Artificial Intelligence Review (2019) 52:2705–2742
https://doi.org/10.1007/s10462-018-9643-1

Recent granular computing frameworks for mining relational data

Piotr Hońko[1]

Published online: 30 June 2018

Abstract
A lot of data currently being collected is stored in databases with a relational structure. The process of knowledge discovery from such data is a more challenging task compared with single table data. Granular computing, which has successfully been applied to mining data storable in single tables, is a promising direction for discovering knowledge from relational data. This paper summarizes some recent developments in the area of application of granular computing to mining relational data. Four granular computing frameworks for processing relational data are introduced and compared. The paper shows how each of the frameworks represents relational data, constructs information granules and build patterns based on the granules. A generic system that can employ any of the frameworks to discover knowledge from relational data is also outlined.

Keywords Relational data mining · Granular computing · Information systems · Association discovery · Classification

1 Introduction

Many techniques in data mining are usually designed for individual problems such as classification, clustering, or association discovery. One of the main challenges of this field is, however, to develop a common theory (Yang and Wu 2006) that encompasses different data mining tasks. A theoretical unifying framework can provide a concise look at the field of data mining as well as contribute to improving the process of knowledge discovery.

The problem of developing a unified framework is a more complicated issue if the data to be mined is stored in databases with a relational structure. Such data is distributed over multiple tables, and the central issue in the specification of a relational data mining problem is the definition of a model of the data. Such a model directly determines the type of patterns that will be considered and thus the direction of the search.

✉ Piotr Hońko
p.honko@pb.edu.pl

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

An example of a framework that unifies data mining tasks performed for data stored in single table databases is one constructed in the field of granular computing (Skowron and Stepaniuk 2001). The basic idea underlying the field is the way data is perceived. Many approaches for data analysis and processing treat attribute values as primitives. One can observe that for the purposes of identification of subgroups of objects that share the same features, it is convenient to examine not particular features (i.e. attribute values) of an object, but objects that have the same feature. In such an approach, a primitive is not an attribute value but a granule of objects sharing the value.

Much research has been devoted to granular computing in data mining (see, e.g., Bargiela and Pedrycz 2003; Lin and Zadeh 2004; Lin 2005; Pedrycz et al. 2008; Al-Hmouz et al. 2015; Pal et al. 2015). Therefore, the usefulness of granular computing based approaches to mining data stored in single tables has become a driving force for adapting this paradigm to relational data.

The goal of this paper is to summarize and compare recent granular computing frameworks for processing relational data. The paper shows how each of the four described approaches presents relational data, defines information granules for relational data, and constructs relational patterns. Since the frameworks are not equipped with a pattern generation algorithm, a generic system that shows how a granular computing framework can be used in the whole process of relational knowledge discovery is introduced.

The remaining sections of this paper are organized as follows. Section 2 introduces to relational data mining and granular computing. It also provides an overview of approaches to processing relational data using granular computing tools. Sections 3–6 describe granular computing frameworks for mining relational data. The comparison and evaluation of the frameworks are provided in Sect. 7. Concluding remarks are given in Sect. 8.

2 Relational data mining and granular computing

This section provides an introduction to relational data mining and granular computing. It also reviews granular computing approaches to mining relational data.

2.1 Relational data mining

Multi-relational data mining (MRDM) (Džeroski and Lavrač 2001b) concerns knowledge discovery from relational databases consisting of multiple relations (tables). MRDM aims to integrate methods from existing fields applied to an analysis of data represented by multiple relations; producing new techniques for mining multi-relational data.

MRDM can be treated as an extension of standard data mining to a relational case. One of relational database tables, called target table, is usually an equivalent of a single table database used in non-relational data mining. The remaining tables, called background tables, include additional data directly or indirectly joined with the objects from the target table. Background tables are useful or even necessary to properly describe target objects or to distinguish them if they come from different classes.

One can indicate two commonly used frameworks for mining relational data: inductive logic programming and relational database theory frameworks.

Early approaches for pattern discovery in relational data were defined in an inductive logic programming (ILP) framework (Džeroski and Lavrač 2001a).

ILP is a research field at the intersection of machine learning and logic programming. It provides a formal framework as well as practical algorithms for learning in an inductive way relational descriptions from data represented by target examples and background knowledge.

In ILP, data and induced patterns are represented as formulas in a first-order language. Data is stored in deductive databases, where relations can be defined extensionally as sets of ground facts and intensionally as sets of database clauses. Patterns are typically expressed as logic programs, i.e., sets of Horn clauses.

In ILP, the pattern structure is determined by the so-called declarative bias. It imposes some constraints on the patterns to be discovered. Thanks to the bias, one can determine which relations and how many times may be used in patterns; how to replace a relation attribute with a variable; what values a relation variable may take, and the like.

An alternative framework (Knobbe 2006; Knobbe et al. 2000) for discovering patterns in relational data is defined in a relational database theory (RDB). In relational database, relations are usually defined extensionally as sets of tuples of constants. However, they can also be defined intensionally as sets of views. Relational patterns discovered in a relational database can be expressed as SQL queries.

Unlike in the ILP framework, a specification of the pattern structure is not required. Instead, the patterns are specified by the relationships that occur between the database entities and are shown by an entity-relationship diagram. Alternatively, a class diagram that is a part of Unified Modeling Language (UML) (Knobbe et al. 2000) is used to express a bias. UML class diagram shows how associations (i.e., structural relationships) between given classes (which correspond to database tables) determine how objects in each class relate to objects in another class. Moreover, multiplicities of associations are also considered. Such an association multiplicity provides information how many objects in one class are related to a single object in another, and vice versa.

To deal with data with a relational structure, MRDM often employs tools from standard data mining. One of these trends is to upgrade a data mining algorithm to a relational case (Van Laer and De Raedt 2001). The idea is to preserve as many features of the original algorithm as possible. It means that the general mechanism of data mining is used directly (e.g. search strategy, pattern construction method) and only crucial notions are adjusted (e.g. pattern representation, pattern satisfiability).

Another trend, called propositionalization (Lavrač et al. 1991; Van Laer and De Raedt 2001; Kramer et al. 2001; Kuželka and Železný 2008), is to transform relational data into a single table and then to use a standard data mining algorithm to discover patterns. They can alternatively be transformed into a relational form. The crucial task of this approach is to find essential features over relational data to be used to create attributes of a single table. They can be constructed using relational techniques such as frequent pattern discovery (Blaťák 2005) or subgroup discovery (Železný and Lavrač 2006).

MRDM also extends approaches that were not developed in the data mining framework but were adapted for analyzing data storable in single table databases. Those that have been considerably employed for mining relational data are graph-based data mining (Washio and Motoda 2003) and formal concept analysis (Ganter et al. 2005).

2.1.1 Graph-based relational data mining approaches

Relational data can successfully be modeled as graphs (Ketkar et al. 2005). For example, objects of tables can be encoded as vertices and relationships between the objects as edges. Relational data graphs can be mined using existing techniques of graph theory.

The problem of concept discovery in multi-relational data was addressed using a graph-based approach (Kavurucu et al. 2016). The proposed graph-based concept discovery system generates disconnected graph structures for each target relation and its related background knowledge, which are initially stored in a relational database. Then the structures are used to generate a summary graph, which is finally traversed to find concept descriptors.

Subdue (Cook and Holder 2000) is a system for discovering interesting substructures in structural data. The discovered substructures are organized into a hierarchical description of the structural regularities in the data. Having a graph of relational data divided into positive and negative classes, the system is also able to find a subgraph that summarizes the positive class distinguishing it from the negative one.

An efficient version of Subdue, a graph-based relational learning algorithm, was proposed in Guo et al. (2007). The proposal introduces optimizations for reducing the subgraph isomorphism computation and the numbers of subgraph isomorphism testing, which are the major source of complexity in Subdue.

An approach, implemented in the Subdue system, for learning patterns in relational data represented as a graph is proposed in Holder et al. (2005). These patterns can take the form of prevalent subgraphs, a hierarchical, conceptual clustering of subgraphs, or a subgraph that can distinguish positive graphs from negative graphs. The approach was applied in domains related to homeland security and social network analysis.

2.1.2 Formal concept analysis for relational data mining

Formal concept analysis (FCA) (Ganter and Wille 1999) is a tool to classify a set of objects that are described by attributes and presented as a formal context, i.e. a triple of a set of objects, an attribute set, and a binary relation that shows if an object possesses a given attribute. A concept is defined as a pair of a subset of objects (X) and a subset of the attribute set (B) such that each object from X is in the relation with each attribute from B, and vice versa. All concepts can be hierarchically ordered using a concept lattice.

FCA can be used to construct for relational data a formal context that not only expresses which objects have which attributes but also shows interactions among objects of different database tables.

One of the approaches, called relational concept analysis (RCA), relies on extending FCA to deal with relational data. RCA constructs conceptual abstractions from objects described by both own properties and inter-object links while dealing with several sorts of objects. RCA produces lattices for each category of objects and those lattices are connected via relational attributes that are abstractions of the initial links.

To discover relational concept a relational context family, i.e., a set of formal contexts whose objects are related by links, was used in Huchard et al. (2007). Formal concepts to be discovered are characterized by both the shared features of members objects and by their relations to other formal concepts.

RCA was used in Dolques et al. (2016) to improve knowledge discovery from relational data. The idea of the approach is to reduce the complexity of relational data and to obtain relevant results faster by computing less lattices (preferably only the lattices that are of interest).

RCA was also used in a query-based navigation approach to helps an expert to explore a concept lattice family (Azmeh et al. 2011).

Other extensions of FCA are described below.

The problem of mining quantitative association rules from a multi-relational database was considered in Nagao and Seki (2016). To handle numerical data in a precise and efficient way,

one of the notions of FCA, i.e. closed interval patterns, was used to construct logical conjunctions with interval constraints. The proposed algorithm returns quantitative association rules that satisfy given minimum support and confidence.

An extension of FCA which considers links between relational objects of different types was proposed in Kötters (2011). It uses the notion of linked context family that enables to reflect the schema of the database to a high degree. Thanks to it a query graph can be refined to a more informative, summarized view of the underlying data without exposing too much information at once.

Ferré et al. (2005) relations are introduced using concept lattices and their labeling. The information from both object and relation contexts is combined in a single concept lattice. Relational features express properties over objects w.r.t. their related objects and can be used both for redefining a set of objects, as usual, and for traversing some relation from a set of objects to another.

Another approach, which has been relatively recently adapted to mine relational data, is the paradigm of granular computing. This approach will be described in more detail in the remaining subsections.

2.2 Granular computing

When analyzing data to discover knowledge, regardless of the tool used, we usually aggregate the objects with common features into the same clusters (i.e., groups). Such clusters can be treated as information derived from the database, which is, in turn, the basis for the discovery of knowledge. The clusters can be obtained in a variety of ways depending, among others, on the task to be performed. Moreover, one can receive many different partitions of the universe, i.e., families of clusters, for the same task. The choice of the most proper partition can depend on which solution accuracy of the problem under consideration is sufficient. The challenge is thus to develop a framework for constructing and processing such clusters of data.

A field within which frameworks are developed for problem solving by the use of granules (e.g., clusters of data) is granular computing (GC) (Bargiela and Pedrycz 2003; Pedrycz et al. 2008). This is a relatively new, rapidly growing field of research (see, e.g., Yao et al. 2013; Pal and Banerjee 2013; Li et al. 2015; Hu et al. 2015; Wilke and Portmann 2016). It can be viewed as a label of theories, methodologies, techniques, and tools that make use of granules in the process of problem solving (Yao 2000). Problems recently addressed in GC ranges from developing theoretical foundations (e.g. Dubois and Prade 2016; Yao 2016; Mendel 2016), through dealing with uncertainty (e.g. Kreinovich 2016; Ciucci 2016; Livi and Sadeghian 2016), supervised and unsupervised learning (Peters and Weber 2016; Antonelli et al. 2016; Lingras et al. 2016; Apolloni et al. 2016), to interactive granular computing (e.g. Skowron et al. 2016; Wilke and Portmann 2016; Apolloni et al. 2016).

A granule is a collection of entities drawn together by indistinguishability, similarity, proximity or functionality (Zadeh 1997). Therefore, a granule can be defined as any object, subset, class, or cluster of a given universe. The process of the formation of granules is called granulation. To clearly differentiate granulation from clustering, the semantic aspect of GC is taken into account. Therefore, we treat information granulation as a semantically meaningful grouping of elements based on a given criterion (Bargiela and Pedrycz 2008). An information granule can be represented by an expression of the form (*name*, *content*), where *name* is the granule identifier and *content* is a set of objects identified by *name* (Stepaniuk 2008).

