Ontology Functional Dependencies

Alexander Keller, and Jaroslaw Szlichta

University of Ontario Institute of Technology, Oshawa, Canada
{Alexander.Keller, Jaroslaw.Szlichta}@uoit.ca

Abstract. We extend traditional functional dependencies (FDs) for data quality purposes to accommodate ontological variations in the attribute values. We begin by formally defining a novel class of dependencies called ontological FDs, which strictly generalize traditional FDs by allowing differences controlled by an ontology database. The ontology databases contain information about synonyms. We then focus on designing the efficient algorithm for data verification over ontology FDs as well as discuss current and future work.

1 Introduction

Poor data quality is a bottleneck in data analytics to make effective business decisions. With the interest in data analytics at an all-time high, data quality has become a critical issue in research and practice. Integrity constraints are commonly used to characterize and ensure data quality [1]. In particular, functional dependencies (FDs) traditionally used in schema design, have been utilized for data quality purposes. A traditional FD states that if two tuples agree on the antecedent attributes, then they also must agree on the consequent attributes.

When integrating data from various sources, it is often that small variations occur which cause traditional FDs to be violated. We introduce ontology FDs to replace strict equality with a notion of similarity controlled by an ontology database. To illustrate the utility of ontology FDs, consider the medical trials dataset shown in Table 1, which was merged from various hospitals (and countries). In a table such as this, we would expect the FD {Country} $\rightarrow$ {Country Code} to hold. However, different hospitals may use synonyms for country codes, e.g., *CA* and *CAD* for the country *Canada*.

As another example, consider an FD {Disease, Country} $\rightarrow$ {Medicine}. In this case, within the country *Canada* and disease *Common cold*, the prescribed medicine *Benz* and *Benzonatate* are synonyms. Similarly within the country *United States* and disease *Common cold* patients are prescribed *Advil* and *Ibuprofen* that are also synonyms. In such settings, where traditional FDs seem to be overly restrictive, we introduce synonym ontology FDs. Thus, ontology FDs strictly generalize FDs and can express the additional semantics of ontological variations.

The remainder of this paper is organized as follows. In Section 2.1, we introduce formally a novel class of data dependencies called ontology FDs, which describe integrity constraints on tuples with ontologically similar attribute values which are useful in data quality. Next, we present efficient algorithms for

Patient ID	Country	Country Code	Disease	Medicine
1	Canada	CA	Common cold	Benzonatate
2	Canada	CAD	Common cold	Benz
3	Canada	CAD	Common cold	Benz
4	United States	USA	Common cold	Advil
5	United States	US	Common cold	Ibuprofen
6	United States	USA	Common cold	Ibuprofen

Table 1: Medical Trials Dataset

data verification over ontology FDs in Section 2.2. We implemented the data verification algorithms and experimentally verified their efficiency in practice over a medical trails dataset with 500K tuples (Section 2.2). We conclude the paper and discuss current and future work in Section 3.

2 Verification Problem

2.1 Problem Statement

To accommodate ontological variations in attribute values, we define *ontology FDs*. This is a departure from traditional FDs, which enforce equality on both sides of a dependency. Before we define ontology FDs, we first define notational conventions. Let $\mathbf{R}$ be a relation on which a set of dependencies is defined and let $\mathbf{r}$ be a relation instance (table) of $\mathbf{R}$. Capital letters near the beginning of the alphabet represent single attributes, e.g., A and B. Calligraphic letters near the end of the alphabet stand for sets of attributes, e.g., $\mathcal{X}$ and $\mathcal{Y}$. Tuples are marked with small letters in italics: t and s, where t[A] denotes the value of an attribute A in tuple t.

We assume ontology databases contain a set of *classes* (concepts). We assume the relation contains, as attribute values, string representations of classes called string terms. Terms are string representations of classes in the ontology defined with predicate $synonyms(C)$. A class with multiple instances, i.e., $|synonyms(C)| > 1$, contains alternative string representations for the class (synonyms). Also, each term can appear in multiple classes. (It has multiple meanings.)

Let $g_x[\mathcal{X}] = \{t \mid t \in \mathbf{r} \text{ and } t[\mathcal{X}] = x\}$.

Definition 1. A relation $\mathbf{r}$ satisfies a *synonym ontology FD* $\mathcal{X} \overset{s}{\mapsto} \mathcal{Y}$, if for each attribute A $\in \mathcal{Y}$, for each $x \in \Pi_{\mathcal{X}}(\mathbf{r})$, there exists a class C, such that $\Pi_A(g_x[\mathcal{X}]) \subseteq synonyms(C)$.

By definition, ontology FDs can be normalized similarly as traditional FDs to single attributes on the left side of a dependency. Synonym FDs subsume traditional FDs, as an ontology database in which all classes have a single string representation can be created (i.e., for all classes C, $|synonyms(C)| = 1$). In the previous work, the authors have studied *metric FDs* in the context of data verification [1] and data cleaning [2] . Metric FDs strictly generalize traditional FDs by allowing small differences controlled by a metric distance. For instance,

Algorithm 1 Verify synonym ontology FD

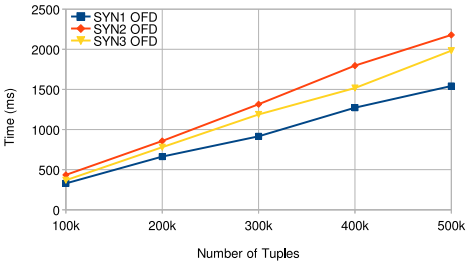
Input: Relation $\mathbf{r}$, set of attributes $\mathcal{X}$ and an attribute A

