Fortuna: A Visualization Tool for Probabilistic Cardinality Constraints

Tania Roblot, Sebastian Link

Department of Computer Science, University of Auckland, New Zealand
[tkr|s.link]@auckland.ac.nz

Abstract. Probabilistic cardinality constraints stipulate lower bounds on the marginal probability of cardinality constraints in probabilistic databases. The demo shows how the computation of Armstrong PC-sketches helps design teams identify lower bounds that separate meaningful from meaningless probabilistic databases in an application domain.

Keywords: probabilistic database; cardinality constraint; Armstrong database

1 Introduction

Background. Cardinality constraints are fundamental for understanding the structure and semantics of data. They were introduced in Chen's seminal paper [2] and have attracted interest and tool support ever since. A cardinality constraint $card(X) \leq b$ is satisfied by a relation r iff r does not contain more than b different tuples that all have matching values on all the attributes in X. Consider a wireless sensor network in which we record the RFID tag number of wolverines along with the time and zone of their sighting within a year. We may specify the cardinality constraint $card(RFID, Zone) \leq 365$ which says that the same wolverine is sighted in the same zone up to 365 times a year. Today, uncertain data is at the core of an increasing number of modern applications. To help manage the requirements of such applications, probabilistic databases emerged. A probabilistic relation $r = (\{W1, \ldots, Wn\}, P)$ is a probability distribution P over a finite set $\{W1, \ldots, Wn\}$ of relations, each representing a possible world [8]. Recently, probabilistic cardinality constraints (pCCs) were proposed [5,6] as extensions of both probabilistic keys [1] and traditional cardinality constraints. A pCC $(card(X) \leq b, \geq p)$ is satisfied by a probabilistic relation $r = (\{W1, \ldots, Wn\}, P)$ iff the marginal probability of $card(X) \leq b$ in r is at least p, that is, the sum of the probabilities $P(Wi)$ of the worlds Wi that satisfy $card(X) \leq b$ is at least p. In our example, we specify the following set Σ of pCCs: $(card(\text{RFID}, \text{Zone}) \leq 365, \geq 1)$, $(card(\text{RFID}, \text{Time}) \leq 1, \geq 1)$ and $(card(\text{Time}, \text{Zone}) \leq 2, \geq 0.5)$. The contributions of [5,6] were the proposal of pCCs, the demonstration of their usefulness in data quality and query estimation, axiomatic and algorithmic characterizations of their implication problem, and an algorithm for computing for any given finite set Σ of pCCs, a single Armstrong PC-sketch for Σ.

A PC-sketch is Armstrong for Σ iff for every pCC φ, Σ implies φ iff the Armstrong PC-sketch satisfies φ. In particular, for every cardinality constraint $card(X) \leq b$ the largest probability p such that Σ implies $(card(X) \leq b, \geq p)$ coincides with the marginal probability of $card(X) \leq b$ in an Armstrong PC-sketch for Σ. This property is appealing during requirements acquisition where design teams must identify constraints that separate meaningful from meaningless databases in the application domain. For the set Σ of pCCs in our running example, Figure 1 shows a probabilistic Armstrong sketch where $P(W1) = P(W2) = 0.5$. Every tuple of a sketch has actual domain values on a given attribute set, the symbol $*$ on all remaining attributes, and a cardinality that says how many different tuples with the domain values on the attribute set it represents. For instance, (365,v_RFID,4,∗,v_Zone,4) represents 365 tuples that all have the value v_RFID,4 on attribute *RFID*, the value v_Zone,4 on attribute *Zone*, and unique values on attribute *Time*. Sketches finitely represent infinite worlds which result from attributes for which no finite bound has been specified with probability 1. For example, (INFTY,∗,∗,v_Zone,1) indicates that no finite bound has been specified on *Zone*.