Granulation can be performed by applying a top-down or a bottom-up method. The former concerns the process of dividing a larger granule into smaller and lower level granules, and

the latter the process of forming a larger and higher level granule with smaller and lower level sub-granules (Yao 2005).

One can obtain many granularities of the same universe which differ in their levels. A granule of high-level granularity, i.e., a high-level granule represents a more abstract concept, and a low-level granule a more specific one. A basic task of GC is to switch between different levels of granularity. A more specific level granularity may reveal more detailed information. On the other hand, a more abstract level granularity may improve a problem's solution thanks to omitting irrelevant details.

Information granules are often constructed using approaches that were originally defined in separation from granular computing. The most frequently used are rough set (e.g. Eissa et al. 2016; Skowron et al. 2012; Stepaniuk 2008; Pal et al. 2005) and fuzzy set (e.g. Ray et al. 2016; Kundu and Pal 2015; Pal et al. 2012; Ganivada et al. 2011) approaches.

Rough set theory (Pawlak 1991) was proposed as a mathematical tool to deal with uncertainty in data. The key concept is an approximation space, which, in its simplest form, is a pair of a universe (a set of objects under consideration) and a relation that divides the universe into (usually disjoint) subsets. Each of them can be treated as an elementary granule. Any concept (represented by a subset of the universe) that is not certain, i.e. indefinable using any union of elementary granules, can be represented by a rough set. This is a pair of the maximal union of elementary granules included in the concept (lower approximation) and the minimal union of elementary granules that includes the concept (upper approximation). The lower and upper approximations include, respectively, objects that *certainly* and *possibly* belong to the concept. New knowledge about the concept can be derived from the approximations. For example, decision rules constructed based on the lower approximation show features of objects that certainly belong to the concept, whereas those generated from the upper one describe objects whose membership in the concept is possible.

Fuzzy set theory (Zadeh 1997) was proposed as an extension of classical set theory by a generalization of the membership function, which may take not only either 0 or 1 but also any intermediate value. It enables to define for any element its membership degree to a given concept, i.e. fuzzy set. Since the membership function can be any function mapping a given set of elements to [0, 1], a wide range of concepts can be modeled using fuzzy sets. Like in rough set theory, it is possible to deal with uncertainty in data. An object that not certainly belongs to defined concepts can be classified based on its membership degree to the concepts.

A granular aspect of fuzzy sets can be seen e.g. when measuring the similarity of elements of a fuzzy set. The membership value of an element enables to define the degree of its similarity to other elements. Therefore, a fuzzy granule is defined as a clump of fuzzy set elements drawn together by similarity.

A similarity function can, in turn, be used to form granules based on the elements.

2.3 Granular computing in relational data mining

Rough set theory is one of the main tools of GC, which has relatively often been applied to relational data.

Yan et al. (2010) the approximation space is defined as a triple of two distinct universes and binary relation that is a subset of the Cartesian product of the universes. Approximations are defined for a subset of one of the universes. They include objects from the other universe that are in the relation with objects of the subset. Such an approach can be viewed as a generalization of that introduced in Yao (2004) where approximations are defined in a formal

context that is a triple of the universe of objects, universe of attributes, and a binary relation between the universes.

Lan and Xiangzhi (2007) approximations are defined in an information system that is a pair of the double universe (the Cartesian product of two particular universes) and the attribute set. Equivalence classes defined over the double universe are used to construct approximations of a double universe subset.

Additionally, a constrained version of the information system is introduced. It is a triple of the double universe, a constraint relation on the universe, and the attribute set.

To handle with data stored in many tables a multi-table information system is proposed in Milton et al. (2005). The system is a finite set of tables (each table is viewed as an information system). Approximations are defined for a subset of the universe of one specified table, this is, the decision table. Elementary sets of a given universe are used to define the approximations. Indiscernibility of objects from the decision table is defined using the information available in all the tables of the multi-table information system.

An approach that is not oriented to a particular granular computing tool and can be used for relational data was introduced in Lin (2008). A relational database can be represented by a relational granular model which is the pair of the universe (a family of classical sets, called the family of universes) and the collection of relations on the Cartesian product of sets from the universe. The sets from the universe correspond to objects of a relational database, and the relations (of various arities) define constraints for these objects. Some granules considered in fields such as data mining, web/text mining, and social networks can be modeled into the relational granular model.

The approaches described above provide useful tools to deal with data stored in a relational structure; however, they are one task oriented (e.g. dealing with uncertainty in data) or do not study extensively or at all the problem of constructing information granules for improving mining relational data.

The remaining part of this paper shows granular computing approaches that are dedicated to relational data or can be used to processes such a kind of data.

3 Constrained sums of information systems

This section presents the first granular computing framework developed for processing complex data that is applicable to relational data (Skowron and Stepaniuk 2004). In spite of the fact that the framework was not proposed recently, it is described and used in this paper as a reference point to better show the development of applications of granular computing tools for relational data.

The approach can be summarized as follows. Each of the relations under consideration is represented by an information system (Definition 1). The universe of the system includes tuples of the relation, whereas the attribute set consists of the names of relation's attributes. A set of relations is represented by a sum of information systems (Definition 2). The universes of particular systems are joined using the Cartesian product, the attribute sets are merged using the union operations, preserving the distinguishability of attributes having the same name but coming from different information systems.

To make it possible to express relationships between/among particular universes, the sum of information systems is constrained (Definition 3). Namely, tuples of the complex universe (the Cartesian product of particular universes) are filtered according to a given constraint.

In particular, such a constraint enables to express relationships of a relational database, e.g. natural join of two or more tables.

Granules in a (constrained) sum of information systems are constructed based on formulas, i.e. Boolean combinations of descriptors over particular information systems. An atomic formula is the expression of the form (a, v), where a is an attribute, and v is one of its possible values. More advanced formulas are constructed recursively using logical operators (Definition 4). For each formula its semantics is expressed as the set of objects that satisfy the formula (Definition 5). A pair of a formula and its semantics is treated as a granule (Definition 6).

A granule that comprises a formula constructed using the conjunction operator is seen as a basic pattern. An atomic component in a pattern may be defined either in a particular information system or as an element of the constraint of a constrained sum of information systems (Definition 7). More advanced patterns are formed according to the standard principle of building data mining patterns (e.g. association or classification rules).

The details of the approach are given in the following subsections.

3.1 Relational data representation

The starting point for constructing a structure for storing complex data is an information system defined for a single table.

Definition 1 (Pawlak 1991; *Information system*) An information system is a pair $IS = (U, A)$, where U is a non-empty finite set of objects, called the universe, and A is a non-empty finite set of attributes. Each attribute $a \in A$ is treated as a function $a : U \to V_a$, where V_a is the value set of a, i.e. $a(x) = v$ where $x \in U$ and $v \in V_a$.

The structure used to store data is a combination of information systems, each corresponding to one database table.

Definition 2 (*Sum of information systems*) Let $IS_i = (U_i, A_i)$ for $i = 1, \ldots, k$ be information systems. The sum of IS_i $(i = 1, \ldots, k)$, denoted by $+(IS_1, \ldots, IS_k)$, is defined by

1. The objects of $+(IS_1, \ldots, IS_k)$ consist of tuples $(x_1, \ldots, x_k)$ of objects from IS_i, i.e. $U = U_1 \times \cdots \times U_k$.
2. The attributes of $+(IS_1, \ldots, IS_k)$ consist of the attributes of IS_i where distinct copies are made for attributes in common.

To define dependencies among particular systems a constrained sum of information systems is introduced. It is done using a constraint relation that shows which tuples of objects, each coming from a different information system, can be considered as a whole, e.g. a customer and a product can be joined by a constraint relation that indicates which customer buys which product.

Definition 3 (*Constrained sum of information systems*) Let $IS_i = (U_i, A_i)$ for $i = 1, \ldots, k$ be information systems and let $R \subseteq U_1 \times \cdots \times U_k$ be a constraint relation. The constrained sum of IS_i $(i = 1, \ldots, k)$, denoted by $+_R(IS_1, \ldots, IS_k)$ is defined by

1. The objects of $+_R(IS_1, \ldots, IS_k)$ consist of k-tuples $(x_1, \ldots, x_k)$ of objects from R, i.e. all objects from $U_1 \times \cdots \times U_k$ satisfying the constraint R.
2. The attributes of $+_R(IS_1, \ldots, IS_k)$ consist of the attributes of $A_1, \ldots, A_k$ where distinct copies are made for attributes in common.

The system form Definition 2 can be used to represent a database whose tables are connected using the Cartesian product, whereas the system from Definition 3 is more suitable for relational databases where tables are connected using one of the possible joins, e.g. natural join.

Constraints in a constrained sum of information systems can be defined internally, i.e. by a Boolean combination of attribute-value descriptors where attributes come from particular information systems, or externally, i.e. by additional attributes (different than those from the sum of information systems) that define the relationship between/among particular information systems. The information system $+_R(IS_1, \ldots, IS_k)$ can also be defined as a subsystem of $+(IS_1, \ldots, IS_k)$ by imposing on it a constraint being the characteristic function of the relation R.

3.2 Relational information granule construction

Granules in particular information systems are defined using the following language. Let $\Sigma(IS)$ denote the set of formulas, i.e. Boolean combinations of descriptors over an information system $IS = (U, A)$. Descriptors are of the form $(a \text{ in } V)$ where $a \in A$ and $V \subseteq V_a$.

Definition 4 (*Set of formulas*) The set $\Sigma(IS)$ of formulas is defined recursively by

1. $(a \text{ in } V) \in \Sigma(IS)$ for any $a \in A$ and $V \subseteq V_a$.
2. If $\alpha \in \Sigma(IS)$, then $\neg\alpha \in \Sigma(IS)$.
3. If $\alpha, \beta \in \Sigma(IS)$, then $\alpha \wedge \beta \in \Sigma(IS)$.
4. If $\alpha, \beta \in \Sigma(IS)$, then $\alpha \vee \beta \in \Sigma(IS)$.

Let $||\alpha||_{IS} \subseteq U$ denote the semantics of a formula α in IS, i.e. the set of objects that satisfy α.

Definition 5 (*Semantics of formulas*) The semantics of formulas from $\Sigma(IS)$ with respect to an information system $IS = (U, A)$ is defined recursively by

1. $||a \text{ in } V||_{IS} = \{x \in U : a(x) \in V\}$.
2. $||\neg\alpha||_{IS} = U \setminus ||\alpha||_{IS}$.
3. $||\alpha \wedge \beta||_{IS} = ||\alpha||_{IS} \cap ||\beta||_{IS}$.
4. $||\alpha \vee \beta||_{IS} = ||\alpha||_{IS} \cup ||\beta||_{IS}$.

Definition 6 (*Granules in information system, sum of information systems, and constrained sum of information systems*)

1. A granule in IS constructed over a formula $\alpha \in \Sigma(IS)$ is defined by $(\alpha, ||\alpha||_{IS})$.
2. A granule in $+(IS_1, IS_2)$ constructed over formulas $\alpha \in \Sigma(IS_1)$ and $\beta \in \Sigma(IS_2)$ is defined by $(\alpha \wedge \beta, ||\alpha||_{IS_1} \times ||\beta||_{IS_2})$.
3. A granule in $+_R(IS_1, IS_2)$ constructed over formulas $\alpha \in \Sigma(IS_1)$ and $\beta \in \Sigma(IS_2)$ is defined by $(\alpha \wedge \beta, ||\alpha||_{IS_1} \times ||\beta||_{IS_2} \cap R)$.

Granules in $+(IS_1, \ldots, IS_k)$ and $+_R(IS_1, \ldots, IS_k)$ can be defined analogously to those from $+(IS_1, IS_2)$ and $+_R(IS_1, IS_2)$, respectively.

Figure 1 shows two granules constructed based on some formulas α and β (points denoted by '+' and 'x' with ellipses represent the meaning of the granules) and one granule $\alpha \wedge \beta$ formed based on them (points denoted by '*' are understood as the join of points '+' and 'x' by the $\times$ operation).

