Probabilistic Keys for Data Quality Management

Pieta Brown and Sebastian Link(✉)

Department of Computer Science, University of Auckland, Auckland, New Zealand
{pieta.brown,s.link}@auckland.ac.nz

Abstract. Probabilistic databases address well the requirements of an increasing number of modern applications that produce large volumes of uncertain data from a variety of sources. We propose probabilistic keys as a principled tool helping organizations balance the consistency and completeness targets for their data quality. For this purpose, algorithms are established for an agile schema- and data-driven acquisition of the marginal probability by which keys should hold in a given application domain, and for reasoning about these keys. The efficiency of our acquisition framework is demonstrated theoretically and experimentally.

Keywords: Acquisition · Key · Probability · Quality · Visualization

1 Introduction

Background. The notion of a key is fundamental for understanding the structure and semantics of data. For relational databases, keys were already introduced in Codd's seminal paper [6]. Here, a key is a set of attributes that holds on a relation if there are no two different tuples in the relation that have matching values on all the attributes of the key. Keys uniquely identify tuples of data, and are applied in data cleaning, integration, modeling, processing, and retrieval.

Motivation. Relational databases target applications with certain data, such as accounting, inventory and payroll. Modern applications, such as data integration, information extraction, and financial risk assessment produce large volumes of uncertain data from a variety of sources. For instance, RFID (radio frequency identification) is used to track movements of endangered species of animals, such as wolverines. Here it is sensible to apply probabilistic databases. Table 1 shows a probabilistic relation (p-relation), which is a probability distribution over a finite set of possible worlds, each being a relation.

Keys address the consistency dimension of data quality in traditional databases. Due to the veracity inherent to probabilistic databases as well as the variety of sources the data originates from, the traditional concept of a key requires revision in this context. In our example, for instance, there is no non-trivial key that is satisfied by all possible worlds: the key $k1 = k\{time, zone\}$ holds in the worlds W_1 and W_2, $k2 = k\{rfid, time\}$ holds in W_2 and W_3, and $k3 = k\{rfid, zone\}$ holds in W_3 and W_4. One may argue to remove possible worlds that violate a key but this would neither address the completeness dimension of

J. Zdravkovic et al. (Eds.): CAiSE 2015, LNCS 9097, pp. 118–132, 2015.
DOI: 10.1007/978-3-319-19069-3_8

Table 1. Probabilistic relation

W_1 $(p_1 = 0.2)$			W_2 $(p_2 = 0.45)$			W_3 $(p_3 = 0.3)$			W_4 $(p_4 = .05)$		
rfid	*time*	*zone*	*rfid*	*time*	*zone*	*rfid*	*time*	*zone*	*rfid*	*time*	*zone*
w1	2pm	z1	w1	2pm	z1	w1	2pm	z1	w1	3pm	z1
w1	3pm	z1	w1	3pm	z1	w1	3pm	z2	w1	3pm	z2
w1	3pm	z2	w2	3pm	z2	w2	3pm	z2	w2	3pm	z2

data quality nor would it make sensible use of probabilistic databases. Instead, we propose the new concept of a *probabilistic key*, or p-key for short, which stipulates a lower bound on the marginal probability by which a traditional key holds in a probabilistic database. In our example, $k1$, $k2$, and $k3$ have marginal probability 0.65, 0.75, and 0.35, respectively, which is the sum of the probabilities of those possible worlds which satisfy the key. Indeed, the marginal probability of a key provides a control mechanism to balance consistency and completeness targets for the quality of data. Larger marginal probabilities represent stricter consistency and more liberal completeness targets, while smaller marginal probabilities represent more liberal consistency and stricter completeness targets. Having fixed these targets in the form of a marginal probability, p-keys can be utilized to control these data quality dimensions during updates. When new data arrives, p-keys can help detect anomalous patterns of data in the form of p-key violations. That is, alerts can be automatically sent out when a data set would not meet a desired lower bound on the marginal probability of a key. In a different showcase, p-keys can also be used to infer probabilities that query answers are unique. In our example, we may wonder about the chance that different wolverines are in the same zone at the same time, indicating potential mating behavior. We may ask

SELECT DISTINCT *rfid* FROM Tracking WHERE *zone*='z2' AND *time*='2pm'

and using our p-keys enables us to derive a minimum probability of 0.65 that a unique answer is returned, that is, different wolverines are in zone z2 at 2pm at most with probability 0.35. These bounds can be inferred without accessing any portion of a potentially big data source at all, only requiring that the key $k1$ has at least marginal probability 0.65 on the given data set.

Contributions. The examples motivate us to stipulate lower bounds on the marginal probability of keys. The main inhibitor for the uptake of p-keys is the identification of the right lower bounds on their marginal probabilities. While it is already challenging to identify traditional keys which are semantically meaningful in a given application domain, identifying the right probabilities is an even harder problem. Lower bounds appear to be a realistic compromise here. Our contributions can be summarized as follows. **Modeling.** We propose p-keys $kX_{\geq p}$ as a natural class of semantic integrity constraints over uncertain data. Their main target is to help organizations balance consistency and completeness targets for the quality of their data. P-keys can distinguish semantically meaningful from meaningless patterns in large volumes of uncertain data from

CD table

rfid	*time*	*zone*	W
w1	2:00pm	z1	$1, 2, 3$
w1	3:00pm	z1	$1, 2, 4$
w1	3:00pm	z2	$1, 3, 4$
w2	3:00pm	z2	$2, 3, 4$

