International Journal of Approximate Reasoning 63 (2015) 89–100

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

www.elsevier.com/locate/ijar

Interactive error correction in implicative theories

Sergei O. Kuznetsov [a], Artem Revenko [a,b,*]

[a] National Research University Higher School of Economics, Pokrovskiy bd. 11, 109028 Moscow, Russia
[b] Technische Universität Dresden, Zellescher Weg 12-14, 01069 Dresden, Germany

ARTICLE INFO

Article history:
Received 20 October 2014
Received in revised form 3 April 2015
Accepted 10 June 2015
Available online 16 June 2015

Keywords:
Implicative theory
Error correction
Closure system
Formal concept analysis

ABSTRACT

Errors in implicative theories coming from binary data are studied. First, two classes of errors that may affect implicative theories are singled out. Two approaches for finding errors of these classes are proposed, both of them based on methods of Formal Concept Analysis. The first approach uses the cardinality minimal (canonical or Duquenne–Guigues) implication base. The construction of such a base is computationally intractable. Using an alternative approach one checks possible errors on the fly in polynomial time via computing closures of subsets of attributes. Both approaches are interactive, based on questions about the validity of certain implications. Results of computer experiments are presented and discussed.

1. Introduction

1.1. Motivation

Implicative theories consisting of formulas of the form “if A, then B” provide a standard way for describing the structure of domain knowledge. They are extensively used in various research areas, e.g., biology [18], pharmacology [6,5], semantic web [19], knowledge discovery [12,34], decision making [26], classification [24], ontology engineering [2]. In many cases the exactness of rules plays a crucial role, for example in research related to strictly formalized domains like Boolean algebras [22], algebraic lattices [7], or algebraic identities [27].

In many applications an exact implicative theory is constructed from a piece of available data. It is well known that a single mistake in this data can drastically change the resulting implicative theory [14] (the same is true for association rules if there are some exceptions and an error). The implicative theory is not going to recover from this error even if further error-free data is added to the underlying set. Therefore, implicative theories are not error tolerant. However, in the real-world applications, especially if multiple users are expected to work with data, one cannot guarantee the absence of errors. More than that, someone may be willing to spoil the result on purpose by adding erroneous instances, in order to prevent from discovering valid implications. Therefore, a procedure for recovering from errors is essential for the usage of implicative theories.

Here we assume that in the beginning there is already some data on hands and new data arrives in the work flow. The goal is to guarantee the correctness of the implicative theory with respect to the initial data which are considered to be reliable. We do not assume that a user, which is going to work with the data and the implicative theory, is always able to explicitly state any knowledge about data domain or has any knowledge about methods in use. That is why it is

[*] Corresponding author at: Technische Universität Dresden, Zellescher Weg 12-14, 01069 Dresden, Germany.
E-mail addresses: skuznetsov@hse.ru (S.O. Kuznetsov), artem_viktorovich.revenko@mailbox.tu-dresden.de (A. Revenko).

http://dx.doi.org/10.1016/j.ijar.2015.06.004
0888-613X/© 2015 Elsevier Inc. All rights reserved.

Name	Sex	Year of Birth	Lawful Age	...
Helga	f	1995	n	...
Daria	f	1980	y	...
Patrick	m	1986	y	...
John	m	1996	n	...
		⇑		
George	m	1980	n	...

Fig. 1. Data table and new entry from Example 1.

important to develop a transparent and easy method for error correction. In particular, it is important to find and output possible errors in a human understandable form. To attain this goal a natural framework can be that of Formal Concept Analysis (FCA) [14], where methods and algorithms for finding implicative theories of binary data (formal contexts) are well elaborated and widely used [13,30].

Example 1. To illustrate our ideas we provide a use-case example. Let there be data from Fig. 1 on hands. New data is coming from an untrusted source and it is intended to be added to the existing data. The user expects possible errors in new data, however, the user is not able to check every single entry (possibly, due to a large number of columns). The solution we propose in this paper would output the question: Does 'Year of Birth: 1980' imply 'Lawful Age'? As we are now in year 2015, the answer is obviously 'Yes' and, therefore, an error is revealed.

1.2. Related work

Methods for imputing missing values are well studied. In [33] and [31] detailed overviews of existing techniques are presented. Among others there are techniques of ignoring entries with missing values, imputing average values, and more complicated ones such as decision trees, neural networks [31], Nearest Neighbor approach [16]. Having a missing value, there is no need to search for an error, as it is clear from the problem statement which value should be changed (or imputed). An approach proposed in this paper bares some similarity to the Nearest Neighbor method, but aims at solving a different task. Besides that, the imputation techniques (like, e.g. averaging) are mostly not relevant for binary data.

Error finding and eliminating are widely discussed in various fields of computer science. The problems of lineage or data provenance, where one needs to explain errors, trace reasons for a query, etc. are well known in KDD domain [32]. These techniques are very useful and efficient, however, they are not appropriate for correcting errors in binary data tables.

In [9] an impressive way of using expert knowledge presented in the form of editing rules and certain regions for databases are surveyed. Information in the form of editing rules prevents the errors from getting in to the database. The approach presented in this paper aims at finding and correcting errors without any previously formalized knowledge.

The paper [10] presents an interesting approach to dealing with mistakes in answering questions (like the ones we will discuss below) in the process of knowledge base completion within the framework of Description Logics. This approach allows recovering from such mistakes in such an effective manner that the information input is used upon mistake recovery. However, the detection and correction of mistakes is left to pinpointing.

Pinpointing is a very helpful technique for recovering from inconsistencies. The goal of pinpointing is the following: for a given inconsistent set of rules (not only implicative) find minimal inconsistent subsets [3,23]. The inconsistency is detected via checking if a certain erroneous consequence holds. This technique is successfully applied in different description logics. The complexity of pinpointing is normally beyond polynomial. An approach introduced in this paper (Section 4, base approach) is closely related to pinpointing; it proceeds from knowledge base constructed from data. The complexity is also beyond polynomial. However, an alternative approach (Section 4, closure approach) takes the advantage of having the data and proposes a polynomial-time solution. In this work we do not modify the knowledge base directly, but we correct the errors in data in such a way that the corresponding implicative theory becomes error-free.

As implicative theories is another view of Horn theories [14], the problem of finding explanations in Horn theories turns out to be closely connected to our problem. Namely, an entry in the binary data table can also be considered as a fact to be explained. In [17] it is shown that such explanations may be found in polynomial time. However, here we aim at explaining existence or absence of all attributes at the same time. Also we state our task and our solutions in a different language and provide algorithms for practical usage. The case of negative attributes is not covered in [17] as opposed to this work.

The present paper is a follow-up work to [28].

Remark 1. In this paper we assume that we can put questions to an expert in the domain who gives correct answers.

Remark 2. All sets and contexts we consider in this paper are assumed to be finite, which practically means an obvious constraint on finiteness of data at hand.