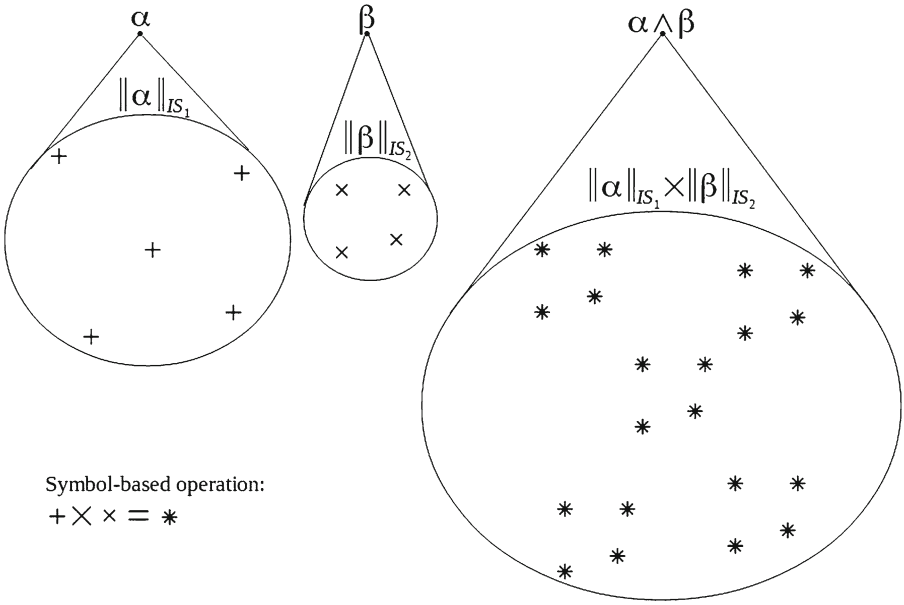

Fig. 1 Construction of granules in $+(IS_1, IS_2)$

3.3 Relational pattern construction

Patterns in $+(IS_1, IS_2)$ can be constructed by modeling the constraint R. Such a constraint forms a subsystem of $+(IS_1, IS_2)$ that consists of only objects that satisfy the pattern defined by R.

Let $\alpha \in \Sigma(+(IS_1, IS_2))$ denote a constraint defined externally, i.e. using attributes from outside IS_1 and IS_2. Such a constraint can show a relation between objects of two universes. For example if one universe includes characteristics of triangles and the other—circles, then pairs of triangle and circle can be limited by the constraint to those who hold the condition: triangle is inscribed in circle.

Definition 7 (*Pattern in sum of information systems*) Let $\alpha = \alpha_1 \wedge \cdots \wedge \alpha_{n_1} \in \Sigma(IS_1)$, $\beta = \beta_1 \wedge \cdots \wedge \beta_{n_2} \in \Sigma(IS_2)$ and $\gamma = \gamma_1 \wedge \cdots \wedge \gamma_{n_3} \in \Sigma(+(IS_1, IS_2))$ where $n_1, n_2, n_3 \geq 0$. A pattern in $+(IS_1, IS_2)$ is defined by the granule $(\alpha \wedge \beta \wedge \gamma, ||\alpha||_{IS_1} \times ||\beta||_{IS_2} \cap ||\gamma||_{+(IS_1,IS_2)})$.

If n_1, n_2 or n_3 equals to 0, it means that no formula from, respectively, IS_1, IS_2 or $+(IS_1, IS_2)$ is used in the construction of a pattern. For example, α, β, and γ impose constraints on the universe of triangles (IS_1), circles (IS_2), and pairs of both ($+(IS_1, IS_2)$), respectively.

Patterns in $+_R(IS_1, IS_2)$ can be constructed in an analogous way, except that the system $+(IS_1, IS_2)$ is initially filtered by the proper constraint R.

Patterns in $+(IS_1, \ldots, IS_k)$ and $+_R(IS_1, \ldots, IS_k)$ can be defined analogously to those from $+(IS_1, IS_2)$ and $+_R(IS_1, IS_2)$, respectively.

3.4 Illustrative example

The following running examples illustrate the notions introduced in this section.

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

Id	Age	Gender	Income	Class
Customer				
1	30	Male	1500	Yes
2	33	Female	2500	Yes
3	30	Female	1800	No
4	30	Female	1800	Yes
5	26	Female	2500	Yes
6	29	Male	3000	Yes
7	30	Male	1800	No

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

Id	Cust_id	Prod_id	Amount	Date
Purchase				
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

1. Relational data representation.
   The sum of information systems is $+(IS_1, IS_2)$, where $IS_1 = (U_1, A_1)$ and $IS_2 = (U_2, A_2)$ are constructed based on relations *customer* and *product*, respectively. Namely, $U_1 = \{x_i : i \in V_{customer.id}\}$, $A_1 = \{customer.age, customer.gender, customer.income, customer.class\}$, $U_2 = \{x_i : i \in V_{product.id}\}$, $A_2 = \{product.name, product.price\}$.
   The constrained sum of information systems is $+_R(IS_1, IS_2)$, where $R \subset U_1 \times U_2$ is defined by the condition: *A customer made a purchase*. In fact, R is defined by the *purchase* table, i.e. $R = \pi_{cust\_id,prod\_id}(purchase)$.[1]
   Comment: Due to the limitation of the R relation, only a part of the purchase table is used in the constrained sum of information systems. To take into account the remaining data from this table, one can sum three information systems, each corresponding to one table, and define the relation as follows $\pi_{customer.id,purchase.id,product.id}(customer \bowtie purchase \bowtie product)$.[2] *Such a solution enables to reflect the data and relationships from the original database, but the constructed system is redundant compared with the database.*

[1] $\pi_A(\bullet)$ is understood as a projection over the attributes from A.

[2] $rel_1 \bowtie rel_2$ is understood as the natural join of relations rel_1 and rel_2.

2. Relational information granule construction.
   Consider formulas $\alpha = (age \text{ in } [20, 30]) \wedge (gender \text{ in } \{female\}) \in \Sigma(IS_1)$ and $\beta = (price \text{ in } [4.00, 12.00])$. Information granules in $+(IS_1, IS_2)$ and $+_R(IS_1, IS_2)$ constructed based on α and β are $(\alpha \wedge \beta, \{3, 4, 5\} \times \{4, 5, 6\})$ and $(\alpha \wedge \beta, \{(4, 6), (5, 5)\})$,[3] respectively.
   Comment: In practice, when the cardinality of R is relatively small, the universe of $+(IS_1, IS_2)$ can be filtered by R before the computation of granules in particular universes of IS_1 and IS_2.
3. Relational pattern construction.
   Consider additional formula $\gamma = (income \text{ in } (0, 500 * price)) \in \Sigma(+(IS_1, IS_2))$ to construct the pattern to show women aged between 20 and 30 and the products priced between 4.00 and 12.00 if the women are able, taking their payments into account, to buy up to 500 pieces of such products. The pattern $\alpha \wedge \beta \wedge \gamma$ in $+(IS_1, IS_2)$ and $+_R(IS_1, IS_2)$ are $(\alpha \wedge \beta \wedge \gamma, \{(3, 4), (3, 5), (3, 6), (4, 4), (4, 5), (4, 6), (5, 5), (5, 6)\})$ and $(\alpha \wedge \beta \wedge \gamma, \{(4, 6), (5, 5)\})$, respectively. The pattern in $+_R(IS_1, IS_2)$ limits the results to pairs $(woman, product)$ such that $woman$ bought $product$ at least once.
   Comment: Formulas defined in $\Sigma(+(IS_1, IS_2))$ should be used to express relationships between descriptive attributes only, since relationships between key attributes are encoded in the R relation.[4]

Example 2 The framework is analyzed over a real-life and real-time financial data-base too estimate its suitability for real problem solving. The database can be used for defining the credibility of bank clients. The characteristic of a client can be constructed based on them bank history, e.g. transactions, the loans already granted, the credit cards issued (Fig. 2).

The below analyzes the data representation and information granule construction. The pattern construction is omitted (it is a direct extension of information granule construction) and left to the reader.

1. Relational data representation.
   Each table, except $Disposition$, can be represented using an information system that includes the descriptive attributes from the table, objects are identified by the primary key attribute, and a relationship defined in the table by a foreign key attribute is expressed using a constraint. For example the $Client$ table is defined by the information system $IS_{Client} = (U_{Client}, A_{Client})$, where $U_{Client} = \{x_i : i \in V_{client-id}\}$, $A_{Client} = \{birth{-}date, gender\}$, and the constraint R_{Client} specified as a function $R_{Client} : U_{Client} \mapsto U_{District}$, which assigns exactly one district for each client. The $Disposition$ table, which in fact is needed to join tables $Client$, $Account$, and $Card$, can be replaced with the $R_{Disposition}$ function defined as $R_{Disposition} : U_{Card} \mapsto U_{Client} \times U_{Account}$, which assigns exactly one pair of a client and an account for each card.
2. Relational information granule construction.
   The constrained sum of information systems enables to define, based on descriptive attributes, a granule for each its component separately as well as for any combination of the components. For example, we can consider the following elementary granules: $(\alpha, ||\alpha||_{IS_{Client}})$, $(\beta, ||\beta||_{IS_{Account}})$, $(\alpha \wedge \beta, ||\alpha||_{IS_{Client}} \times ||\beta||_{IS_{Account}})$ where

[3] To simplify the notations, objects from universes are refereed to by their identifiers only.

[4] A key attribute is understood as any attribute (usually primary or foreign key attribute) by which one system can be joined with another one or with itself. A descriptive attribute is any attribute shows features of object of a give systems.

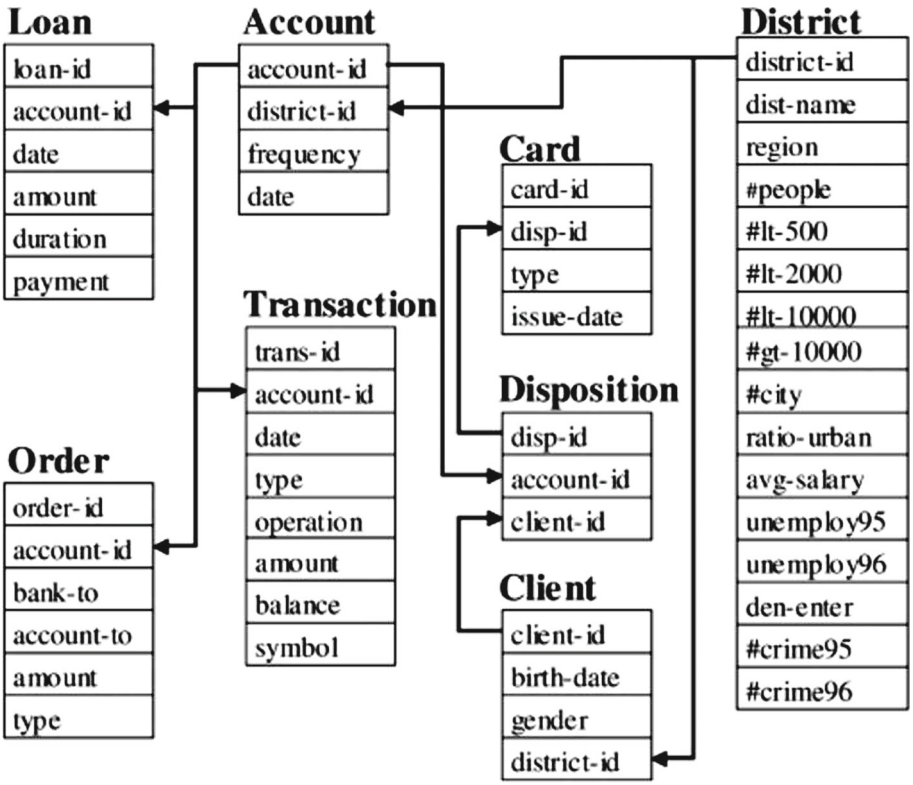

Fig. 2 The financial database (Yu et al. 2006). An arrow denotes one-to-many relationship

$\alpha = (gender, male)$ and $\beta = (frequency, month)$. The construction of more advanced granules requires a redefinition. For example to find clients who have a given type of the card it is needed to filter the $R_{District}$ function.
Comment: The above shows that the construction of a sum of information systems for a data mining task may be a dynamic process and the final form of the system may depend on not only the data but also the problem to be solved.

4 Granular association rule approach

The approach described in this section is dedicated to discovering association rules from complex data (Min et al. 2012).