P table

W	$\mathcal{P}$
1	.2
2	.45
3	.3
4	.05

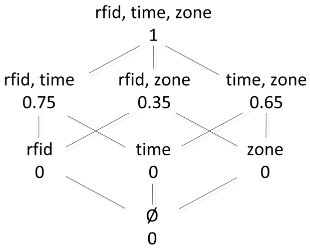


Fig. 1. Armstrong PC-table for $\{k1_{\geq 0.65}, k2_{\geq 0.75}, k3_{\geq 0.35}\}$ and its profile of p-keys

a variety of sources, and help quantify the probability for unique query answers. **Reasoning.** We characterize the implication problem of p-keys by a simple finite set of Horn rules, as well as a linear time decision algorithm. This enables organizations to reduce the overhead of data quality management by p-keys to a minimal level necessary. For example, enforcing $k\{\mathit{rfid}\}_{\geq 0.3}$, $k\{\mathit{rfid,time}\}_{\geq 0.25}$, and $k\{\mathit{rfid,zone}\}_{\geq 0.35}$, would be redundant as the enforcement of $k\{\mathit{rfid,time}\}_{\geq 0.25}$ is already implicitly done by enforcing $k\{\mathit{rfid}\}_{\geq 0.3}$. **Visualization.** For the schema-driven acquisition of the right marginal probabilities by which keys should hold, we show how to visualize concisely any given system of p-keys in the form of an Armstrong PC-table. An Armstrong PC-table is a perfect semantic summary of all p-keys currently perceived meaningful by the analysts. That is, the Armstrong PC-table satisfies every key with the exact marginal probability that is perceived to best represent the application domain. Any problems with such perceptions are explicitly pointed out by the PC-table. For example, the left of Figure 1 shows an Armstrong PC-table for $\{k1_{\geq 0.65}, k2_{\geq 0.75}, k3_{\geq 0.35}\}$. In the *CD* table, the W column of a tuple shows the identifiers of possible worlds to which the tuple belongs. The P-table shows the probability distribution on the possible worlds. Any p-key that is not implied by this set is violated, in particular the keys $k\{\mathit{rfid}\}$, $k\{\mathit{time}\}$ and $k\{\mathit{zone}\}$ all have marginal probability zero in the p-relation from Table 1, which is represented by this PC-table. **Profiling.** For the data-driven acquisition of p-keys we compute the marginal probability of every key from a given PC-table. This is also known as data profiling, and our paper is the first to propose probabilistic data profiling techniques. For example, if we want to know the marginal probabilities by which an attribute set forms a key in the PC-table from Figure 1, then our algorithm would return the profile $k\emptyset_{\geq 0}$, $k\{\mathit{rfid}\}_{\geq 0}$, $k\{\mathit{time}\}_{\geq 0}$, $k\{\mathit{zone}\}_{\geq 0}$, $k\{\mathit{rfid,time}\}_{\geq 0.75}$, $k\{\mathit{rfid,zone}\}_{\geq 0.35}$, $k\{\mathit{time,zone}\}_{\geq 0.65}$, and $k\{\mathit{rfid,time,zone}\}_{\geq 1}$, as visualized on the right of Figure 1. **Experiments.** Our experiments demonstrate that our visualization and profiling techniques work efficiently in the context of our acquisition framework.

Organization. We discuss related work in Section 2. P-keys are introduced in Section 3, and axiomatic and linear-time algorithmic characterizations of their implication problem are established in Section 4. These lay the foundation for the schema- and data-driven discovery algorithms of p-keys in Section 5. Experiments with these algorithms are presented in Section 6. We conclude and sketch future work in Section 7.

2 Related Work

Poor data quality is arguably the biggest inhibitor to deriving value from big data [31]. P-keys provide a principled tool to balance the consistency and completeness requirements of an organization on the quality of their data [23,30]. Primary impact areas of p-keys include data integration [5] where keys cannot be expected to hold with probability one; data modeling [27] where p-keys may represent target constraints that avoid data redundancy with certain degrees of probability; data processing [18] where p-keys facilitate updates and query answer exploration of targeted degrees of quality; compliance validation of business rules [25] where data is uncertain; in duplicate detection [3] where anomalous patterns of uncertain data are found; and in data cleaning and linkage [2]. The concept of probabilistic keys is new but naturally derived from previous research.

Our contributions extend results on keys from traditional relations, covered by our framework as the special case where the p-relation consists of one possible world only. Extensions include work on the classical implication problem [1,7,11,13–15], Armstrong relations [4,9,12,13,21,29] and the discovery of keys from relations [16,22,29]. In fact, our axiomatic and algorithmic characterizations of the implication problem as well as the schema- and data-driven discovery of the right probabilities of keys is novel. Specifically, Armstrong databases and data profiling have not been studied yet for probabilistic data. For certain relations there is empirical evidence that Armstrong databases help with the acquisition of meaningful business rules [4,20,21,29]. Our techniques will make it possible to conduct such empirical studies for p-keys in the future.

There is a large body of work on the discovery of "approximate" business rules, such as keys, functional and inclusion dependencies [10,17,24]. Approximate means here that not all tuples satisfy the given rule, but some exceptions are tolerable. Our constraints are not approximate since they are either satisfied or violated by the given p-relation or the PC-table that represents it. Again, it is future work to investigate approximate versions of probabilistic keys.

Closest to our approach is the work on possibilistic keys [19], where tuples are attributed some degree of possibility and keys some degree of certainty saying to which tuples they apply. In general, possibility theory is a qualitative approach, while probability theory is a quantitative approach to uncertainty. This research thereby complements the qualitative approach to keys in [19] by a quantitative approach.

Keys have also been included in description logic research [26,33], but we are unaware of any work concerning keys on probabilistic data.

3 Probabilistic Keys

We introduce some preliminary concepts from probabilistic databases and the central notion of a probabilistic key.

A *relation schema* is a finite set R of attributes A. Each attribute A is associated with a domain $dom(A)$ of values. A tuple t over R is a function that assigns to each attribute A of R an element $t(A)$ from the domain $dom(A)$. A *relation* over R is a finite set of tuples over R. Relations over R are also called *possible worlds* of R here. An expression kX over R with $X \subseteq R$ is called a *key*. A key kX is said to hold in a possible world W of R, denoted by $W \models kX$, if and only if there no two tuples $t_1, t_2 \in W$ such that $t_1 \neq t_2$ and $t_1(X) = t_2(X)$. A *probabilistic relation* (p-relation) over R is a pair $r = (\mathcal{W}, P)$ of a finite non-empty set $\mathcal{W}$ of possible worlds over R and a probability distribution $P : \mathcal{W} \to (0, 1]$ such that $\sum_{W \in \mathcal{W}} P(W) = 1$ holds. Table 1 shows a probabilistic relation over relation schema Wolverine=$\{$*rfid*,*time*,*zone*$\}$. World W_2, for example, satisfies the keys $k\{$*rfid*, *time*$\}$ and $k\{$*zone*, *time*$\}$, but violates the key $k\{$*rfid*, *zone*$\}$. The *marginal probability* of a key kX in the p-relation $r = (\mathcal{W}, P)$ over relation schema R is the sum of the probabilities of those possible worlds in r which satisfy the key. We will now introduce the central notion of a probabilistic key.