Fig. 1. Armstrong sample

Possible World W1

card	RFID	Time	Zone
INFTY	*	*	v_Zone,1
INFTY	v_RFID,2	*	*
INFTY	*	v_Time,3	*
365	v_RFID,4	*	v_Zone,4
2	*	v_Time,5	v_Zone,5

Possible World W2

card	RFID	Time	Zone
INFTY	*	*	v_Zone,1
INFTY	v_RFID,2	*	*
INFTY	*	v_Time,3	*
365	v_RFID,4	*	v_Zone,4
INFTY	*	v_Time,5	v_Zone,5

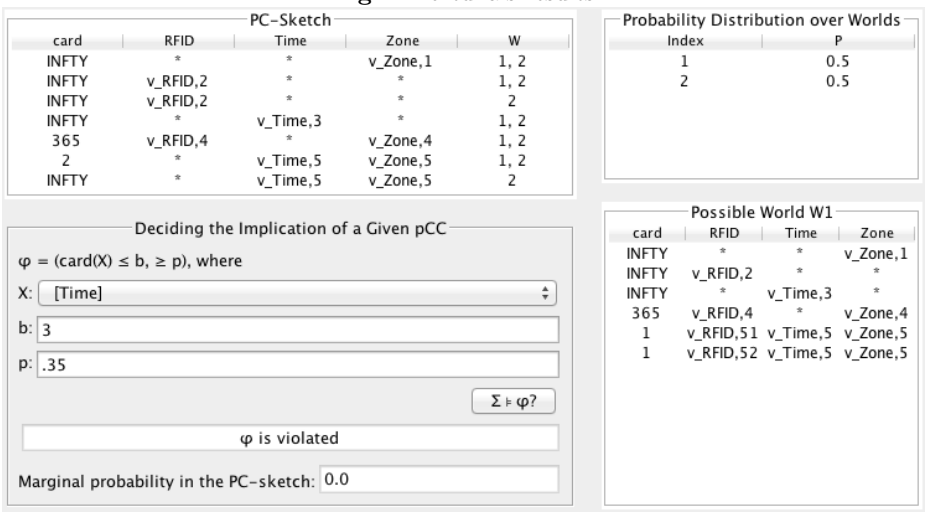
Contribution. The demonstration showcases our tool *Fortuna* which computes Armstrong PC-sketches for finite sets of pCCs [5,6]. The main contribution is Fortuna's ability to transfer the concepts of pCCs and their Armstrong PC-sketches into the practice of requirements engineering. The demonstration illustrates how Fortuna facilitates communication between design teams and domain experts, showing how users identify more meaningful probabilistic cardinality constraints.

Organization. We discuss novelty in Section 2. The GUI is presented in Section 3. The demonstration is outlined in Section 4. We conclude in Section 5.

2 Related Systems and Novelty

Inspired by the design-by-example paradigm of Armstrong databases [3,4,7], our tool allows the user to specify a set Σ of pCCs for which an Armstrong PC-sketch will be computed by *Fortuna*. Fortuna functions as an oracle to translate sets of pCCs, currently perceived as meaningful by the designers, into a concise representation of a probabilistic relation which can be inspected jointly with domain experts to point out flaws and shortcomings. While there is a plethora of research about Armstrong databases for constraints on certain data, see [3,5,6] for references, Fortuna is the first tool to compute Armstrong databases for constraints on uncertain data. Tool support seems even more important for probabilistic constraints, since not only the right bounds but also the right marginal probabilities must be identified.

Fig. 2. Fortuna's results



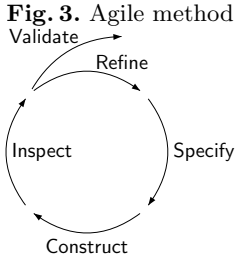
3 Application

Fortuna is a GUI developed in Java 1.7, and is available for download from `cs.auckland.ac.nz/~tkr/fortuna.html`. Users can specify the set Σ from the GUI's form or a file. Fortuna computes an Armstrong PC-sketch for Σ as per [6]. Figure 2 shows the output on our running example. The two possible worlds from Figure 1 are represented as a single Armstrong PC-sketch. Users can check for any pCC if it is implied by Σ. For example, $(card(\text{Time}) \leq 3, \geq 0.35)$ is violated as $card(\text{Time}) \leq 3$ has marginal probability 0 in the PC-sketch.

