ARTICLE IN PRESS

Discrete Applied Mathematics ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Discrete Applied Mathematics

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

Characterizing approximate-matching dependencies in formal concept analysis with pattern structures

Jaume Baixeries [a,*], Victor Codocedo [c], Mehdi Kaytoue [d,e], Amedeo Napoli [b]

[a] Departament de Ciències de la Computació, Universitat Politècnica de Catalunya, 08032 Barcelona, Catalonia, Spain
[b] LORIA (CNRS – Inria Nancy Grand-Est – Université de Lorraine), B.P. 239, 54506 Vandœuvre-lès-Nancy, France
[c] Universidad Técnica Federico Santa María, Avenida España 1680, Valparaíso, Chile
[d] Université de Lyon, CNRS, INSA-Lyon, LIRIS, UMR5205, F-69621, France
[e] Infologic, 99 Avenue de Lyon, F-26500, Bourg-lès-Valence, France

ARTICLE INFO

Article history:
Received 13 February 2016
Received in revised form 15 March 2018
Accepted 28 March 2018
Available online xxxx

Keywords:
Functional dependencies
Similarity
Tolerance relation
Formal concept analysis
Pattern structures
Attribute implications

ABSTRACT

Functional dependencies (FDs) provide valuable knowledge on the relations between attributes of a data table. A functional dependency holds when the values of an attribute can be determined by another. It has been shown that FDs can be expressed in terms of partitions of tuples that are in agreement w.r.t. the values taken by some subsets of attributes. To extend the use of FDs, several generalizations have been proposed. In this work, we study approximate-matching dependencies that generalize FDs by relaxing the constraints on the attributes, i.e. agreement is based on a similarity relation rather than on equality. Such dependencies are attracting attention in the database field since they allow uncrisping the basic notion of FDs extending its application to many different fields, such as data quality, data mining, behavior analysis, data cleaning or data partition, among others. We show that these dependencies can be formalized in the framework of Formal Concept Analysis (FCA) using a previous formalization introduced for standard FDs. Our new results state that, starting from the conceptual structure of a pattern structure, and generalizing the notion of relation between tuples, approximate-matching dependencies can be characterized as implications in a pattern concept lattice. We finally show how to use basic FCA algorithms to construct a pattern concept lattice that entails these dependencies after a slight and tractable binarization of the original data.

1. Introduction

In the relational database model, functional dependencies (FDs) are among the most popular types of dependencies since they indicate a functional relation between sets of attributes [10,30,37]: the values of a set of attributes are determined by the values of another set of attributes. Such FDs can be used to check the consistency and the quality of a database [19], but also to guide the database design [31].

However, the definition of FDs is too strict for several useful tasks, for instance when dealing with data imprecision i.e. errors and uncertainty in real-world data. To overcome this problem, different generalizations of FDs have been defined. These generalizations can be classified according to the criteria by which they relax the equality condition of FDs [11]. According to this classification, two main strategies are presented: "extent relaxation" and "attribute relaxation" (in agreement with the terminology introduced in [11]).

* Corresponding author.
E-mail address: jbaixer@cs.upc.edu (J. Baixeries).

https://doi.org/10.1016/j.dam.2018.03.073
0166-218X/

Characterizing and computing FDs are two tasks strongly related to lattice theory. For example, lattice characterizations of a set of FDs are studied in [12,14,15,29]. Following the same line, a characterization of FDs within Formal Concept Analysis (FCA) is proposed in [22]. In the latter case, FDs are shown to be in one-to-one correspondence with the set of implications of a formal context (binary table) generated from a database. However, such a formal context has a quadratic number of objects w.r.t. the tuples of the original database. To avoid this, [6] and [8] show how to use pattern structures, introduced by [21] as an extension of FCA. Moreover, in [7] it is shown how this framework can be extended to Similarity Dependencies, another generalization of FDs.