Definition 1. *A* probabilistic key, *or* p-key *for short, over relation schema R is an expression $kX_{\geq p}$ where $X \subseteq R$ and $p \in [0, 1]$. The p-key $kX_{\geq p}$ over R is* satisfied by, *or said to* hold in, *the p-relation r over R if and only if the marginal probability of kX in r is not smaller than p.*

In our running example over relation schema Wolverine, the p-relation from Table 1 satisfies the p-keys $k\{$*rfid*, *time*$\}_{\geq 0.75}$ and $k\{$*rfid*, *zone*$\}_{\geq 0.35}$, but violates the p-keys $k\{$*rfid*, *time*$\}_{\geq 0.9}$ and $k\{$*rfid*, *zone*$\}_{\geq 0.351}$.

4 Reasoning Tools

When using sets of p-keys to manage the consistency and completeness targets on the quality of an organization's data, it is important that their overhead is reduced to a minimal level necessary. In practice, this requires us to reason about p-keys efficiently. It is the goal of this section to establish basic tools to reason about the interaction of p-keys. This will help us identify efficiently the largest probability by which a given key is implied from a given set of p-keys, and to optimize the efficiency of updates and query answers, for example. The results will also help us develop our acquisition framework later.

Let $\Sigma \cup \{\varphi\}$ denote a set of constraints over relation schema R. We say Σ *implies* φ, denoted by $\Sigma \models \varphi$, if every p-relation r over R that satisfies Σ, also satisfies φ. We use $\Sigma^* = \{\varphi : \Sigma \models \varphi\}$ to denote the *semantic closure* of Σ. For a class $\mathcal{C}$ of constraints, the $\mathcal{C}$-implication problem is to decide for a given relation schema R and a given set $\Sigma \cup \{\varphi\}$ of constraints in $\mathcal{C}$ over R, whether Σ implies φ. We will now characterize the $\mathcal{C}$-implication problem for the class of p-keys axiomatically by a simple finite set of Horn rules, and algorithmically by a linear time algorithm.

Axioms. We determine the semantic closure by applying *inference rules* of the form $\frac{\text{premise}}{\text{conclusion}}$. For a set $\mathfrak{R}$ of inference rules let $\Sigma \vdash_{\mathfrak{R}} \varphi$ denote the

Table 2. Axiomatization $\mathfrak{P} = \{\mathcal{T}, \mathcal{Z}, \mathcal{S}, \mathcal{W}\}$

$\dfrac{}{kR_{\geq 1}}$	$\dfrac{}{kX_{\geq 0}}$	$\dfrac{kX_{\geq p}}{kXY_{\geq p}}$	$\dfrac{kX_{\geq p+q}}{kX_{\geq p}}$
(Trivial, $\mathcal{T}$)	(Zero, $\mathcal{Z}$)	(Superkey, $\mathcal{S}$)	(Weakening, $\mathcal{W}$)

inference of φ from Σ by $\mathfrak{R}$. That is, there is some sequence $\sigma_1, \ldots, \sigma_n$ such that $\sigma_n = \varphi$ and every σ_i is an element of Σ or is the conclusion that results from an application of an inference rule in $\mathfrak{R}$ to some premises in $\{\sigma_1, \ldots, \sigma_{i-1}\}$. Let $\Sigma^+_{\mathfrak{R}} = \{\varphi : \Sigma \vdash_{\mathfrak{R}} \varphi\}$ be the *syntactic closure* of Σ under inferences by $\mathfrak{R}$. $\mathfrak{R}$ is *sound* (*complete*) if for every set Σ over every R we have $\Sigma^+_{\mathfrak{R}} \subseteq \Sigma^*$ ($\Sigma^* \subseteq \Sigma^+_{\mathfrak{R}}$). The (finite) set $\mathfrak{R}$ is a (finite) *axiomatization* if $\mathfrak{R}$ is both sound and complete. The set $\mathfrak{P}$ of inference rules from Table 2 forms a finite axiomatization for the implication of p-keys. Here, R denotes the underlying relation schema, X and Y form attribute subsets of R, and p, q as well as $p + q$ are probabilities.

Theorem 1. *$\mathfrak{P}$ forms a finite axiomatization for p-keys.* □

For example, the set $\Sigma = \{k\{\mathit{time}\}_{\geq 0.2}, k\{\mathit{rfid}\}_{\geq 0.3}\}$ imply the p-key $\varphi = k\{\mathit{rfid}, \mathit{time}\}_{\geq 0.25}$, but not the p-key $\varphi' = k\{\mathit{rfid}, \mathit{time}\}_{\geq 0.35}$. Indeed, φ can be inferred from Σ by applying $\mathcal{S}$ to $k\{\mathit{rfid}\}_{\geq 0.3}$ to infer $k\{\mathit{rfid}, \mathit{time}\}_{\geq 0.3}$, and applying $\mathcal{W}$ to $k\{\mathit{rfid}, \mathit{time}\}_{\geq 0.3}$ to infer φ. If a data set is valid for the set Σ of p-keys, it is also valid for every p-key φ implied by Σ. The larger the data set, the more time we save by avoiding redundant validation checks.

Algorithms. In practice, the semantic closure Σ^* of a finite set Σ is infinite and even though it can always be represented finitely, it is often unnecessary to determine all implied p-keys. In fact, the implication problem for p-keys has as input $\Sigma \cup \{\varphi\}$ and the question is whether Σ implies φ. Computing Σ^* and checking whether $\varphi \in \Sigma^*$ is not feasible. In fact, we will now establish a linear-time algorithm for computing the maximum probability p, such that $kX_{\geq p}$ is implied by Σ. The following theorem allows us to reduce the implication problem for p-keys to a single scan of the input.