The approach can be summarized as follows. A relation is represented by an information system (Definition 1). The universe of the system includes tuples of the relation, whereas the attribute set consists of the names of relation's attributes. Two relations are joined using a many-to-many entity-relationship system (Definition 9). The system is a tuple consisting of the universes and attribute sets of both information systems and a binary relation over both universes. The binary relation makes it possible to express the connection between two tables of a relational database using one of the joins, e.g. natural join, or using an additional table, i.e. the join table. The system can be generalized to more than two relations.

Granules in a many-to-many entity-relationship system are constructed based on so-called representations, which correspond to Boolean combinations of descriptors over two information systems (Definition 11). An atomic representation is the expression of the form $(a : a(x))$, where a is an attribute, and $a(x)$ is one of its possible values. More advanced representations are constructed using the conjunction operator (Definition 10). A granule in a many-to-many entity-relationship system is a triple of the name of the granule, its representation and the meaning, which is expressed as the equivalence class that contains objects belonging to the granule (Definition 8).

Two granules, each of which is seen as a basic pattern, and the binary relation are used to construct a granular association rule (Definition 12). Four types of this rules are defined (Definition 13). The type of a rule is determined by the match of its granules (partial or complete match).

The details of the approach are given in the following subsections.

4.1 Relational data representation

A structure for storing relational data is constructed based on an information system (see Sect. 3). To construct granules the universe of the system is partitioned according to the values of selected attributes.

Definition 8 1. An equivalence relation $E_{A'}$ on the universe U of an information system $IS = (U, A)$, where $A' \subseteq A$, is defined by

$$E_{A'} = \{(x, y) \in U \times U : \forall_{a \in A'} a(x) = a(y)\}.$$

2. An equivalence class that contains an object $x \in U$ is defined by

$$E_{A'}(x) = \{y \in U : (x, y) \in E_{A'}\}.$$

Relational data is stored in a many-to-many entity-relationship system.

Definition 9 A many-to-many entity-relationship system is defined by $ES = (U, A, V, B, R)$ where (U, A) and (V, B) are information systems, and $R \subseteq U \times V$ is a binary relation.

The above system enables to represent data of a relational database consisting of three tables. The information systems correspond to tables that are in many-to-many relation, whereas the binary relation can store data from the joining table that is limited to two foreign keys referring to the remaining tables.

4.2 Relational information granule construction

A granule in a many-to-many entity-relationship system is constructed based on granules from particular information systems.

Definition 10 (*Granule in information system*) Given an information system $IS = (U, A)$, a subset $A' \subseteq A$, and an object $x \in U$.

1. A granule in IS is a triple of the form $G = (g, i(g), e(g))$ where $g = (A', x)$, $i(g) = \bigwedge_{a \in A'}(a : a(x))$, and $e(g) = E_{A'}(x)$ are, respectively, the name, the representation, and the meaning of G.
2. The support of G is $supp(G) = \frac{|e(g)|}{|U|}$.

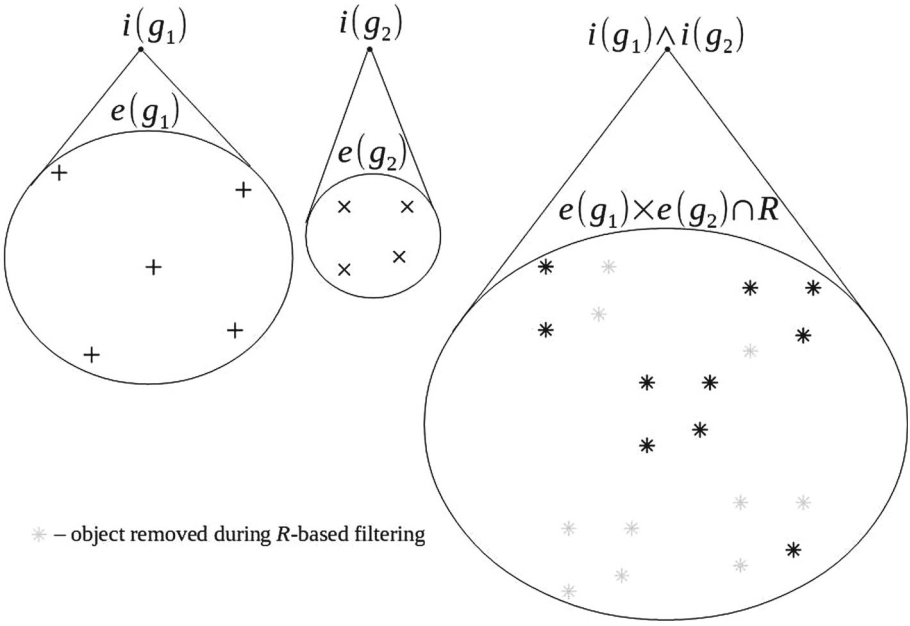

Fig. 3 Construction of granules in $ES = (U, A, V, B, R)$

Definition 11 (*Granule in many-to-many entity-relationship system*) Given a many-to-many entity-relationship system $ES = (U, A, V, B, R)$ and subsets $A' \subseteq A$ and $B' \subseteq B$. A granule in ES can be represented by a triple $(G_1, G_2, R_{1,2})$, where $G_1 = (g_1, i(g_1), e(g_1))$ and $G_2 = (g_2, i(g_2), e(g_2))$, defined as follows

1. G_1: $g_1 = (A', x)$, $i(g_1) = \bigwedge_{a \in A'}(a : a(x))$, $e(g_1) = E_{A'}(x)$;
2. G_2: $g_2 = (B', y)$, $i(g_2) = \bigwedge_{b \in B'}(b : b(y))$, $e(g_2) = E_{B'}(y)$;
3. $R_{1,2} = e(g_1) \times e(g_2) \cap R$.

Figure 3, analogously to the first one, shows the construction of granules based on representations $i(g_1)$ and $i(g_2)$ and the combination of them $i(g_1) \wedge i(g_2)$. The meaning of a combined granule is computed in two stages: joining objects from the meaning of granules $i(g_1)$ and $i(g_2)$; filtering the set of joined objects according to the give binary relation R.

4.3 Relational pattern construction

The framework provides one type of patterns, i.e. association rule.

Definition 12 (*Association rule*) Let $(G_1, G_2, R_{1,2})$ be a granule in a many-to-many entity-relationship system $ES = (U, A, V, B, R)$.

1. A granular association rule in ES is defined by the granule

$$GR = (g, i(g), e(g))$$

   where $g = (g_1, g_2)$, $i(g) = i(g_1) \Rightarrow i(g_2)$, and $e(g) = R_{1,2}$.
2. The source support is $ssupp(GR) = \frac{|e(g_1)|}{|U|}$.

3. The target support is $tsupp(GR) = \frac{|e(g_2)|}{|V|}$.
4. The support is $supp(GR) = \frac{|e(g)|}{|U \times V|}$.
5. The confidence is $conf(GR) = \frac{|e(g)|}{|e(g_1) \times e(g_2)|}$.

Four types are defined for granular association rules. Let $R(x) = \{y \in V : (x, y) \in R\}$.

Definition 13 (*Four types of granular association rule*) A granular association rule GR in $ES = (U, A, V, B, R)$ is called:

1. *Complete match rule* if and only if $e(g_1) \times e(g_2) \subseteq R$.
   - The support of GR is $supp_c(GR) = ssupp(GR)$.
   - The confidence of GR is $conf_c(GR) = conf(GR) = 1$.
2. *Left-hand side partial match rule* if and only if $\exists_{x \in e(g_1)} e(g_2) \subseteq R(x)$.[5]
   - The support of GR is $supp_{lp}(GR) = \frac{|\{x \in e(g_1) : e(g_2) \subseteq R(x)\}|}{|U|}$.
   - The confidence of GR is $conf_{lp}(GR) = \frac{|\{x \in e(G_1) : e(g_2) \subseteq R(x)\}|}{|e(g_1)|}$.
3. *Right-hand side partial match rule* if and only if $\exists_{x \in e(g_1)} e(g_2) \cap R(x) \neq \emptyset$.
   - The support of GR is $supp_{rp}(GR) = ssupp(GR)$.
   - The confidence of GR is $conf_{lp}(GR) = min\{\frac{|e(g_2) \cap R(x)|}{e(g_2)} : x \in e(g_1)\}$.
4. *Partial match rule* if and only if $e(g_1) \times e(g_2) \cap R \neq \emptyset$.
   - The support of GR with respect to a target confidence threshold $tc \in (0, 1]$ is $supp(GR, tc) = \frac{|\{x \in e(g_1) : \frac{|R(x) \cap e(g_2)|}{|e(g_2)|} \geq tc\}|}{|U|}$.
   - The source confidence of GR with respect to tc is $sconf(GR, tc) = \frac{|\{x \in e(g_1) : \frac{|R(x) \cap e(g_2)|}{|e(g_2)|} \geq tc\}|}{|e(g_1)|}$.

A complete rule shows that all objects that satisfy the left-hand side are associated with all objects that satisfy the right-hand side. A left-hand side partial match rule shows that a part of objects that satisfy the left-hand side are associated with all objects that satisfy the right-hand side. A right-hand side partial match rule shows that all objects that satisfy the left-hand side are associated with a part of objects that satisfy the right-hand side. A partial match rule shows that a part of objects that satisfy the left-hand side are associated with a part of objects that satisfy the right-hand side.

For partial match rules additional measure, called target confidence, is defined. Let sc be the source confidence threshold and let K be integer such that

$$\begin{aligned} |\{x \in e(g_1) : |R(X) \cap e(g_2)| \geq K + 1\}| \\ < sc|e(g_1)| \\ \leq |\{x \in e(g_1) : |R(X) \cap e(g_2)| \geq K\}| \end{aligned}$$

The target confidence of GR with respect to tc is $tconf(GR, sc) = \frac{K}{|e(g_2)|}$.

Applying the above calculations we can find the biggest natural number K such that each of $sc \cdot 100\%$ of objects that satisfies the left-hand side is associated with at least K objects that satisfy the right-hand side.

[5] The condition can be interpreted in such a way that at least one object that satisfies the left-hand side is associated with all objects that satisfy the right-hand side, e.g. at least one customer of age 30 buys all kinds of alcohol.

4.4 Illustrative example

The following example illustrates the notions introduced in this section.

Example 3 Consider the grocery store database from Example 1.

1. Relational data representation.
   The many-to-many entity-relationship system is $ES = (U, A, V, B, R)$ where $U = \{x_i : i \in V_{customer.id}\}$, $A = customer.A$, $V = \{x_i : i \in V_{product.id}\}$, $B = product.A$, and $R = \pi_{cust\_id, prid\_id}(purchase)$.
   Comment: Due to the limitation of the R relation (binary relation), only a part of the purchase table can be used to build the many-to-many entity-relationship system.
2. Relational information granule construction.
   Consider granules $G_1 = (g_1, i(g_1), e(g_1))$ and $G_2 = (g_1, i(g_1), e(g_1))$ defined as follows $g_1 = (\{age, gender\}, 3)$, $i(g_1) = (age : [20, 30]) \wedge (gender : female)$, $e(g_1) = \{3, 4, 5\}$ and $g_2 = (price, 4)$, $i(g_2) = (price : [4.00, 12.00])$, $e(g_2) = \{4, 5, 6\}$.[6]
   The information granule in ES constructed based on G_1 and G_2 is $(G_1, G_2, R_{1,2})$ where $R_{1,2} = e(g_1) \times e(g_2) \cap R = \{(4, 6), (5, 5)\}$.
   Comment: In practice, when the cardinality of R is relatively small, the condition $i(g_j)$ can be checked over $R_j = \pi_j(R_{1,2})$ $(j = 1, 2)$ only.
3. Relational pattern construction.
   Consider the association rule $GR = G_1 \Rightarrow G_2 = ((g_1, g_2), i(g_1) \Rightarrow i(g_2), R_{1,2})$ with $ssup(GR) = \frac{3}{7}$ and $tsup(GR) = \frac{1}{2}$. The rule can be interpreted in the following way: if a person is a woman aged between 20 and 30, then she buys products priced between 4.00 and 12.00 where 43% of persons are women aged between 20 and 30, and 50% of products are priced between 4.00 and 12.00.
   The association rule is a partial match one, since $R_{1,2} \neq \emptyset$. Consider then the target confidence threshold $tc = 0.1$. The support and confidence of GR are $supp(GR, tc) = \frac{|\{4\}|}{|U|} = 0.14$ and $sconf(GR, tc) = \frac{|\{4\}|}{|\{3,4,5\}|} = 0.33$.
   Let $sc = 0.33$ be the source confidence threshold. For $K = 1$ we obtain $|\emptyset| < 0.33|\{3, 4, 5\}| \geq |\{4, 5\}|$. Hence, target confidence of GR with respect to tc is $tconf(GR, sc) = \frac{1}{|\{3,4,5\}|} = 0.33$. The rule with $supp(GR, tc) = 0.33$ and $tconf(GR, sc) = 0.33$ can be read as: 33% of women aged between 20 and 30 buy at least 33% of products priced between 4.00 and 12.00.
   Comment: Since the many-to-many entity-relationship system is very specialized, then its any double universe granule is a potential association rule.

