Applied Soft Computing 37 (2015) 467–478

Contents lists available at ScienceDirect

Applied Soft Computing

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

Relation-based granules to represent relational data and patterns

Piotr Hońko*

Faculty of Computer Science, Bialystok University of Technology, Wiejska 45A, 15-351 Białystok, Poland

ARTICLE INFO

Article history:
Received 27 June 2014
Received in revised form 6 August 2015
Accepted 24 August 2015
Available online 2 September 2015

Keywords:
Granular computing
Data mining
Relational databases

ABSTRACT

The complex structure of relational data makes the process of knowledge discovery from data a more challenging task compared with the single table data structure. The usefulness of granular computing based approaches to mining data stored in a single table is a driving force for adapting this method to relational data. This paper proposes relation-based granules that are defined in a granular computing based approach to mining relational data. The relations are used to represent relational data and patterns to be discovered. Thanks to this representation, the generation of patterns can be speeded up. The representation also makes it possible to discover richer knowledge from relational data.

1. Introduction

Granular computing provides a general framework for problem solving. It covers theories, methodologies, techniques, and tools that make use of granules [1]. They are, in general, understood as collections formed in the process of a semantically meaningful grouping of elements based on their indistinguishability, similarity, proximity or functionality [2].

The main idea of granular computing is to make it possible to view the same problem at many levels of granularity. Switching between different levels enables to choose the representation best matching to the problem. A more specific level granularity may reveal more detailed information, whereas a more abstract level granularity may improve a problem solution thanks to omitting irrelevant details.

A granular computing approach has successfully found application in data mining (e.g. [3–7]). Some attempts have also been made to adapt the idea of granulation to mining data stored in a relational structure (e.g. [8–10]).

Granulation tools can provide an alternative representation of the data to be mined. The primitives in this case are defined not by attribute values, but by granules of entities. Using granules, one can form collections of objects that share the same features (e.g. attribute values). A granular representation of the data facilitates the generation of patterns. Since elementary granules reveal basic features hidden in data, they are used as atoms in the construction of patterns.

A granular computing based approach for data mining can be defined using a description language for information granules [11]. The data is primarily stored in an information system. However, it can also be represented in a granular form that is constructed using atomic formulas of the language. Each information granule is characterized by a pair of syntax and semantics. The former is defined by a formula constructed using attributes and values that describe objects, whereas the latter is understood as the set of objects that satisfy the formula. Information granules can be used to express patterns hidden in the data.

The above approach can be upgraded to a relational case by expanding the description language by additional atomic formulas that identify pairs of joinable objects from different database tables [12,13]. The data in such an approach is represented by a compound information system that is a combination of particular information systems (each corresponding to one database table). These systems are combined according to the connections that occur among database tables.

The compound information system can be directly mined or can be beforehand transformed into a granular form. The former facilitates the construction of patterns over many tables since the connection among tables are included in the system; however, elementary granules that show objects sharing the same features are not contained. The latter, in turn, includes elementary granules (each associated with one table or with two tables to show the connection between them) but the construction of relational patterns over the description language requires granules to be defined so that each of them is associated with all tables under consideration.

To construct a relational data representation that is more coherent and useful for pattern discovery, relation-based granules are proposed in this study. They are formed using relations that join relational information granules with their features. They are used to represent both the data and patters. Relation-based granules are more informative than the granules based on which they are constructed. They include information about how a given granule

* Tel.: +48 857469059.
E-mail address: p.honko@pb.edu.pl

http://dx.doi.org/10.1016/j.asoc.2015.08.045
1568-4946/

can alternatively be joined with another from a different information system. The relations used to represent data are fundamental components of patterns. Since the relations express basic features of objects, the process of the generation of patterns can be speeded up. Furthermore, the structure of relation-based granules facilities the formation of more advanced conditions. They correspond to those that can be formed by using aggregation functions in relational databases. Therefore, patterns constructed based on such relations show richer knowledge than standard relational patterns.

The organization of the remaining part of the paper is as follows. Section 2 introduces compound information systems and description languages defined for relational information granules. Sections 3 and 4 propose relation-based granules and show their application to the representation of relational data and patterns. In Section 5, the approach is evaluated by analyzing its time complexity. Section 6 compares the approach with other related approaches. Finally, Section 7 concludes the study.

2. Description languages for relational information granules

This section introduces the definitions of information systems and their description languages defined for relational data. The languages are expansions of the description language defined for data stored in the standard information system [11].

Throughout this paper the following running example will be used.