Theorem 2. *Let $\Sigma \cup \{kX_{\geq p}\}$ denote a set of p-keys over relation schema R. Then Σ implies $kX_{\geq p}$ if and only if $X = R$ or $p = 0$ or there is some $kZ_{\geq q} \in \Sigma$ such that $Z \subseteq X$ and $q \geq p$.* □

Theorem 2 enables us to design Algorithm 1, which returns the maximum probability p by which a given key kX is implied by a given set Σ of p-keys over R. If $X = R$, then we return probability 1. Otherwise, starting with $p = 0$ the algorithm scans all input keys $kZ_{\geq q}$ and sets p to q whenever q is larger than the current p and X contains Z. We use $|\Sigma|$ and R to denote the total number of attributes that occur in Σ and R, respectively.

Theorem 3. *On input (R, Σ, kX), Algorithm 1 returns in $\mathcal{O}(|\Sigma| + |R|)$ time the maximum probability p with which $kX_{\geq p}$ is implied by Σ.* □

Algorithm 1. Inference

```
Require: $R, \Sigma, kX$
Ensure: $\max\{p : \Sigma \models kX_{\geq p}\}$
if $X = R$ then
    $p \leftarrow 1$;
else
    $p \leftarrow 0$;
    for all $kZ_{\geq q} \in \Sigma$ do
        if $Z \subseteq X$ and $q > p$ then
            $p \leftarrow q$;
return $p$;
```

Given $R, \Sigma, kX_{\geq p}$ as an input to the implication problem we can use Algorithm 1 to compute $p' := \max\{q : \Sigma \models kX_{\geq q}\}$ and return an affirmative answer if and only if $p' \geq p$.

Corollary 1. *The implication problem of p-keys is decidable in linear time.* □

Given the p-key set $\Sigma = \{k\{\mathit{time}\}_{\geq 0.2}, k\{\mathit{rfid}\}_{\geq 0.3}\}$ and the key $k\{\mathit{rfid}, \mathit{time}\}$, Algorithm 1 returns $p = 0.3$. Consequently, the p-key $k\{\mathit{rfid}, \mathit{time}\}_{\geq 0.25}$ is implied by Σ, but $k\{\mathit{rfid}, \mathit{time}\}_{\geq 0.35}$ is not implied by Σ.

5 Tools for Acquiring Probabilistic Keys

Applications will benefit from the ability of analysts to acquire a good lower bound for the marginal probability by which keys hold in the domain of the application. For that purpose, analysts should communicate with domain experts. We establish two major tools that help analysts to communicate effectively with domain experts. We follow the framework in Figure 2. Here, analysts use our algorithm to visualize abstract sets Σ of p-keys in the form of some Armstrong PC-table, which is then inspected jointly with domain experts. In particular, the PC-table represents simultaneously for every key kX the marginal probability that quality data sets in the target domain should exhibit. Domain experts may change the PC-table or supply new PC-tables to the analysts. For that case we establish an algorithm that profiles p-keys. That is, the algorithm computes the marginal probability of each key in the given PC-table. Such profiles are also useful for query optimization, for example.

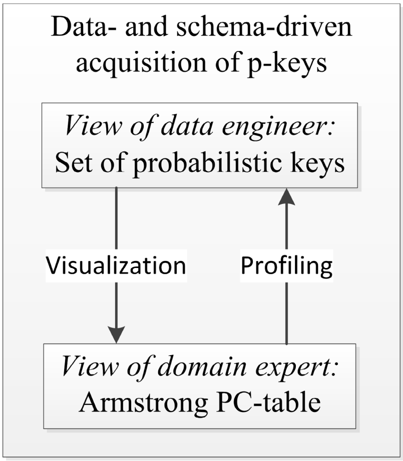


Fig. 2. Acquisition framework

5.1 Visualizing Abstract Sets of p-keys as Armstrong PC-tables

Our results will show that every abstract set of p-keys can be visualized in the form of a single PC-table that represents a p-relation that satisfies all given p-keys and violates all those p-keys not implied by the given set. This notion is known as an *Armstrong database*, which we formally recall here [8]. Let Σ denote a set of p-keys over a given relation schema R. A p-relation $r = (\mathcal{W}, P)$ over R is *Armstrong* for Σ if and only if for all p-keys φ it holds that r satisfies φ if and only if Σ implies φ. The following theorem shows that every distribution of probabilities to keys, that follows the inference rules from Table 2, can be represented by a single p-relation which exhibits this distribution in the form of marginal probabilities.

Theorem 4. *Let $l : R \rightarrow [0, 1]$ be a function such that $l(R) = 1$ and for all $X, Y \subseteq R$, $l(XY) \geq l(X)$ holds. Then there is some p-relation r over R such that r satisfies $kX_{\geq l(X)}$, and for all $X \subseteq R$ and for all $p \in [0, 1]$ such that $p > l(X)$, r violates $kX_{\geq p}$.*

Proof. Let $\{l_1, \ldots, l_n\} = \{l(X) : X \subseteq R\}$ such that $l_1 < l_2 < \ldots < l_n$, and let $l_0 = 0$. Define a probabilistic relation $r = (\{W_1, \ldots, W_n\}, P)$ as follows. For all $i = 1, \ldots, n$, the world W_i is an Armstrong relation for the key set $\Sigma_i = \{kY : l(Y) \geq l_i\}$, and $P(W_i) = l_i - l_{i-1}$. For all $X \subseteq R$, let $l(X) = l_j$ for $j \in \{1, \ldots, n\}$. Then, kX holds on W_i if and only if $i \leq j$. Consequently, kX has marginal probability $l(X)$ with respect to r, and $kX_{\geq l(X)}$ is satisfied. However, r violates $kX_{\geq p}$ for every $p > l(X)$. □

Let Σ be a set of p-keys. For all $X \subseteq R$, let $p_X := \sup\{p : \exists Y \subseteq X(kY_{\geq p} \in \Sigma \cup \{kR_{\geq 1}\})\}$. Then for all $Z \subseteq R$, Σ implies $kZ_{\geq p}$ if and only if $p \leq p_Z$. Now, let $l(X) := p_X$. Then $l(R) = p_R = 1$ and $l(XY) = p_{XY} \geq p_X = l(X)$. By Theorem 4 it follows that there is some Armstrong p-relation r, since for all $Z \subseteq R$ and all $p \in [0, 1]$, Σ implies $kZ_{\geq p}$ if and only if r satisfies $kZ_{\geq p}$.