Example 4 Analyze the suitability of the framework using the financial database from Example 2.

1. Relational data representation.
   The data can be represented using a set of many-to-many entity-relationship systems. For example, for tables $Client$ and $Disposition$ the following systems can be used: $ES = (U_{Client}, A_{Client}, U_{Disposition}, A_{Disposition}, R)$ where U_{Client} and A_{Client} are defined as in Example 2, $U_{Disposition} = \{x_i : i \in V_{disp-id}\}$, $A_{Disposition} = \emptyset$, and R is a relation on $U_{Client} \times U_{Disposition}$. Systems for joining each of tables $Account$

[6] The 3 value in the name of g_1, i.e. $(\{age, gender\}, 3)$, may be replaced by any remaining one from $e(g_1)$. Analogously for g_2.

and $Card$ with $Disposition$ are constructed in an analogous way. In spite of the fact that $A_{Disposition}$ is empty, $U_{Disposition}$ is needed to reflect relationships occurring in the database.

2. Relational information granule construction.
   Each component of a given system with a non-empty attribute set can be used to define granules. For example, let $(G_1, G_2, R_{1,2})$ be a granule defined in ES_1 such that $G_1 = (g_1, i(g_1), e(g_1))$, $G_2 = (g_2, i(g_2), e(g_2))$, $g_1 = (\{gender\}, x)$, $i(g_1) = (gender : female)$, $g_2 = (\emptyset, y)$, and $i(g_2) = null$.
   To construct granules over the set of all systems we need to introduce additional operations enabling a proper communication among the systems. Given many-to-many entity-relationship systems ES and ES' such that there exist a system (U, A) that is a subsystem of both ES and ES'. Granules $(G_1, G_2, R_{1,2})$ and $(G'_1, G'_2, R'_{1,2})$ defined, respectively, in ES and ES' can be joined as follows $(G_1, G_2, G'_1, G'_2, R_{1,2} \bowtie R'_{1,2})$.
   For example, consider systems ES (defined as previously) and $ES' = (U_{Account}, A_{Account}, U_{Disposition}, A_{Disposition}, R')$ where R' is a relation on $U_{Account} \times U_{Disposition}$. Consider also granules $(G_1, G_2, R_{1,2})$ (defined as previously) and $(G'_1, G'_2, R'_{1,2})$ defined in ES' as follows: $G'_1 = (g'_1, i(g'_1), e(g'_1))$, $G'_2 = G_2$,$g'_1 = (\{frequency\}, x)$, $i(g'_1) = (frequency : month)$. We obtain a granule $(G_1, G_2, G'_1, G'_2, R_{1,2} \bowtie R'_{1,2})$ where the relation formed by joining the binary relations has the following schema $R_{1,2} \bowtie R'_{1,2}(gender, disp{-}id, frequency)$.

5 Generalized related set based approach

This section introduces a framework that uses relational information granules constructed based on the notion of generalized related set (Hońko 2013a, b).

The approach can be summarized as follows. Each tuple of a relation is represented by a relational object, i.e. the tuple together with the relation name (Definition 14). An extension of the standard information system is used to store relational data (Definition 15). The universe consists of relational objects of all relations of a database. The attribute set includes attributes of all the relations, preserving the distinguishability of attributes having the same name but coming from different information systems.

Granules in an extended information system are constructed based on target objects and the background objects related to them (Definitions 16 and 17). A target object with its related background objects are generalized: constants that occur in objects are replaced by variables (Definitions 18 and 19). Such a generalization is an abstract representation of a target object and shows the relationship between the target object and its related objects. A granule is defined by a pair of an abstract representation of a target object and the semantics, i.e. the set of all target objects that possess properties defined in the representation (Definition 20). Such a granule corresponds to a basic pattern (Definition 21). More advanced patterns are formed according to the standard principle of building data mining patterns (e.g. association or classification rules).

The details of the approach are given in the following subsections.

5.1 Relational data representation

To consider objects apart from the tables they belong to, the notion of relational object is used.

Definition 14 (*Relational object*) Given a database relation with the schema $R(a_1, a_2, \ldots, a_n)$. An expression of the form $R(v_1, v_2, \ldots, v_n)$ is an object of R if and only if $(v_1, v_2, \ldots, v_n)$ is a tuple of R.

A relational database is represented by a complex information system that is constructed based on a standard information system (see Sect. 3).

Let

- D_T and D_B denote, respectively, the sets of target and background relations of database D (i.e. a set of all relations);
- $U_{D_T} = \bigcup_{R \in D_T} R$ and $U_{D_B} = \bigcup_{R \in D_B} R$ be, respectively, the set of all target and background objects of database D
- $A_{D_T} = \bigcup_{R \in D_T} A_R$[7] and $A_{D_B} = \bigcup_{R \in D_B} A_R$ be, respectively, the set of all attributes of the target and background relations of database D.

The following representation of a relational database is introduced.

Definition 15 (*Information system for a relational database*) A relational database $D = T \cup B$ is represented by an information system $IS_D = (U_D, A_D)$, where

1. $U_D = U_{D_T} \cup U_{D_B}$ is a non-empty finite set of objects, called the universe,
2. $A_D = A_{D_T} \cup A_{D_B}$ is a non-empty finite set of attributes.

The information system defined above includes objects together with the names of tables they belong to. Information on table joins is not directly stored in the system. They can be reconstructed based on metadata on primary and foreign keys.

5.2 Relational information granule construction

In this approach, essential information acquired from the relational data are descriptions of target objects that are constructed based on the notion of related set (Hońko 2010, 2013a).

Definition 16 (*Related object and set*)

1. Object o is related to object o' if and only if there exists a key attribute joining o with o'.[8]
2. A related set of a target object o, denoted by $rlt(o)$, is a set of background objects directly or indirectly related to the target object.

In this approach, the key attribute is, in general, understood as an important attribute for joining tables. It is usually a primary or foreign key. However, in some cases, it can also be another attribute by which one table can be joined with another table or with itself.

A target object's description is expressed by a set of background objects joined with the target object. For a given target object one can usually obtain more than one description, each of which describes the object with different precision. The objective is to choose an appropriate description of the target object with respect to a given data mining task. The precision of the target object's description (i.e., the related set) can be tuned by its depth level. To define a related set of a given depth level, Definition 15 is generalized.

[7] A_R denotes here the set of all attributes of relation R.

[8] The tables the objects belong to are not assumed to be different.

Definition 17 (*n-related object and set*)

1. Object o_0 is n-related to object o_n if and only if there exist $n-1$ objects such that o_i is related to o_{i+1}, where $n > 0$ and $0 \leq i \leq n-1$.
2. The n-th depth level related set of a target object o, denoted by $rlt^n(o)$, is a set of background objects, each of which are m-related to object o and $m \leq n$.

A related set of a given target object can be viewed as its specific description. In order to derive relational patterns, the target object's description is generalized. To obtain a general description of a target object itself and its related set, they are both generalized.

Definition 18 (*Generalized object*) Let $o = R(v_1, v_2, \ldots, v_n)$ be an object where $(v_1, v_2, \ldots, v_n)$ is a tuple of a relation R. A generalized object o, denoted by o_{gen}, is defined by

$$o_{gen} = o\sigma$$

where $\sigma = \{v_{i_1}/t_1, v_{i_2}/t_2, \ldots v_{i_m}/t_m\}$ is a substitution such that $v_{i_j} \in \{v_1, v_2, \ldots, v_n\}$ $(j = 1, \ldots, m, m \leq n)$, and t_i is either a variable, a list of constants, or symbol "_" if the component is not important for the consideration.[9]

Definition 19 (*Generalized related set*) Let $rlt(o) = \{o_1, \ldots, o_n\}$ be the related set of a target object o. A generalized related set of a target object o, denoted by $rlt_{gen}(o)$, is defined by

$$rlt_{gen}(o) = rlt(o)\sigma = \{o_1\sigma_1, \ldots, o_n\sigma_n\}$$

where σ is a substitution, there exists $\sigma_0 \subseteq \sigma$ such that $o_{gen} = o\sigma_0$, and $\sigma_i \subseteq \sigma$ $(i = 1, \ldots, n)$.

A generalized n-related set is defined in an analogous way.

Related sets can be generalized in a variety of ways (for more details, see Hońko 2010). A method for generalization can be developed taking into consideration language bias.

Definition 20 A granule in an information system $IS_D = (U_D, A_D)$ is a pair of the form $(g, SEM_{IS_D}(g))$ where

1. $g = (o_{gen}, rlt_{gen}(o))$;
2. $SEM_{IS_D}(g)) = (SEM_{IS_D}(o_{gen}), SEM_{IS_D}(rlt_{gen}(o)))$;
3. $SEM_{IS_D}(o_{gen})$ is the set of target objects that satisfy the descriptor;
4. $SEM_{IS_D}(rlt_{gen}(o))$, is the set of target objects for each of which there exists a substitution such that each descriptor under the substitution is satisfied.

The information granules as defined above can be viewed as an abstract representation of relational data. The accuracy level of the representation can easily be changed by taking another depth level of the related sets.

Figure 4 shows how the meaning of a granule of the form $(o_{gen}, rlt^i_{gen}(o))$ changes depending on the depth level i. Note that $rlt^0_{gen} = \emptyset$.

[9] The notion of substitution is borrowed from ILP where it is used to constructed Horn clauses based on target and background examples. For more details, see Džeroski and Lavrač (2001b).

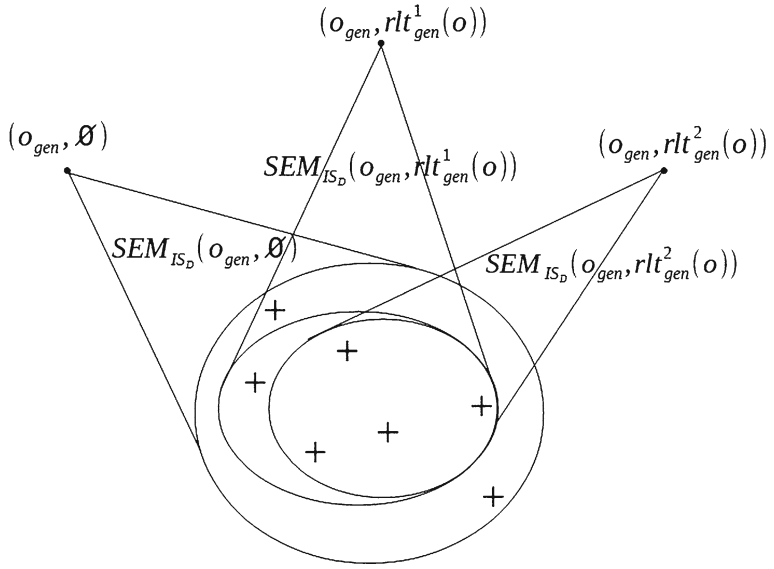

Fig. 4 Construction of granules in IS_D

5.3 Relational pattern construction

Granular patterns introduced in this subsection correspond to standard relational patterns, i.e relational frequent patterns, relational association and classification rules.

Definition 21 Given information system $IS_D = (U_D, A_D)$. Relational patterns are defined using information granules as follows.