Output: true if dependency $\mathcal{X} \overset{s}{\mapsto} A$ holds, otherwise false

1: **for all** $x \in \Pi_{\mathcal{X}}(\mathbf{r})$ **do**
2: $\quad \mathbf{t} = \Pi_A(g_x[\mathcal{X}])$
3: $\quad$ Let $\mathbf{t} = \{t_1, \ldots, t_n\}$
4: $\quad$ **if** $classes(t_1) \cap \cdots \cap classes(t_n) = \emptyset$ **then**
5: $\quad\quad$ **return false**
6: $\quad$ **end if**
7: **end for**
8: **return true**

(a) Verify $\mathcal{X} \overset{s}{\mapsto} A$

#	A	B	Classes for attr B
1	a	b	$\{C,D\}$
2	a	c	$\{D,F\}$
3	a	d	$\{C,F,G\}$



(b) Sample Table and Scalability

Fig. 1: Data Verification, Sample Table and Performance Evaluation

one source might report the movie *A Beautiful Mind* to have a running time of 135 minutes, while another source might report it as 138 minutes.

While metric FDs can be defined wrt two tuples [2, 1] (i.e., if the two tuples agree on the antecedent attributes, then their consequent values must have similar but not necessarily equal values wrt the metric distance), the definition of ontology FDs must be prescribed over the entire partition identified by the unique values of the antecedent attributes. This difference is illustrated in the table in Figure 1b. The synonym ontology FD $A \overset{s}{\mapsto} B$ is falsified in this table, because even though all pairs of elements have a common class ($\{b,c\}$: D, $\{b,d\}$: C, $\{c,d\}$: F), the intersection among the entire partition over classes is empty. Furthermore, ontological similarity is not a metric as it does not satisfy identity of indiscernibles (e.g., for synonyms). Thus, ontology FDs are not a subclass of metric FDs.

We study the problem of *data verification* for ontology FDs, that is determining whether a given ontology FD holds for a given relation.

2.2 Data Verification

In order to verify that the traditional FD $\mathcal{X} \rightarrow \mathcal{Y}$ holds in a relation instance, for each $x \in \Pi_{\mathcal{X}}(\mathbf{r})$, we have to check whether $|\mathbf{t}| = 1$, where $\mathbf{t} = \Pi_A(g_x[\mathcal{X}])$. For ontology FDs, more complex algorithms are required. The choice of ontological relation (in our case synonym relationship) directly impacts the complexity of the verification algorithms. Although we study the algorithms for the data verification problem for other classes of ontology FDs, due to the space limit, we present the algorithm for synonym ontology FDs. (The remaining designed algorithms will appear in the extended version of this paper.)

To verify that the synonym ontology FD holds over a relation instance $\mathbf{r}$, we first partition $\mathbf{r}$ over $\mathcal{X}$ (see Figure 1a). Then, for each value of x in $\Pi_{\mathcal{X}}(\mathbf{r})$, we check whether the intersection of $classes(t_1), \ldots, classes(t_n)$ is not empty, where

$\mathbf{t} = \{t_1, \ldots, t_n\} = \Pi_A(g_x[\mathcal{X}])$. If this condition is satisfied then the verification algorithm returns true, otherwise it returns false. Therefore, the worst-case time complexity of the verification algorithm is quadratic in the number of tuples (similarly as for metric FDs [1]). We assume that the access to the synonym ontology is indexed and can be achieved within a constant factor.

Our experiments were run on an Intel Xeon CPU E5-2630 v3, 2.40GHz with 8GB of memory. The algorithms were implemented in the Go language. We present an evaluation of data verification algorithm for synonym ontology FDs over medical trials dataset with 500K tuples. Our evaluation focuses on the scalability and performance over different sizes of the dataset. From Figure 1b it can be concluded that the algorithm allows for the efficient data verification as well as it scales well for large datasets. (The running times are comparable to the results achieved for data verification of metric FDs in [1].)

3 Summary and Future Work

In this work, we introduced a novel class of integrity constraints called ontology FDs as well as presented the efficient algorithm for the data verification problem. In the future work we are planning to investigate the following.

- We are currently working on developing effective algorithms for data repairs [2, 4] over datasets that violate ontology FDs.
- However, we have also observed that ontological repositories may evolve over the time as applications change (e.g., a new drug appears on the market). In such environments, when an error with respect to ontology FDs arise, it is no longer clear if there is an error in the data, and the data should be repaired, or if the ontology semantics have evolved, and the ontological repositories (such as UMLS, WordNet) should be repaired. We plan to extend our framework by developing a model that allows data and ontological database repairs. We will develop classification algorithms, driven by collected statistics over the dataset that predict data versus ontological database repairs.
- We will investigate, similarly as for other constraints such as FDs and order dependencies [3], a sound and complete axiomatization for ontology FDs. We will also study the complexity of the inference problem for ontology FDs.

References

1. Koudas, N., Saha, A., Srivastava, D., Venkatasubramanian, S.: Metric functional dependencies. In: ICDE. pp. 1275–1278. IEEE (2009)
2. Prokoshyna, N., Szlichta, J., Chiang, F., Miller, R., Srivastava, D.: Combining quantitative and logical data cleaning. PVLDB 9(4), 300–311 (2015)
3. Szlichta, J., Godfrey, P., Gryz, J.: Fundamentals of order dependencies. PVLDB 5(11), 1220–1231 (2012)
4. Volkovs, M., Chiang, F., Szlichta, J., Miller, R.: Continuous data cleaning. In: ICDE. pp. 244–255. IEEE (2014)