Instead of computing Armstrong p-relations we compute PC-tables that are concise representations of Armstrong p-relations. We call these *Armstrong* PC-tables. Recall the following standard definition from probabilistic databases [32]. A *conditional table* or *c-table*, is a tuple $CD = \langle r, W \rangle$, where r is a relation, and W assigns to each tuple t in r a finite set W_t of positive integers. The set of *world identifiers* of CD is the union of the sets W_t for all tuples t of r. Given a world identifier i of CD, the possible world associated with i is $W_i = \{t | t \in r \text{ and } i \in W_t\}$. The semantics of a c-table $CD = \langle r, W \rangle$, called *representation*, is the set $\mathcal{W}$ of possible worlds W_i where i denotes some world identifier of CD. A *probabilistic conditional database* or *PC-table*, is a pair $\langle CD, P \rangle$ where CD is a c-table, and P is a probability distribution over the set of world identifiers of CD. The set of possible worlds of a PC-table $\langle CD, P \rangle$ is the representation of CD, and the probability of each possible world W_i is defined as the probability of its world identifier. For example, Figure 1 shows a PC-table $\langle CD, P \rangle$ that is Armstrong for the p-relation in Table 1.

We will now describe an algorithm that computes an Armstrong PC-table for every given set Σ of p-keys. In our construction, the number of possible worlds

Algorithm 2. Armstrong PC-table

```
**Require:** $R, \Sigma$
**Ensure:** Armstrong PC-table $\langle CD, P \rangle$ for $\Sigma$
 1: Let $p_1, \ldots, p_n$ denote the $i$-th smallest probabilities $p_i$ occurring in $\Sigma$; ▷ If $p_n < 1$,
    $n \leftarrow n + 1$ and $p_n \leftarrow 1$
 2: $p_0 \leftarrow 0$;
 3: $P \leftarrow \emptyset$;
 4: for $i = 1, \ldots, n$ do
 5:     $P \leftarrow P \cup \{(i, p_i - p_{i-1})\}$;        ▷ World $i$ has probability $p_i - p_{i-1}$
 6:     $A_i^{-1} \leftarrow$ Set of anti-keys for $\Sigma_{p_i}$;        ▷ Anti-keys to be realized in world $i$
 7: $A^{-1} \leftarrow \emptyset$;
 8: for all $X \in A_1^{-1} \cup \cdots \cup A_n^{-1}$ do
 9:     $A^{-1} \leftarrow A^{-1} \cup \{(X, \{i : X \in A_i^{-1}\})\}$;        ▷ Worlds in which $X$ is an anti-key
10: for all $A \in R$ do
11:     $t_0(A) \leftarrow c_{A,0}$;
12: $CD \leftarrow \{(t_0, \{1, \ldots, n\})\}$;        ▷ Tuple $t_0$ is part of every world
13: $j \leftarrow 0$;
14: for all $(X, W) \in A^{-1}$ do  ▷ For each $X$ that is an anti-key in every world in $W$...
15:     $j \leftarrow j + 1$;
16:     for all $A \in R$ do▷ Add some $t_j$ that realizes agree set $X$ in every world in $W$
17:         $t_j(A) \leftarrow \begin{cases} c_{A,0} & \text{, if } A \in X \\ c_{A,j} & \text{, otherwise} \end{cases}$;
18:     $CD \leftarrow CD \cup \{(t_j, W)\}$;
19: return $\langle CD, P \rangle$;
```

is determined by the number of distinct probabilities that occur in Σ. For that purpose, for every given set Σ of p-keys over R and every probability $p \in [0, 1]$, let $\Sigma_p = \{kX : \exists kX_{\geq q} \in \Sigma \wedge q \geq p\}$ denote the *p-cut* of Σ, i.e., the set of keys over R which have at least marginal probability p. It is possible that Σ does not contain any p-key $kX_{\geq p}$ where $p = 1$. In this case, Algorithm 2 computes an Armstrong PC-table for Σ that contains one more possible world than the number of distinct probabilities occurring in Σ. Processing the probabilities Σ from smallest p_1 to largest p_n, the algorithm computes as possible world with probability $p_i - p_{i-1}$ (line 5) a traditional Armstrong relation for the p_i-cut Σ_{p_i}. For this purpose, the anti-keys are computed for each p_i-cut (line 6), and the set W of those worlds i is recorded for which X is an anti-key with respect to Σ_{p_i} (line 9). The CD-table contains one tuple t_0 which occurs in all possible worlds (line 12), and for each anti-key X another tuple t_j that occurs in all worlds for which X is an anti-key and that has matching values with t_0 in exactly the columns of X (lines 14-18).

Theorem 5. *For every set Σ of p-keys over relation schema R, Algorithm 2 computes an Armstrong PC-table for Σ in which the number of possible worlds coincides with the number of distinct probabilities that occur in $\Sigma \cup \{kR_{\geq 1}\}$.* □

In our running example, Σ contains $k\{\mathit{rfid}, \mathit{time}\}_{\geq 0.75}$, $k\{\mathit{time}, \mathit{zone}\}_{\geq 0.65}$, and $k\{\mathit{rfid}, \mathit{zone}\}_{\geq 0.35}$. Applying Algorithm 2 to WOLVERINE and Σ may result in the Armstrong PC-table of Figure 3. Finally, we derive some bounds on the time complexity of finding Armstrong PC-tables. Additional insight is given by our experiments in Section 6.