4 Demonstration

The demonstration showcases how Fortuna aids with the visualization of missing but semantically meaningful constraints thanks to a simple use case scenario: our running example. The audience is given the role of the design team while the demonstrators act as domain experts. The process is agile, facilitating interaction and iteration as shown in Figure 3: the designers *specify* pCCs they consider meaningful for the given scenario; next they feed those into Fortuna to *construct* an Armstrong PC-sketch that visualizes the designer's perception for the domain experts; the domain experts then *inspect* the sketch and provide feedback to the designers who *refine* their constraints accordingly. After inspecting the Armstrong PC-sketch of Figure 2, the domain experts may be concerned about the occurrences

Fig. 3. Agile method



of INFTY in the sketch. Discussion may result in the specification of additional cardinality constraints, such as for the number of times each wolverine can be sighted. This process is iterated until the domain experts *validate* the perceptions of the designers. Thus the audience gets to simulate the real use of Fortuna to experience how well it facilitates the communication between designers and domain experts, by pinpointing flaws and shortcomings and allowing on-the-fly testing and corrections.

5 Conclusions

As the Einstein-accredited quote states: "Example is not another way to teach, it is the only way to teach." The design-by-example paradigm is a natural approach which inspired the creation of *Fortuna*: a GUI that constructs an Armstrong PC-sketch that perfectly visualizes the marginal probabilities of any set of pCCs. Fortuna can easily be used to overcome any mismatch in expertise by facilitating the communication of pCCs to stakeholders of the target probabilistic database. Any party involved with probabilistic databases should welcome the integration of cardinality constraints. Our work is a first and major step towards making this new notion accessible to database practice. Future work should look into introducing other integrity constraints to Fortuna.

References

1. Brown, P., Link, S.: Probabilistic keys for data quality management. In: Zdravkovic, J., Kirikova, M., Johannesson, P. (eds.) Advanced Information Systems Engineering - 27th International Conference, CAiSE 2015, Stockholm, Sweden, June 8-12, 2015, Proceedings. Lecture Notes in Computer Science, vol. 9097, pp. 118–132. Springer (2015)
2. Chen, P.P.: The Entity-Relationship model - toward a unified view of data. ACM Trans. Database Syst. 1(1), 9–36 (1976)
3. Hartmann, S., Kirchberg, M., Link, S.: Design by example for SQL table definitions with functional dependencies. VLDB J. 21(1), 121–144 (2012)
4. Mannila, H., Räihä, K.J.: Design by example: An application of Armstrong relations. J. Comput. Syst. Sci. 33(2), 126–141 (1986)
5. Roblot, T., Link, S.: Probabilistic cardinality constraints. Tech. Rep. 481, `https://www.cs.auckland.ac.nz/research/groups/CDMTCS/researchreports/` (2015)
6. Roblot, T., Link, S.: Probabilistic cardinality constraints. In: Johannesson, P., Lee, M., Liddle, S., Pastor, O. (eds.) Conceptual Modelling - 34th International Conference, ER 2015, Stockholm, Sweden, October 19-22, 2015, Proceedings. Lecture Notes in Computer Science, Springer (2015)
7. Silva, A., Melkanoff, M.: A Method for Helping Discover the Dependencies of a Relation, pp. 115–133. Springer US (1981), `http://dx.doi.org/10.1007/978-1-4615-8297-7_5`
8. Suciu, D., Olteanu, D., Ré, C., Koch, C.: Probabilistic Databases. Synthesis Lectures on Data Management, Morgan & Claypool Publishers (2011)