Journal of Computer and System Sciences 80 (2014) 1448–1489

Contents lists available at ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

Consistency checking and querying in probabilistic databases under integrity constraints

Sergio Flesca *, Filippo Furfaro, Francesco Parisi

DIMES, University of Calabria, Via Bucci, Rende (CS), Italy

ARTICLE INFO

Article history:
Received 13 March 2013
Received in revised form 11 April 2014
Accepted 16 April 2014
Available online 24 April 2014

Keywords:
Probabilistic databases
Integrity constraints
Consistency checking

ABSTRACT

We address the issue of incorporating a particular yet expressive form of integrity constraints (namely, denial constraints) into probabilistic databases. To this aim, we move away from the common way of giving semantics to probabilistic databases, which relies on considering a unique interpretation of the data, and address two fundamental problems: *consistency checking* and *query evaluation*. The former consists in verifying whether there is an interpretation which conforms to both the marginal probabilities of the tuples and the integrity constraints. The latter is the problem of answering queries under a "cautious" paradigm, taking into account all interpretations of the data in accordance with the constraints. In this setting, we investigate the complexity of the above-mentioned problems, and identify several tractable cases of practical relevance.

1. Introduction

Probabilistic databases (PDBs) are widely used to represent uncertain information in several contexts, ranging from data collected from sensor networks, data integration from heterogeneous sources, bio-medical data, and, more in general, data resulting from statistical analyses. In this setting, several relevant results have been obtained regarding the evaluation of conjunctive queries, thanks to the definition of probabilistic frameworks dealing with two substantially different scenarios: the case of *tuple-independent* PDBs [1,2], where all the tuples of the database are considered independent one from another, and the case of PDBs representing probabilistic networks encoding even complex forms of correlations among the data [3]. However, none of these frameworks takes into account integrity constraints in the same way as it happens in the deterministic setting, where constraints are used to enforce the consistency of the data. In fact, the former framework strongly relies on the independence assumption (which clearly is in contrast with the presence of the correlations entailed by integrity constraints). The latter framework is closer to an AI perspective of representing the information, as it requires the correlations among the data to be represented as data themselves. This is different from the DB perspective, where constraints are part of the schema, and not of the data.

In this paper, we address the issue of incorporating integrity constraints into probabilistic databases, with the aim of extending the classical semantics and usage of integrity constraints of the deterministic setting to the probabilistic one. Specifically, we consider one of the most popular logical models for the probabilistic data, where information is represented into tuples associated with probabilities, and give the possibility of imposing *denial constraints* on the data, i.e., constraints forbidding the co-existence of certain tuples. In our framework, the role of integrity constraints is the same as

* Corresponding author.
E-mail addresses: flesca@dimes.unical.it (S. Flesca), furfaro@dimes.unical.it (F. Furfaro), fparisi@dimes.unical.it (F. Parisi).

http://dx.doi.org/10.1016/j.jcss.2014.04.026
0022-0000/

	Id	*Hid*	*Price*	*Type*	*View*	*P*
t_1	1	1	120	Std	Sea	p_1
t_2	2	1	70	Suite	Courtyard	p_2
t_3	3	1	120	Std	Sea	p_3

Fig. 1. Relation instance $room^p$.

in the deterministic setting: they can be used to decide whether a new tuple can be inserted in the database, or to decide (a posteriori w.r.t. the generation of the data) if the data are consistent.

Before explaining in detail the main contribution of our work, we provide a motivating example, which clarifies the impact of augmenting a PDB with (denial) constraints. In particular, we focus on the implications on the consistency of the probabilistic data, and on the evaluation of queries. We assume that the reader is acquainted with the data representation model where uncertainty is represented by associating tuples with a probability, and with the notion of possible world (however, these concepts will be formally recalled in the first sections of the paper).

Motivating example. Consider the PDB schema $\mathcal{D}^p$ consisting of the relation schema $Room^p(Id, Hid, Price, Type, View, P)$, and its instance $room^p$ in Fig. 1.

Every tuple in $room^p$ is characterized by the room identifier *Id*, the identifier *Hid* of the hotel owning the room, its price per night, its type (e.g., "Standard", "Suite"), and the attribute *View* describing the room view. The attribute P specifies the probability that the tuple is true. For now, we leave the probabilities of the three tuples as parameters (p_1, p_2, p_3), as we will consider different values to better explain the main issues related to the consistency and the query evaluation.

Assume that the following constraint ic is defined over $\mathcal{D}^p$: "*in the same hotel, standard rooms cannot be more expensive than suites*". This is a denial constraint, as it forbids the coexistence of tuples not satisfying the specified property. In particular, ic entails that t_1 and t_2 are mutually exclusive, as, according to t_1, the standard room 1 would be more expensive than the suite room 2 belonging to the same hotel as room 1. For the same reason, ic forbids the coexistence of t_2 and t_3.

Finally, consider the following query q on $\mathcal{D}^p$: "*Are there two standard rooms with sea view in hotel* 1?". We now show how the consistency of the database and the answer to q vary when changing the probabilities of $room^p$'s tuples.

Case 1 *(No admissible interpretation).* $p_1 = \frac{3}{4};\ p_2 = \frac{1}{2};\ p_3 = \frac{1}{2}$.

In this case, we can conclude that the database is inconsistent. In fact, ic forbids the coexistence of t_1 and t_2, which means that the possible worlds containing t_1 must be distinct from those containing t_2. But the marginal probabilities of t_1 and t_2 do not allow this: the fact that $p_1 = \frac{3}{4}$ and $p_2 = \frac{1}{2}$ implies that the sum of the probabilities of the worlds containing either t_1 or t_2 would be $\frac{3}{4} + \frac{1}{2}$, which is greater than 1.

Case 2 *(Unique admissible interpretation).* $p_1 = \frac{1}{2};\ p_2 = \frac{1}{2};\ p_3 = \frac{1}{2}$.

In this case, the database is consistent, as it represents two possible worlds: $w_1 = \{t_1, t_3\}$ and $w_2 = \{t_2\}$, both with probability $\frac{1}{2}$ (correspondingly, the possible worlds representing the other subsets of $\{t_1, t_2, t_3\}$ have probability 0). Observe that there is no other way to interpret the database, while making the constraint satisfied in each possible world, and the probabilities of the possible worlds compatible w.r.t. the marginal probabilities of t_1, t_2, t_3. Thus, the database is consistent and has a unique admissible interpretation.

Now, evaluating the above-defined query q over all the admissible interpretations of the database yields the answer *true* with probability $\frac{1}{2}$ (which is the probability of w_1, the only non-zero-probability world, in the unique admissible interpretation, where q evaluates to *true*). Note that, if ic were disregarded and q were evaluated using the independence assumption, the answer to q would be *true* with probability $\frac{1}{4}$.

Case 3 *(Multiple admissible interpretations).* $p_1 = \frac{1}{2};\ p_2 = \frac{1}{4};\ p_3 = \frac{1}{2}$.

In this case, we can conclude that the database is consistent, as it admits at least the interpretations I_1 and I_2 represented in the two rows of the following table (each cell is the probability of the possible world reported in the column header).

	$\emptyset$	$\{t_1\}$	$\{t_2\}$	$\{t_3\}$	$\{t_1, t_2\}$	$\{t_1, t_3\}$	$\{t_2, t_3\}$	$\{t_1, t_2, t_3\}$
I_1	0	1/4	1/4	1/4	0	1/4	0	0
I_2	1/4	0	1/4	0	0	1/2	0	0