1. A relational (frequent) pattern α in IS_D is represented by the granule $(g, SEM_{IS_D}(g))$ where $g = (o_{gen}, rlt_{gen}(o))$ and $\alpha \Leftrightarrow \bigwedge_{x \in \{o_{gen}\} \cup rlt_{gen}(o)} x$. The pattern's frequency can be calculated by $freq_{IS_D}(\alpha) = \frac{|SEM_{IS_D}(rlt_{gen}(o))|}{|SEM_{IS_D}(o_{gen})|}$.
2. A relational association rule $\alpha \rightarrow \beta$ in IS_D is represented by the granule (α, β), where α and β are defined, respectively, by $(o_{gen}, rlt'_{gen}(o))$ and $(o_{gen}, rlt_{gen}(o))$ such that $SEM_{IS_D}(rlt_{gen}(o)) \subseteq SEM_{IS_D}(rlt'_{gen}(o))$.[10] The meaning of the granule is $SEM_{IS_D}((\alpha, \beta)) = (SEM_{IS_D}(\alpha), SEM_{IS_D}(\beta))$.
   Since any association rule is constructed based on patterns that are discovered over the same relation (i.e., both patterns are checked to be satisfied for objects of the same relation), the meaning of the granule can be written in a simpler form, that is, $SEM_{IS_D}((\alpha, \beta)) = (SEM_{IS_D}(o_{gen}), SEM_{IS_D}(rlt'_{gen}(o)), SEM_{IS_D}(rlt_{gen}(o)))$. The rule's frequency and confidence can be calculated by $freq_{IS_D}(\alpha \rightarrow \beta) = freq_{IS_D}(\beta)$ and $conf_{IS_D}(\alpha \rightarrow \beta) = \frac{freq_{IS_D}(\beta)}{freq_{IS_D}(\alpha)}$, respectively.
3. A relational classification rule[11] $\alpha \leftarrow \beta$ in IS_D is represented by granule (α, β), where α and β correspond to o_{gen} and $rlt_{gen}(o)$, respectively. The meaning of the granule is

[10] Here, $rlt'(o)$ means a different related set than $rlt(o)$.

[11] The reversed direction in rule notation is adapted from relational language (see e.g. Džeroski and Lavrač 2001b).

$SIM_{IS_D}((\alpha,\beta)) = (SEM_{IS_D}(\alpha), SEM_{IS_D}(\beta))$. The rule's accuracy and coverage can be calculated by $acc_{IS_D}(\alpha \leftarrow \beta) = \frac{|SEM_{IS_D}(o_{gen}) \cap SEM_{IS_D}(rlt_{gen}(o))|}{|SEM_{IS_D}(rlt_{gen}(o))|}$ and $cov_{IS_D}(\alpha \leftarrow \beta) = \frac{|SEM_{IS_D}(o_{gen}) \cap SEM_{IS_D}(rlt_{gen}(o))|}{|SEM_{IS_D}(o_{gen})|}$, respectively.

5.4 Illustrative example

The notions introduced in this section are illustrated by the following example.

Example 5 Consider the grocery store database from Example 1.

1. Relational data representation.
   The database can be represented by the information system $IS_D = (U_D, A_D)$, where $U_D = U_{D_T} \cup U_{D_B}$, $A_D = A_{D_T} \cup A_{D_B}$ are defined as follows:
   $U_{D_T} = \{c_1, \ldots, c_7\}$,[12] $U_{D_B} = \{p_1, \ldots, p_8, p'_1, \ldots, p'_6\}$,
   $A_{D_T} = \{id, age, gender, income, class\}$, $A_{D_B} = \{p'.id, name, price, p.id, cust\_id, prod\_id, amount, date\}$.
   Comment: From the logical viewpoint the construction of the information system varies depending on the table to be specified as the target one. In practice the information system can be always the same, e. g. $U_D = \{c_1, \ldots, c_7, p_1, \ldots, p_8, p'_1, \ldots, p'_6\}$, *and the target table can be specified at the stage of granule construction.*
2. Relational information granule construction.
   In order to define conditions on attributes age and $gender$ (as in Examples 1 and 3) and to store them in related sets, one can use a projection on a copy of table $customer$ as a background table, i.e. $customer' = \pi_{id,age,gender}(customer)$. Let also c'_i be the i-the object of relation $customer'$.
   Consider the target object $o = c_5$ with its related sets $rlt^1(o) = \{c'_5, p_7\}$ and $rlt^2(o) = \{c'_5, p_7, p'_5\}$. Let $\sigma = \{c[1]/A, c[2]/\_, c[3]/\_, c[4]/\_, c[5]/\_, c'[1]/A, c'[2]/[20, 30], p[1]/B, p[2]/A, p[3]/C, p[4]/\_, p[5]/\_, p'_1/B, p'_2/\_, p'_3/[4.00, 12.00]\}$.
   The generalization of target object o and its related sets using σ are $o_{gen} = c(A, \_, \_, \_, \_)$, $rlt^1_{gen}(o) = \{c'(A, [20, 30], female), p(B, A, C, \_, \_)\}$, and $rlt^2_{gen}(o) = \{c'(A, [20, 30], female), p(B, A, C, \_, \_), p'(C, \_, [4.00, 12.00])\}$.
   The meaning of the granules $(o_{gen}, rlt^1_{gen}(o))$ and $(o_{gen}, rlt^2_{gen}(o))$ are $SEM_{IS_D}\left((o_{gen}, rlt^1_{gen}(o))\right) = (\{1, \ldots, 7\}, \{4, 5, 6\})$ and $SEM_{IS_D}\left((o_{gen}, rlt^2_{gen}(o))\right) = (\{1, \ldots, 7\}, \{4, 5\})$.
   Comment: 1. An attribute that is not important for further computations is replaced with "_". 2. In practice a target object and its related set can be treated as one set, i.e. $\{o\} \cup rlt(o)$. *Thanks to this, a copy of the target object does not have to be added to the related set to build conditions on target attributes.*
3. Relational pattern construction.
   Given patterns $\alpha = c'(A, [20, 30], female, \_, \_) \wedge p(B, A, C, \_, \_)$ and $\beta = \alpha \wedge p'(C, \_, [4.00, 12.00])$.
   (a) Patterns α and β can be represented, respectively, by granules $(o_{gen}, rlt^1_{gen}(o))$ and $(o_{gen}, rlt^2_{gen}(o))$.

[12] For simplicity purposes, relations $customer$, $purchase$, and $product$ are abbreviated to c, p and p', respectively. The i-th object of a relation $relation$ is denoted by r_i.

The frequencies of α and β are $freq_{IS_D}(\alpha) = \frac{|SEM_{IS_D}(rlt^1_{gen}(o))|}{|SEM_{IS_D}(o_{gen})|} = 3/7$ and $freq_{IS_D}(\beta) = \frac{|SEM_{IS_D}(rlt^2_{gen}(o))|}{|SEM_{IS_D}(o_{gen})|} = 2/7$.

(b) Consider the association rule $\alpha \rightarrow \beta$. The meaning of the rule is $SEM_{IS_D}(\alpha \rightarrow \beta) = (\{1, \ldots, 7\}, \{4, 5, 6\}, \{4, 5\})$. The frequency and confidence of $\alpha \rightarrow \beta$ are $freq_{IS_D}(\alpha \rightarrow \beta) = freq_{IS_D}(\beta) = 2/7$ and $conf_{IS_D}(\alpha \rightarrow \beta) = \frac{freq_{IS_D}(\beta)}{freq_{IS_D}(\alpha)} = 2/3$.

(c) Consider also pattern $\gamma = customer(A, \_, \_, \_, yes)$. Let $\sigma' = (\sigma \setminus \{c[5]/\_\}) \cup \{c[5]/D\}$. Hence, we obtain $o_{gen'} = c(A, \_, \_, \_, D)$.
The classification rule $\gamma \leftarrow \beta$ can be represented by the granule $(o_{gen'}, rlt^2_{gen}(o))$ with the meaning $SEM_{IS_D}(\gamma \leftarrow \beta) = (\{1, 2, 4, 5, 6\}, \{4, 5\})$. The rule's accuracy and coverage are $acc_{IS_D}(\gamma \leftarrow \beta) = \frac{|SEM_{IS_D}(o_{gen'}) \cap SEM_{IS_D}(rlt^2_{gen}(o))|}{|SEM_{IS_D}(rlt^2_{gen}(o))|} = 1$ and $cov_{IS_D}(\gamma \leftarrow \beta) = \frac{|SEM_{IS_D}(o_{gen'}) \cap SEM_{IS_D}(rlt^2_{gen}(o))|}{|SEM_{IS_D}(o_{gen})|} = 2/5$.

Comment: If the generalization of the target object is most general, i.e. $SEM_{IS_D}(o_{gen}) = U_T$, then the meaning of an association rule can be written in a shorter form. For example, $SEM_{IS_D}(\alpha \rightarrow \beta) = (\{1, \ldots, 7\}, \{4, 5, 6\}, \{4, 5\})$ is shortened to $SEM_{IS_D}(\alpha \rightarrow \beta) = (\{4, 5, 6\}, \{4, 5\})$ provided that the universe cardinality is accessible during the computation of the rule quality.

Example 6 Analyze the suitability of the framework using the financial database from Example 2.

1. Relational data representation.
   Each database relation is a component of the universe of the information system. For example the part of the database that concerns tables $Client$, $Account$, and $Disposition$, assuming the $Client$ is the target table, can be represented by the system $IS_D = (U, A)$ where $U_D = U_{D_T} \cup U_{D_B} = \{Client\} \cup \{Account, Disposition\}$, $A_D = A_{D_T} \cup A_{D_B} = \{client{-}id, birth{-}date, gender\} \cup \{Account.account{-}id, frequency, date, disp{-}id, Disposition.account{-}id, Disposition.client{-}id\}$.
2. Relational information granule construction.
   A granule can be constructed based on any target object and its related set. It can also be done based on only the database structure and expert knowledge. For example one can construct a granule using a virtual generalized object $o_{gen} = Client(A, B, C)$ and its related set $rlt_{gen}(o) = \{Disposition(D, E, A), Account(E, F, G)\}$.
   Comment: Since data may change over time, before computing the meaning of a previously defined granule it is required to update target objects representation, i.e. related sets.

6 Description language based approach

This section introduces a framework that uses relational information granules constructed using description languages defined for relational data (Hońko 2014, 2015a).

The approach can be summarized as follows. A relation is represented by an information system (Definition 1). The universe of the system includes tuples of the relation, whereas the attribute set consists of the names of relation's attributes. A set of relations is represented by a compound information system (Definition 22). The universes of particular systems are

joined using the Cartesian product, the attribute sets are merged using the union operations, preserving the distinguishability of attributes having the same name but coming from different information systems.

To make it possible to express relationships between/among particular universes, the compound information system is constrained (Definition 23). Namely, tuples of the complex universe (the Cartesian product of particular universes) are filtered according to a given constraint that is defined using a generalized definition of left outer join. Such a constraint enables to impose more than one relationship for to tables.

Granules in a (compound) information system are constructed based on formulas, i.e. Boolean combinations of descriptors over particular information systems. An atomic formula is the expression of one of the forms: (a, v) (a is an attribute, and v is one of its possible values); (a, V) (V is the set of values the attribute may take); (a, a') (a and a' are attributes that join either the same table with itself or different tables) (Definition 24). More advanced formulas are constructed recursively using logical operators (Definitions 25 and 26). For each formula its semantics is expressed as the set of objects that satisfy the formula. A pair of a formula and its semantics is treated as a granule (Definition 27).

A granule that comprises a formula constructed using the conjunction operator is seen as a basic pattern (Definition 28). More advanced patterns are formed according to the standard principle of building data mining patterns (e.g. association or classification rules) (Definitions 29 and 30).

The details of the approach are given in the following subsections.

6.1 Relational data representation

Each table of a database is represented by an information system (see Sect. 3).

The compound information system corresponding to m database tables is defined as follows.

Definition 22 (*Compound information system* $IS_{(1,2,\ldots,m)}$) Let $IS_i = (U_i, A_i)$ be information systems, where $1 \leq i \leq m$ and $m > 1$ is a fixed number. A compound information system $IS_{(1,2,\ldots,m)}$ is defined by

$$IS_{(1,2,\ldots,m)} = \times(IS_1, IS_2, \ldots, IS_m) = \left(\prod_{i=1}^{m} U_i, \bigcup_{i=1}^{m} A_i\right). \tag{1}$$

To allow the connections between tables that occur in the original database or they are defined by an expert, a constrained version of the compound information system is introduced.

A constraint, denoted by $\bowtie_\Theta$, is defined by the left outer join[13] on disjunction of the formulas from Θ. The Θ set can consists of any formulas defined over the information systems joined by $\bowtie$. The use of left outer join guarantees that an object from the left universe is not removed from the database if it does not join to any object from the right universe.

The constrained compound information system corresponding to m database tables is defined as follows.

[13] A left outer join of tables T_1 and T_2 is the natural join of them expanded by those objects from T_1 that do not hold the joining condition.