Besides FCA and implications, there are many similarities between association rules in data mining and FDs. This is discussed further in the present paper and as well in [1]. In the latter, a unifying framework in which any "well-formed" semantics for rules may be integrated is introduced. Similarly, this is also what we try to define in this paper for generalizations of FDs in the framework of FCA and Pattern Structures.

This paper presents an extended and updated version of [7] and its main objective is to give a characterization of FDs relaxing the attribute comparison within FCA thanks to the formalism of Pattern Structures. While our previous work considered similarity dependencies, we extend the characterization to the family of approximate-matching dependencies using pattern structures and tolerance relations [26,27]. Furthermore, we show that the classical FCA algorithms can be – almost directly – applied to compute approximate-matching dependencies.

This paper is organized as follows. In Section 2 we introduce our notations and the definition of FDs. We present other kinds of generalization of FDs in Section 3. In Section 4, we introduce tolerance relations and we show how the dependencies that are enumerated in Section 3 are based on tolerance relations. In Section 5 we propose a generic characterization and computation of approximate-matching dependencies in terms of Pattern Structures. In Section 6 we present a set of experiments to test the feasibility and scalability of extracting approximate-matching dependencies with pattern structures.

2. Notation and functional dependencies

We deal with datasets which are sets of tuples. Let $\mathcal{U}$ be a set of attributes and *Dom* be a set of values (a domain). For the sake of simplicity, we assume that *Dom* is a numerical set. A tuple t is a function $t : \mathcal{U} \mapsto Dom$ and a table T is a set of tuples $T \subseteq Dom^{|\mathcal{U}|}$. Sometimes the set notation is omitted and we write ab instead of $\{a, b\}$.

Given a tuple $t \in T$ and $X = \{x_1, x_2, \ldots, x_n\} \subseteq \mathcal{U}$, we have:

$$t[X] = \langle t(x_1), t(x_2), \ldots, t(x_n)\rangle .$$

$t[X]$ is called the *projection* of X onto t. In Example 1, we have $t_2[\{a, c\}] = \langle t_2(a), t_2(c)\rangle = \langle 6, 6\rangle$. The definition can also be extended to a set of tuples. Given a set of tuples $S \subseteq T$ and $X \subseteq \mathcal{U}$, we have:

$$S[X] = \{t[X] \mid t \in S\}.$$

Example 1. Table with tuples $T = \{t_1, t_2, t_3, t_4\}$ and attributes $\mathcal{U} = \{a, b, c, d\}$.

id	a	b	c	d
t_1	3	5	6	3
t_2	6	5	6	5
t_3	3	10	6	3
t_4	6	5	9	5

We now formally introduce functional dependencies [37].

Definition 1. Let T be a set of tuples (or a data table), and $X, Y \subseteq \mathcal{U}$. A *functional dependency (FD)* $X \to Y$ holds in T if:

$$\forall t, t' \in T : t[X] = t'[X] \Rightarrow t[Y] = t'[Y].$$

For example, the functional dependencies $a \to d$ and $d \to a$ hold in the table of Example 1, whereas the functional dependency $a \to c$ does not hold since $t_2(a) = t_4(a)$ but $t_2(c) \neq t_4(c)$.

3. Generalizations of functional dependencies

Functional dependencies tell us which attributes are determined by other attributes. As such, FDs are mainly used in databases to determine which attributes are the *keys* of a dataset, i.e. the minimal sets of attributes (if any) determining all other attributes. This information is necessary for maintaining the consistency of the whole database. Moreover, this information can also be useful in data analysis or in data classification, because of the semantics attached to the "determined by" relationship.

However, in practical applications we usually have datasets that contain imprecise or uncertain information. Here, we do not mean *false* information, but information that may contain errors. For example, let us consider a dataset containing information about the name and social security number (SSN) of citizens. Although SSN is supposed to be unique for each

Please cite this article in press as: J. Baixeries, et al., Characterizing approximate-matching dependencies in formal concept analysis with pattern structures, Discrete Applied Mathematics (2018), https://doi.org/10.1016/j.dam.2018.03.073.