot 2^{n_E})$ for an arbitrary constraint.

Functional dependency constraints are a common type of constraint in relational databases. A functional dependency constraint in a probabilistic database can be transformed into a set of mutually exclusive constraints (several tuples are exclusive). For a mutually exclusive constraint over correlated uncertain data, the time complexity of the existing conditioning method is also $O(n_R \cdot 2^{n_E})$.

Due to the high time complexity of the existing conditioning method, we propose a constraint-based conditioning framework to tackle this challenge under a general uncertainty model that allows for probabilistic correlations. This framework is based on two ideas: (1) considering the assignments of variables appearing in the constraint only, instead of enumerating the possible assignments of all of the variables in the formulae of tuples in the probabilistic database; and (2) applying variable-replaced mechanisms to update the formulae of tuples, while avoiding the replacement of every variable in the probabilistic database. In this paper, we provide efficient algorithms for each step of the constraint-based conditioning framework.

For functional dependency constraints, to further improve the efficiency of the constraint-based conditioning framework, we seek optimization strategies that are based on two ideas: (1) *FD_C_based*: for functional dependency constraints, we devise a pruning strategy to further improve the efficiency of the constraint-based conditioning approach; (2) *FD_VE_C_based*: based on *FD_C_based*, to reduce the number of generated variables, multiple satisfactory assignments of the constraint that lead to the same possible world are identified and combined.

Our main contributions in this article are summarized as follows.

(1) For an arbitrary constraint, due to the high time complexity of the existing conditioning method, we propose a constraint-based conditioning framework to tackle this challenge under a general uncertainty model that allows for probabilistic correlations. This framework minimizes the set of tuples whose formulae must be updated.
(2) We prove the correctness of the proposed constraint-based conditioning algorithm by showing that it finds an equivalent representation of a conditioned probabilistic database. We provide efficient algorithms for each step of the constraint-based conditioning framework.
(3) We propose two optimization strategies for functional dependency constraints: a pruning strategy further improves the efficiency of the constraint-based approach, by pruning some impossible assignments as early as possible, and a variable-elimination strategy minimizes the number of variables in the conditioned probabilistic database.