Fig. 3. An Armstrong PC-table

CD table

rfid	*time*	*zone*	W
w1	2pm	z1	$1, 2, 3, 4$
w1	3pm	z2	1
w2	4pm	z1	1
w3	2pm	z3	$1, 2$
w1	5pm	z1	$2, 3, 4$
w4	2pm	z1	$3, 4$
w1	2pm	z4	4

P table

W	$\mathcal{P}$
1	.35
2	.3
3	.1
4	.25

Theorem 6. *The time complexity to find an Armstrong PC-table for a given set Σ of p-keys over relation schema R is precisely exponential in $|\Sigma|$.*

Proof. Given R and Σ as input, Algorithm 2 computes an Armstrong PC-table for Σ in time at most exponential in $|\Sigma|$. Indeed, an Armstrong relation for Σ_{p_i} can be computed in time at most exponential in $|\Sigma_{p_i}| \leq |\Sigma|$, and we require no more than $|\Sigma|$ computations of such relations.

There are cases where the number of tuples in any Armstrong PC-table for Σ over R is exponential in $|\Sigma|$. Such a case is given by $R_n = \{A_1, \ldots, A_{2n}\}$ and $\Sigma_n = \{\{A_1, A_2\}_{\geq 1}, \ldots, \{A_{2n-1}, A_{2n}\}_{\geq 1}\}$ with $|\Sigma_n| = 2 \cdot n$. Every Armstrong PC-table requires $2^n + 1$ tuples, and there is only one possible world. □

There are also cases where the number of tuples in some Armstrong PC-table for Σ over R is logarithmic in $|\Sigma|$. Such a case is given by $R_n = \{A_1, \ldots, A_{2n}\}$ and $\Sigma_n = \{(X_1 \cdots X_n)_{\geq 1} : X_i \in \{A_{2i-1}, A_{2i}\} \text{ for } i = 1, \ldots, n\}$ with $|\Sigma_n| = n \cdot 2^n$. One Armstrong PC-table for Σ represents a single possible world which has $n + 1$ tuples that realize the n agree sets $R - \{A_{2i-1}, A_{2i}\}$, the sets of attributes on which some pair of distinct tuples have matching values.

5.2 Profiling of p-keys from PC-tables

The profiling problem of p-keys from a given PC-table $\langle CD, P\rangle$ over a relation schema R is to determine for all $X \subset R$, the marginal probability p_X of kX in the p-relation $r = (\mathcal{W}, P)$ that $\langle CD, P\rangle$ represents. The problem can be solved as follows: for each $X \subset R$, initialize $p_X \leftarrow 0$ and for all worlds $W \in \mathcal{W}$, add the probability p_W of W to p_X, if X contains some minimal key of W, see Algorithm 3. The set of minimal keys of a world W is given by the set of minimal transversals over the disagree sets of W (the complements of agree sets) [28]. Applying Algorithm 3 to the PC-table from Figure 1 returns the p-keys $k\{\mathit{time}, \mathit{zone}\}_{\geq 0.65}$, $k\{\mathit{rfid}, \mathit{time}\}_{\geq 0.75}$, $k\{\mathit{rfid}, \mathit{zone}\}_{\geq 0.35}$ and $kX_{\geq 0}$ for all remaining $X \subset R$, as illustrated on the right of Figure 1.

Algorithm 3. Profiling

```
Require: PC-table $\langle CD, P\rangle$ over relation schema $R$
Ensure: For all $X \subset R$, the maximum $p_X$ such that $kX_{\geq p_X}$ holds on p-relation
    $r = (\mathcal{W}, P)$ that $\langle CD, P\rangle$ represents
for all $X \subset R$ do
    $p_X \leftarrow 0$;
for all $W \in \mathcal{W}$ do
    $\mathcal{M}(W) \leftarrow$ Set of minimal keys on $W$;        ▷ by known algorithm, e.g., [28]
    for all $X \subset R$ do
        if $X$ contains some $M \in \mathcal{M}(W)$ then
            $p_X \leftarrow p_X + P(W)$;
return $\{(X, p_X) : X \subseteq R\}$;
```

6 Experiments

In this section we report on some experiments regarding the computational complexity of our algorithms for the visualization and discovery of probabilistic keys.

6.1 Visualization

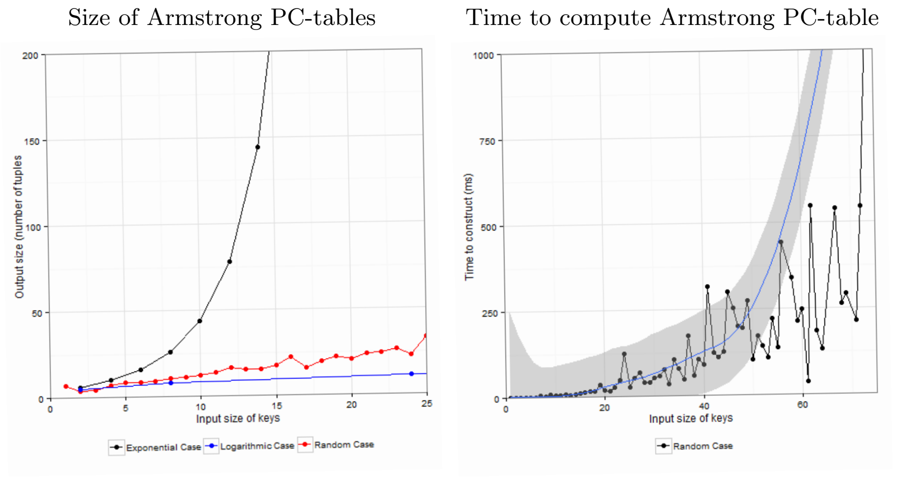
The Armstrong construction takes as input a set Σ of randomly generated p-keys, and outputs an Armstrong PC-table for Σ. The random generation of Σ was achieved by firstly sampling n probabilities p_n from $[0, 1]$ and for each attribute set $X \subset R$, we assign a probability randomly sampled from the set $\{0\} \cup \{p_1, p_2, \ldots, p_n\}$. For our experiments, n was at most 15.

The left of Figure 4 shows the number of tuples in the Armstrong PC-table as a function of applying Algorithm 2 to the exponential case from the proof of Theorem 6 (black line), the logarithmic case described after Theorem 6 (blue line), and the random generation (red line). The figure illustrates that the average size of an Armstrong PC-table grows linearly in the input key size. The worst-case exponential growth occurs rarely on average. This demonstrates that Armstrong PC-tables exhibit small sizes on average, which makes them a practical tool to acquire meaningful p-keys in a joint effort with domain experts.