Definition 23 (*Constrained compound information system* $IS^{\Theta}_{(1,2,\ldots,m)}$) A constrained compound information system $IS^{\Theta}_{(1,2,\ldots,m)}$ is defined by

$$IS^{\Theta}_{(1,2,\ldots,m)} = \bowtie_{\Theta}(IS_1, IS_2, \ldots, IS_m) = \left(U_1 \bowtie_{\Theta} U_2 \bowtie_{\Theta} \cdots \bowtie_{\Theta} U_m, \bigcup_{i=1}^{m} A_i \right). \quad (2)$$

6.2 Relational information granule construction

For each information system that corresponds to a database table a description language is defined. The language enables to define formulas that are used for constructing information granules.

Let $A = A_{des} \cup A_{key}$, where $IS = (U, A)$ is an information system and A_{des} (A_{key}) is the set of descriptive (key) attributes. The descriptive language for IS is denoted by $L_{IS} = L_{IS_{des}} \cup L_{IS_{key}}$. An atomic formula and its negation are defined in L_{IS} by their syntax and semantics.

Definition 24 (*Syntax and semantics of atomic formula* $L_{IS} = L_{IS_{des}} \cup L_{IS_{key}}$) The syntax and semantics of atomic formulas in a language L_{IS} are defined by

1. $a \in A_{des}, v \in V_a \Rightarrow (a, v) \in L_{IS_{des}}$ and $SEM_{IS_{des}}(a, v) = \{x \in U : a(x) = v\}$;
2. $a \in A_{des}, V \subseteq V_a \Rightarrow (a, V) \in L_{IS_{des}}$ and $SEM_{IS_{des}}(a, V) = \{x \in U : a(x) \in V\}$;
3. $\alpha \in L_{IS_{des}} \Rightarrow \neg\alpha \in L_{IS_{des}}$ and $SEM_{IS_{des}}(\neg\alpha) = U \setminus SEM_{IS_{des}}(\alpha)$,
4. $a, a' \in A_{key} \Rightarrow (a, a') \in L_{IS_{key}}$ and $SEM_{IS_{key}}(a, a') = \{x \in U : a(x) = a'(x)\}$;
5. $\alpha \in L_{IS_{key}} \Rightarrow \neg\alpha \in L_{IS_{key}}$ and $SEM_{IS_{key}}(\neg\alpha) = U \setminus SEM_{IS_{key}}(\alpha)$.

More advanced formulas are constructed recursively using logical operators such as conjunction and disjunction. For more details, see Hońko (2015a).

The above-defined language facilitates the construction of formulas that express not only simple features of objects (i.e. formulas with descriptors of the form $(a, v) \in L_{IS_{des}}$) but also relationships between the features (i.e. formulas with descriptors of the form $(a, a') \in L_{IS_{key}}$).

Using any granule description language L, one can define granules of the form $(\alpha, SEM(\alpha))$, where $\alpha \in L$.

A description language corresponding to two database tables is constructed as follows. Let $L_{IS_{(i,j)}} = L_{IS_{i\vee j}} \cup L_{IS_{i\wedge j}}$, where each formula of $L_{IS_{i\vee j}}$ is constructed over either IS_i or IS_j, and $L_{IS_{i\wedge j}}$ consists of formulas constructed over both IS_i and IS_j.

Definition 25 (*Syntax and semantics of atomic formula in* $L_{IS_{(i,j)}}$) The syntax and semantics of atomic formulas in a language $L_{IS_{(i,j)}}$ are defined by

1. $\alpha \in L_{IS_i} \Rightarrow \alpha \in L_{IS_{i\vee j}}$ and $SEM_{IS_{i\vee j}}(\alpha) = SEM_{IS_i}(\alpha) \times U_j$;
2. $\alpha \in L_{IS_j} \Rightarrow \alpha \in L_{IS_{i\vee j}}$ and $SEM_{IS_{i\vee j}}(\alpha) = U_i \times SEM_{IS_j}(\alpha)$;
3. $\alpha \in L_{IS_{i\vee j}} \Rightarrow \neg\alpha \in L_{IS_{i\vee j}}$ and $SEM_{IS_{i\vee j}}(\neg\alpha) = (U_i \times U_j) \setminus SEM_{IS_{i\vee j}}(\alpha)$;
4. $a \in (A_i)_{key}, a' \in (A_j)_{key} \Rightarrow (a, a') \in L_{IS_{i\wedge j}}$ and $SEM_{IS_{i\wedge j}}(a, a') = \{(x, y) \in U_i \times U_j : a(x) = a'(y)\}$;
5. $\alpha \in L_{IS_{i\wedge j}} \Rightarrow \neg\alpha \in L_{IS_{i\wedge j}}$ and $SEM_{IS_{i\wedge j}}(\neg\alpha) = (U_i \times U_j) \setminus SEM_{IS_{i\wedge j}}(\alpha)$.

The above-defined language makes it possible to construct formulas that show features of pairs of objects from different universes. Furthermore, the formulas can also show the relationship between the objects themselves (i.e. formulas with a descriptor of the form $(a, a') \in L_{IS_{i\wedge j}}$).

A description language can be extended to $L_{IS_{(m)}}$ defined for a compound information system $IS_{(m)}$.

Definition 26 (*Syntax and semantics of atomic formula in* $L_{IS_{(m)}}$) The syntax and semantics of atomic formulas in a language $L_{IS_{(m)}}$ are defined by

1. $\alpha \in L_{IS_i} \Rightarrow \alpha \in L_{IS_{(m)}}$ and $SEM_{IS_{(m)}}(\alpha) = U_1 \times \cdots \times U_{i-1} \times SEM_{IS_i}(\alpha) \times U_{i+1} \times \cdots \times U_m$;
2. $\alpha \in L_{IS_{(i,j)}} \Rightarrow \alpha \in L_{IS_{(m)}}$ and $SEM_{IS_{(m)}}(\alpha) = \{(x_1, \ldots, x_i, \ldots, x_j, \ldots, x_m) \in \prod_{k=1}^{m} U_k : (x_i, x_j) \in SEM_{IS_{(i,j)}}(\alpha)\}$;
3. $\alpha \in L_{IS_{(m)}} \Rightarrow \neg\alpha \in L_{IS_{(m)}}$ and $SEM_{IS_{(m)}}(\neg\alpha) = (U_1 \times \cdots \times U_m) \setminus SEM_{IS_{(m)}}(\alpha)$.

Since knowledge discovery is focused on selected database tables only, usually one table (i.e. the target table), the semantics of $L_{IS_{(m)}}$ is extended by the following

1. $\alpha \in L_{IS_{(m)}} \Rightarrow SEM^{\pi_i}_{IS_{(m)}}(\alpha) = \pi_{A_i}(SEM_{IS_{(m)}}(\alpha))$, where $1 \le i \le m$;
2. $\alpha \in L_{IS_{(m)}} \Rightarrow SEM^{\pi_{i_1,i_2,\ldots,i_k}}_{IS_{(m)}}(\alpha) = \pi_{A_{i_1},A_{i_2},\ldots,A_{i_k}}(SEM_{IS_{(m)}}(\alpha))$, where $1 \le i_1, i_2, \ldots, i_k \le m$ and $k < m$.

The syntax and semantics of $L_{IS^{\Theta}_{(m)}}$ are defined in the same way as in Definition 26. It is enough to replace $IS_{(i,j)}$, $IS_{(m)}$, and the $\times$ operation with $IS^{\Theta}_{(i,j)}$, $IS^{\Theta}_{(m)}$, and the $\bowtie_{\Theta}$ operation, respectively.

Definition 27 A granule in a compound information system $IS_{(m)}$ is a pair $(\alpha, SEM_{IS_{(m)}}(\alpha))$ where $\alpha \in L_{IS_{(m)}}$.

A granule in a constrained compound information system $IS_{(m)}$ is defined analogously.

Figure 5, analogously to that from Sect. 4, shows granules constructed based on formulas (a, v) and (a', v') and and its combination that is restricted by an additional formula (b, b') defining how to join two information systems.

6.3 Relational pattern construction

Granular patterns introduced in this subsection correspond to standard relational patterns. More advanced granular patterns can be found in Hońko (2015b).

Let $IS_{(m)} = \times(IS_1, IS_2, \ldots, IS_m)$ be a compound information system.

Definition 28 (*Frequent pattern*)

1. A pattern in $IS_{(m)}$ is an expression of the form $\alpha = \alpha_1 \wedge \cdots \wedge \alpha_k \in L_{IS_{(m)}}$, where $k \ge 1$.
2. The frequency of α is $freq_{IS_{(m)}}(\alpha) = \frac{|SEM_{IS_{(m)}}(\alpha)|}{|U_{(m)}|}$.[14]
3. The frequency of α with respect to IS_i $(1 \le i \le m)$ is $freq^{\pi_i}_{IS_{(m)}}(\alpha) = \frac{|SEM^{\pi_i}_{IS_{(m)}}(\alpha)|}{|U_i|}$.

Definition 29 (*Association rule*)

1. An association rule in $IS_{(m)}$ is an expression of the form $\alpha \to \beta \in L_{IS_{(m)}}$, where α and β are patterns in $IS_{(m)}$, and they have no common descriptor.

[14] $U_{(m)}$ denotes the universe of $IS_{(m)}$.

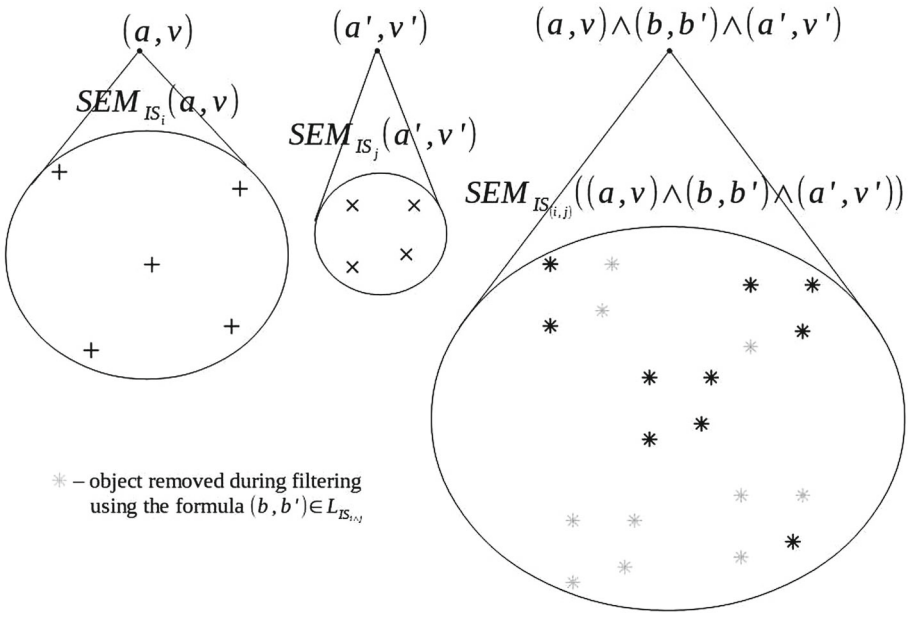

Fig. 5 Construction of granules in $IS_{(i,j)}$

2. The frequency and confidence of $\alpha \rightarrow \beta$ are $freq_{IS_{(m)}}(\alpha \rightarrow \beta) = freq_{IS_{(m)}}(\alpha \wedge \beta)$ and $conf_{IS_{(m)}}(\alpha \rightarrow \beta) = \frac{freq_{IS_{(m)}}(\alpha \wedge \beta)}{freq_{IS_{(m)}}(\alpha)}$, respectively.
3. The frequency and confidence of $\alpha \rightarrow \beta$ with respect to IS_i $(1 \leq i \leq m)$ are $freq^{\pi_i}_{IS_{(m)}}(\alpha \rightarrow \beta) = freq^{\pi_i}_{IS_{(m)}}(\alpha \wedge \beta)$ and $conf^{\pi_i}_{IS_{(m)}}(\alpha \rightarrow \beta) = \frac{freq^{\pi_i}_{IS_{(m)}}(\alpha \wedge \beta)}{freq^{\pi_i}_{IS_{(m)}}(\alpha)}$, respectively.

A classification rule is defined as a special case of an association rule.