Example 1. Given a database for the customers of a grocery store.

customer

Id	Name	Age	Gender	Income	Class
1	Adam Smith	30	Male	1500	Yes
2	Tina Jackson	33	Female	2500	Yes
3	Ann Thompson	30	Female	1800	No
4	Susan Clark	30	Female	1800	Yes
5	Eve Smith	26	Female	2500	Yes
6	John Clark	29	Male	3000	Yes
7	Michael Thompson	30	Male	1800	No

married_to

Id	Cust_id$_1$	Cust_id$_2$
1	5	1
2	6	4
3	3	7

product

Id	Name	Price
1	Bread	2.00
2	Butter	3.50
3	Milk	2.50
4	Tea	5.00
5	Coffee	6.00
6	Cigarettes	6.50

purchase

Id	*Cust_id*	*Prod_id*	Amount	Date
1	1	1	1	24/06
2	1	3	2	24/06
3	2	1	1	25/06
4	2	3	1	26/06
5	4	6	1	26/06
6	4	2	3	27/06
7	5	5	2	27/06
8	6	4	1	27/06

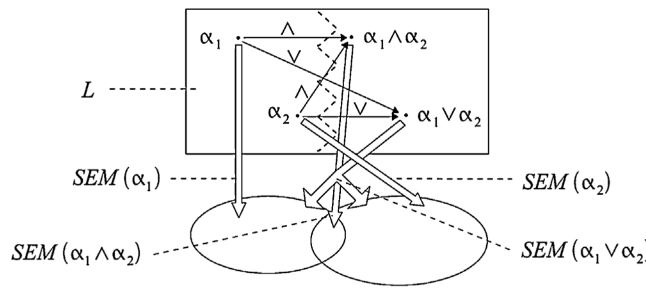

Fig. 1. Expansion of language L by the conjunction and disjunction of atomic formulas [12].

2.1. Information system

The following defines an information system corresponding to a database table.

Definition 2 *([14,12] (Information system for database table)).* An information system for a database table with the schema $R_i(id, a_1, \ldots, a_m)$ is a pair $IS_i = (U_i, A_i)$, where $U_i = \{x : x \in R_i\}$ and $A_i = \{id, a_1, a_2, \ldots, a_m\}$.[1]

To construct granules for an information system, logical formulas over some language are used. Namely, granules are defined by formulas which are used to express the properties of the objects from the granules. Therefore, an information system $IS = (U, A)$ is given along with the following:

- A set of formulas Φ over some language.
- A function $SEM : \Phi \to P(U)$.

The SEM functions shows the meaning of a formula, i.e. it returns a subset of U that consists of objects satisfying a given formula from Φ.

The following general definitions of the syntax and semantics of a language are used. They will be applied in the further part of the paper for defining particular languages. Let L be a language such that the syntax and semantics of an atomic formula $\alpha \in L$ and its negation $\neg\alpha \in L$ are defined.

Definition 3 *([12]).* The syntax and semantics of a language L are defined recursively by those of $\alpha \in L$ and $\neg\alpha \in L$, and by

1. $\alpha_1, \alpha_2 \in L \Rightarrow \alpha_1 \wedge \alpha_2 \in L$ and $SEM(\alpha_1 \wedge \alpha_2) = SEM(\alpha_1) \cap SEM(\alpha_2)$;
2. $\alpha_1, \alpha_2 \in L \Rightarrow \alpha_1 \vee \alpha_2 \in L$ and $SEM(\alpha_1 \vee \alpha_2) = SEM(\alpha_1) \cup SEM(\alpha_2)$.

In Fig. 1, language L is expanded for any atomic formulas by their conjunction and disjunction (black arrows labeled with $\wedge$ and $\vee$). The semantics of new formulas is constructed based on that of the atomic formulas (white arrows).

Let now $L = L_1 \cup \cdots \cup L_k$ $(k > 1)$ be a language such that for each L_i $(1 \le i \le k)$ the syntax and semantics are defined.

Definition 4 *([12]).* The syntax and semantics of a language L are defined recursively by those of each L_i and by the following

1. $\alpha \in L_i \Rightarrow \alpha \in L$ and $SEM(\alpha) = SEM_i(\alpha)$[2];
2. $\alpha \in L \Rightarrow \neg\alpha \in L$ and $SEM(\neg\alpha) = SEM_i(\neg\alpha)$, where $\alpha \in L_i$;
3. $\alpha_1, \alpha_2 \in L \Rightarrow \alpha_1 \wedge \alpha_2 \in L$ and $SEM(\alpha_1 \wedge \alpha_2) = SEM(\alpha_1) \cap SEM(\alpha_2)$;
4. $\alpha_1, \alpha_2 \in L \Rightarrow \alpha_1 \vee \alpha_2 \in L$ and $SEM(\alpha_1 \vee \alpha_2) = SEM(\alpha_1) \cup SEM(\alpha_2)$.

[1] The index (i.e. the relation identifier) is omitted if this does not lead to a confusion.
[2] SEM_i is the semantics of L_i.