The right of Figure 4 shows the time for computing Armstrong PC-tables from the given sets of randomly created p-keys. It shows that Armstrong PC-tables can be computed efficiently for the input sizes considered. In fact, their computation hardly ever exceeded 1 second. The left of Figure 6 shows the graphical user interface of our visualization tool, developed in R. The input interface is shown on the left, and the output PC-table on the right.

6.2 Profiling

Figure 6 shows the time for profiling p-keys from the given Armstrong PC-tables we randomly created previously. It illustrates that the profiling problem can be solved efficiently for input sizes typical for our acquisition framework, see Figure 2. Large input sizes will require more sophisticated techniques.


Size of Armstrong PC-tables

Time to compute Armstrong PC-table

Fig. 4. Results of experiments with visualization

7 Conclusion and Future Work

We have introduced probabilistic keys that stipulate lower bounds on the marginal probability by which keys shall hold on large volumes of uncertain data. The marginal probability of keys provides a principled mechanism to control the consistency and completeness targets for the quality of an organization's data, as illustrated in Figure 5.

We have established axiomatic and algorithmic tools to reason about probabilistic keys. This can minimize the overhead in using them for data quality management and query processing. These applications are effectively unlocked by developing support for identifying the right marginal probabilities by which keys should hold in a given application domain. For this challenging problem, we have developed schema- and data-driven algorithms that can be used by analysts to communicate more effectively with domain experts. The schema-driven algorithm converts any input in the form of an abstract set of probabilistic keys into an Armstrong PC-table that satisfies the input and violates all probabilistic keys not implied by the input. Analysts and domain experts can jointly inspect the Armstrong PC-table which points out any flaws in the current perception of marginal probabilities. The data-driven algorithm computes a profile of the probabilistic keys that a given PC-table satisfies.

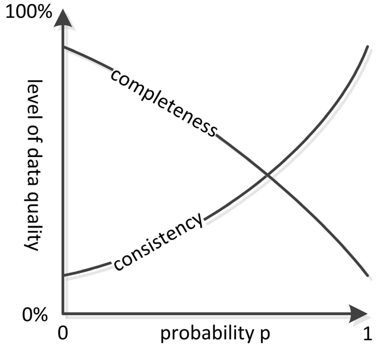
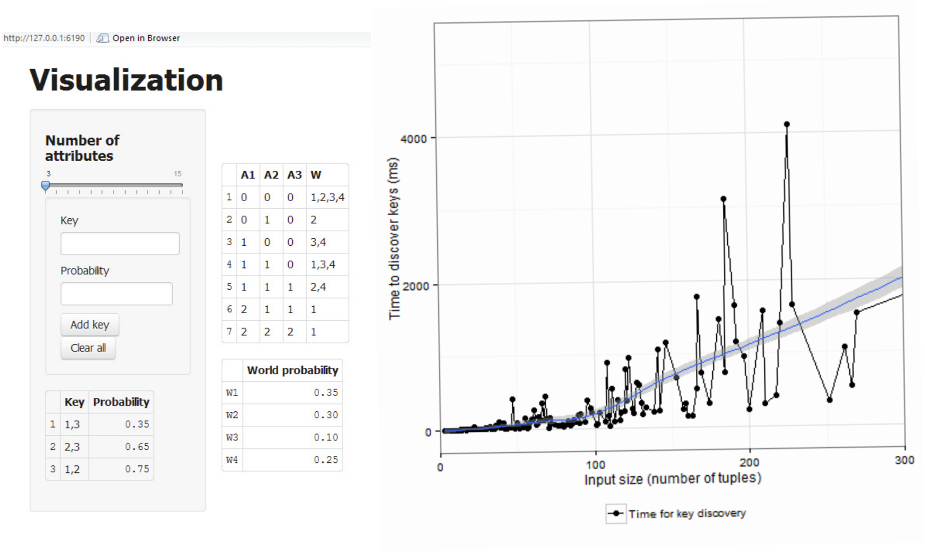


Fig. 5. Control mechanism p



	Key	Probability
1	1,3	0.35
2	2,3	0.65
3	1,2	0.75

	A1	A2	A3	W
1	0	0	0	1,2,3,4
2	0	1	0	2
3	1	0	0	3,4
4	1	1	0	1,3,4
5	1	1	1	2,4
6	2	1	1	1
7	2	2	2	1

	World probability
W1	0.35
W2	0.30
W3	0.10
W4	0.25

Fig. 6. GUI for visualization and times for profiling p-keys

Such PC-tables may represent some exemplary data sets or result from changes to a given Armstrong PC-table in response to identifying some flaws during their inspection. Experiments confirm that the computation of Armstrong PC-tables is typically efficient, their size is small, and profiles of probabilistic keys can be efficiently computed from PC-tables of reasonable size.

In future research we will apply our algorithms to investigate empirically the usefulness of our framework for acquiring the right marginal probabilities of keys in a given application domain. This will require us to extend empirical measures from certain [20–22] to probabilistic data sets. Particularly intriguing is the question whether PC-tables or p-relations are more useful. We will also investigate the scalability of the profiling problem to large data sets, by applying the MapReduce framework to recent data profiling techniques [16]. It is also interesting to raise the expressivity of probabilistic keys by allowing the stipulation of upper bounds or other features.

Acknowledgments. This research is supported by the Marsden fund council from Government funding, administered by the Royal Society of New Zealand.

References

1. Armstrong, W.W.: Dependency structures of data base relationships. In: IFIP Congress. pp. 580–583 (1974)
2. Atencia, M., David, J., Scharffe, F.: Keys and pseudo-keys detection for web datasets cleansing and interlinking. In: ten Teije, A., Völker, J., Handschuh, S., Stuckenschmidt, H., d'Acquin, M., Nikolov, A., Aussenac-Gilles, N., Hernandez, N. (eds.) EKAW 2012. LNCS, vol. 7603, pp. 144–153. Springer, Heidelberg (2012)

3. de Bakker, M., Frasincar, F., Vandic, D.: A hybrid model words-driven approach for web product duplicate detection. In: Salinesi, C., Norrie, M.C., Pastor, Ó. (eds.) CAiSE 2013. LNCS, vol. 7908, pp. 149–161. Springer, Heidelberg (2013)
4. Beeri, C., Dowd, M., Fagin, R., Statman, R.: On the structure of Armstrong relations for functional dependencies. J. ACM **31**(1), 30–46 (1984)
5. Blanco, L., Crescenzi, V., Merialdo, P., Papotti, P.: Probabilistic models to reconcile complex data from inaccurate data sources. In: Pernici, B. (ed.) CAiSE 2010. LNCS, vol. 6051, pp. 83–97. Springer, Heidelberg (2010)
6. Codd, E.F.: A relational model of data for large shared data banks. Commun. ACM **13**(6), 377–387 (1970)
7. Diederich, J., Milton, J.: New methods and fast algorithms for database normalization. ACM Trans. Database Syst. **13**(3), 339–365 (1988)
8. Fagin, R.: Horn clauses and database dependencies. J. ACM **29**(4), 952–985 (1982)
9. Geiger, D., Pearl, J.: Logical and algorithmic properties of conditional independence and graphical models. The Annals of Statistics **21**(4), 2001–2021 (1993)
10. Giannella, C., Robertson, E.L.: On approximation measures for functional dependencies. Inf. Syst. **29**(6), 483–507 (2004)
11. Hannula, M., Kontinen, J., Link, S.: On independence atoms and keys. In: Li, J., Wang, X.S., Garofalakis, M.N., Soboroff, I., Suel, T., Wang, M. (eds.) Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, November 3–7, 2014, pp. 1229–1238 (2014)
12. Hartmann, S., Kirchberg, M., Link, S.: Design by example for SQL table definitions with functional dependencies. VLDB J. **21**(1), 121–144 (2012)
13. Hartmann, S., Leck, U., Link, S.: On Codd families of keys over incomplete relations. Comput. J. **54**(7), 1166–1180 (2011)
14. Hartmann, S., Link, S.: Efficient reasoning about a robust XML key fragment. ACM Trans. Database Syst. **34**(2) (2009)
15. Hartmann, S., Link, S.: The implication problem of data dependencies over SQL table definitions. ACM Trans. Database Syst. **37**(2), 13 (2012)
16. Heise, A., Jorge-Arnulfo, Q.-R., Abedjan, Z., Jentzsch, A., Naumann, F.: Scalable discovery of unique column combinations. PVLDB **7**(4), 301–312 (2013)
17. Huhtala, Y., Kärkkäinen, J., Porkka, P., Toivonen, H.: TANE: an efficient algorithm for discovering functional and approximate dependencies. Comput. J. **42**(2), 100–111 (1999)
18. Jha, A.K., Rastogi, V., Suciu, D.: Query evaluation with soft-key constraints. In: PODS. pp. 119–128 (2008)
19. Koehler, H., Leck, U., Link, S., Prade, H.: Logical foundations of possibilistic keys. In: Fermé, E., Leite, J. (eds.) JELIA 2014. LNCS, vol. 8761, pp. 181–195. Springer, Heidelberg (2014)
20. Langeveldt, W., Link, S.: Empirical evidence for the usefulness of armstrong relations in the acquisition of meaningful functional dependencies. Inf. Syst. **35**(3), 352–374 (2010)
21. Le, V.B.T., Link, S., Ferrarotti, F.: Effective recognition and visualization of semantic requirements by perfect SQL samples. In: Ng, W., Storey, V.C., Trujillo, J.C. (eds.) ER 2013. LNCS, vol. 8217, pp. 227–240. Springer, Heidelberg (2013)
22. Le, V.B.T., Link, S., Memari, M.: Schema- and data-driven discovery of SQL keys. JCSE **6**(3), 193–206 (2012)
23. Link, S.: Consistency enforcement in databases. In: Bertossi, L.E., Katona, G.O.H., Schewe, K., Thalheim, B. (eds.) Semantics in Databases. LNCS 2582, vol. 2582, pp. 139–159. Springer, Heidelberg (2003)

24. Liu, J., Li, J., Liu, C., Chen, Y.: Discover dependencies from data - A review. IEEE Trans. Knowl. Data Eng. **24**(2), 251–264 (2012)
25. López, M.T.G., Gasca, R.M., Pérez-Álvarez, J.M.: Compliance validation and diagnosis of business data constraints in business processes at runtime. Inf. Syst. **48**, 26–43 (2015)
26. Lutz, C., Areces, C., Horrocks, I., Sattler, U.: Keys, nominals, and concrete domains. J. Artif. Intell. Res. (JAIR) **23**, 667–726 (2005)
27. Malhotra, K., Medhekar, S., Navathe, S.B., Laborde, M.D.D.: Towards a form based dynamic database schema creation and modification system. In: Jarke, M., Mylopoulos, J., Quix, C., Rolland, C., Manolopoulos, Y., Mouratidis, H., Horkoff, J. (eds.) CAiSE 2014. LNCS, vol. 8484, pp. 595–609. Springer, Heidelberg (2014)
28. Mannila, H., Räihä, K.J.: Algorithms for inferring functional dependencies from relations. Data Knowl. Eng. **12**(1), 83–99 (1994)
29. Ramdoyal, R., Hainaut, J.-L.: Interactively eliciting database constraints and dependencies. In: Mouratidis, H., Rolland, C. (eds.) CAiSE 2011. LNCS, vol. 6741, pp. 184–198. Springer, Heidelberg (2011)
30. Sadiq, S.: Handbook of Data Quality. Springer (2013)
31. Saha, B., Srivastava, D.: Data quality: The other face of big data. In: ICDE. pp. 1294–1297 (2014)
32. Suciu, D., Olteanu, D., Ré, C., Koch, C.: Probabilistic Databases. Synthesis Lectures on Data Management, Morgan & Claypool Publishers (2011)
33. Toman, D., Weddell, G.E.: On keys and functional dependencies as first-class citizens in description logics. J. Autom. Reasoning **40**(2–3), 117–132 (2008)