Definition 30 (*Classification rule*)

1. A classification rule[15] in $IS_{(m)}$ is an association rule $\alpha \rightarrow \beta \in L_{IS_{(m)}}$ such that β is the decision descriptor.
2. The accuracy and coverage of $\alpha \rightarrow \beta$ are $acc_{IS_{(m)}}(\alpha \rightarrow \beta) = \frac{freq_{IS_{(m)}}(\alpha \wedge \beta)}{freq_{IS_{(m)}}(\alpha)}$ and $cov_{IS_{(m)}}(\alpha \rightarrow \beta) = \frac{freq_{IS_{(m)}}(\alpha \wedge \beta)}{freq_{IS_{(m)}}(\beta)}$, respectively.
3. The accuracy and coverage of $\alpha \rightarrow \beta$ with respect to IS_i $(1 \leq i \leq m)$ are $acc^{\pi_i}_{IS_{(m)}}(\alpha \rightarrow \beta) = \frac{freq^{\pi_i}_{IS_{(m)}}(\alpha \wedge \beta)}{freq^{\pi_i}_{IS_{(m)}}(\alpha)}$ and $cov^{\pi_i}_{IS_{(m)}}(\alpha \rightarrow \beta) = \frac{freq^{\pi_i}_{IS_{(m)}}(\alpha \wedge \beta)}{freq^{\pi_i}_{IS_{(m)}}(\beta)}$, respectively.

For a compound information system $IS^{\mathcal{O}}_{(m)} = \times(IS_1, IS_2, \ldots, IS_m)$ relational patterns are defined in an analogous way.

[15] Here, the notation of classification rule is not reversed since it is an expansion of a standard classification rule.

6.4 Illustrative example

The following example illustrates the notions introduced in this section.

Example 7 Consider the grocery store database from Example 1.

1. Relational data representation.
   The compound information system is $IS_{(3)} = \times(IS_1, IS_2, IS_3)$, where IS_1, IS_2, and IS_3 are constructed based on relations $R_1 = customer$, $R_2 = purchase$, and $R_3 = product$, respectively.
   The constrained compound information system is $IS^{\Theta}_{(3)} = \bowtie_{\Theta}(IS_1, IS_2, IS_3)$, where $\Theta = \{(R_1.id, R_2.cust\_id), (R_2.prod\_id, R_3.id)\}$, and $U_1 \bowtie_{\Theta} U_2 \bowtie_{\Theta} U_3 = \{(1, 1, 1), (1, 2, 3), (2, 3, 1), (2, 4, 3), (3, null, null), (4, 5, 6), (4, 6, 2), (5, 7, 5), (6, 8, 4), (7, null, null)\}$.
   Comment: 1. In spite of the fact that a compound system is an ordered tuple of particular information systems, the order is not essential for data representing and mining. 2. In the constrained compound information system the formulas $(R_1.id, R_2.cust\_id)$ and $(R_2.prod\_id, R_3.id)$ correspond to one-to-many relationships between tables customer and purchase, and purchase and product, respectively.
2. Relational information granule construction.
   Consider the languages $L_{IS_{(3)}}$ and $L_{IS^{\Theta}_{(3)}}$ and formulas defined in these languages $\alpha = (age, [20, 30]) \wedge (age, female) \wedge (R_1.id, R_2.cust\_id) \in L_{IS_{(3)}}, L_{IS^{\Theta}_{(3)}}$ and $\beta = \alpha \wedge (R_2.prod\_id, R_3.id) \wedge (price, [4.00, 12.00]) \in L_{IS_{(3)}}, L_{IS^{\Theta}_{(3)}}$. The semantics of α and β in $IS_{(3)}$ and $IS^{\Theta}_{(3)}$ are $SEM_{IS_{(3)}}(\alpha) = \{(4, 5), (4, 6), (5, 7)\} \times U_3$, $SEM_{IS_{(3)}}(\beta)\{(4, 5, 6), (5, 7, 5)\}$ and $SEM_{IS^{\Theta}_{(3)}}(\alpha) = \{(4, 5, 6), (4, 6, 2), (5, 7, 5)\}$ and $SEM_{IS^{\Theta}_{(3)}}(\beta) = \{(4, 5, 6), (5, 7, 5)\}$.
   Information granules corresponding to formulas α, β in $IS_{(3)}$ and $IS^{\Theta}_{(3)}$ are $(\alpha, SEM_{IS_{(3)}}(\alpha))$, $(\beta, SEM_{IS_{(3)}}(\beta))$ and $(\alpha, SEM_{IS^{\Theta}_{(3)}}(\alpha))$, $(\beta, SEM_{IS^{\Theta}_{(3)}}(\beta))$.
   Comment: The condition $(R_1.id, R_2.cust\_id)$ can be omitted in α when it is considered in $L_{IS^{\Theta}_{(3)}}$, namely the formula $\alpha = (age, [20, 30]) \wedge (age, female)$ is equivalent to α since it is automatically filtered by $(R_1.id, R_2.cust\_id)$ that is included in Θ. Analogously for β.
3. Relational pattern construction.
   Consider the system $IS^{\Theta}_{(3)}$.
   (a) The frequency of pattern α (w.r.t. IS_1) is $freq_{IS^{\Theta}_{(3)}}(\alpha) = \frac{|\{(4,5,6),(4,6,2),(5,7,5)\}|}{|U_1 \bowtie_{\Theta} U_2 \bowtie_{\Theta} U_3|} = 3/10$ ($freq^{\pi_1}_{IS^{\Theta}_{(3)}}(\alpha) = \frac{|\{4,5\}|}{|U_1|} = 2/7$). The frequency of pattern β (w.r.t. IS_1) is $freq_{IS^{\Theta}_{(3)}}(\beta) = \frac{|\{(4,5,6),(5,7,5)\}|}{|U_1 \bowtie_{\Theta} U_2 \bowtie_{\Theta} U_3|} = 1/5$ ($freq^{\pi_1}_{IS^{\Theta}_{(3)}}(\beta) = \frac{|\{4,5\}|}{|U_3|} = 1/2$).
   (b) The frequency and confidence of association rule $\alpha \rightarrow \beta$ (w.r.t. IS_1) are $freq_{IS^{\Theta}_{(3)}}(\alpha \rightarrow \beta) = freq_{IS^{\Theta}_{(3)}}(\beta) = 1/5$ and $conf_{IS^{\Theta}_{(3)}}(\alpha \rightarrow \beta) = \frac{|\{(4,5,6),(5,7,5)\}|}{|\{(4,5,6),(4,6,2),(5,7,5)\}|} = 2/3$ ($freq^{\pi_1}_{IS^{\Theta}_{(3)}}(\alpha \rightarrow \beta) = freq^{\pi_1}_{IS^{\Theta}_{(3)}}(\beta) = 1/2$ and $conf^{\pi_1}_{IS^{\Theta}_{(3)}}(\alpha \rightarrow \beta) = \frac{|\{4,5\}|}{|\{4,5\}|} = 1$).
   (c) Consider also formula $\gamma = (class, yes) \in L_{IS^{\Theta}_{(3)}}$ with the semantics $SEM_{IS_{(3)}}(\gamma) = \{(1, 1, 1), (1, 2, 3), (3, null, null), (4, 5, 6), (4, 6, 2), (5, 7, 5), (6, 8, 4)\}$.
   The accuracy and coverage of classification rule $\beta \rightarrow \gamma$ (w.r.t. IS_1) are $acc_{IS^{\Theta}_{(3)}}(\beta \rightarrow$

$\gamma) = \frac{|\{(4,5,6),(5,7,5)\}|}{|\{(4,5,6),(5,7,5)\}|} = 1$ and $cov_{IS^{\Theta}_{(3)}}(\alpha \rightarrow \gamma) = \frac{|\{(4,5,6),(5,7,5)\}|}{|\{(1,1,1),(1,2,3),(3,null,null),(4,5,6),(4,6,2),(5,7,5),(6,8,4)\}|} = 2/7$ ($acc^{\pi_1}_{IS^{\Theta}_{(3)}}(\alpha \rightarrow \gamma) = \frac{|\{4,5\}|}{|\{4,5\}|} = 1$ and $cov^{\pi_1}_{IS^{\Theta}_{(3)}}(\alpha \rightarrow \gamma) = \frac{|\{4,5\}|}{|\{1,2,4,5,6\}|} = 2/5$).

Comment: The particular information system with respect to which the quality of a pattern (e.g. frequency) is computed is usually the one corresponding to the target table.

Example 8 Analyze the suitability of the framework using the financial database from Example 2.

1. Relational data representation.
   Each database table is represented by one information system. Each relationship between two tables is represented by an expression of attribute-attribute pair. For example the part of the database that concerns tables $Client$, $Account$, and $Disposition$ can be represented by the system $IS^{\Theta}_{(3)} = \bowtie_{\Theta}(IS_1, IS_2, IS_3)$, where IS_1, IS_2, and IS_3 are constructed based on relations $R_1 = Client$, $R_2 = Account$, and $R_3 = Disposition$, respectively, and $\Theta = \{(R_1.client{-}id, R_3.client{-}id), (R_2.account{-}id, R_3.account{-}id)\}$.
2. Relational information granule construction.
   The data representation enables to construct granules using any attribute-value condition as well as attribute-attribute one. For example one can consider a granule defined by the formula $(gender, male) \wedge (R_1.client{-}id, R_3.client{-}id) \wedge (R_2.account{-}id, R_3.account{-}id) \wedge (freaquency, month) \in L_{IS^{\Theta}_{(3)}}$, which can be shortened to $(gender, male) \wedge (freaquency, month)$ if we are also interested in clients who satisfy the first condition but not the second one.

7 Discussion

This section analyzes important stages of the construction of a granular computing framework for mining relational data. It also discusses how to use a granule computing framework for building a complex system for mining relational data.

7.1 Comparative study

For simplicity's sake the frameworks from Sects. 3, 4, 5 and 6 will be referred to as $F1$, $F2$, $F3$ and $F4$, respectively.

1. Relational data representation.
   To enable the application of granular computing tools, relational data should be transformed into a typical data structure used in this paradigm, e.g. information system. The original data structure components essential for constructing patterns such as table names and joins should be preserved during the database transformation. They can be stored as metadata or used directly in the data representation construction.
   Framework $F1$ aims at analyzing data that comes from multiple standard information systems. The systems can be considered independently to one another (sum of information systems) as well as there can be taken in to account a relation joining the systems (constrained sum of information systems). This data representation makes $F1$ a proper approach for solving problems in the multi-agent system environment.

Framework $F2$ is dedicated to analyzing data stored in a double universe. It is assumed that both universes interact with each other, which is expressed by a relation joining the universes. One can observe that this data structure can be considered as a special case of that from $F1$. Namely, the database in $F2$ is that from $F1$ limited to two information systems that are joined by a characteristic function. Such a data structure can be used to encodes data used in recommender systems.

Framework $F3$ is oriented to analyze of target objects in the context of additional knowledge hidden in tables joined (in)directly with the target data. Therefore, the goal of framework $F3$ is to transform data stored in multiple tables into one collective information system with the distinguished target objects. This structure can be useful in building a decision support system, where the decision is made for target objects based on additional data.

Framework from $F4$ enables to store data from a typical relational database in the form of information systems (each corresponds to one database table) that can be joined according to relations occurring in the original database as well as by additional ones defined by an expert. The data structure without joins (compound information system) corresponds to the sum of information systems, whereas the structure with joins (constrained compound information system) can, on the one hand, be considered as a special case of the constrained sum of information systems that is specialized for typical relational databases, and on the other hand, as its extension, since multiple connections (corresponding to outer joins) of two tables can be defined at once. The structure enables to deal with multiple standard information systems that interact to one another (e.g. multi-agent system) as well as to focus on one particular system in the context of the remaining ones (e.g. decision support system).

2. Relational information granule construction.

Since the notion of granule is not, by itself, strictly defined, constructions of information granules for the same data may vary. A common feature is that a transformation of data into a granule form provides a new higher-level representation. Namely, it includes some information derived from data. The crucial task is, therefore, to use a granular representation that provides essential information for further processing, i.e. knowledge discovery.

In $F1$, an information granule in the (constrained) sum of information systems is constructed as a combination of granules defined for each particular information system. A constraint used in the (constrained) sum of information systems serves as a filter that retains granules that are valid in the system.

In $F2$, an information granule is constructed in an analogous way. Namely, granules of two particular systems of a many-to-many entity-relationship system are combined and further filtered by the relation joining the two systems. An information granule of $F2$ can be considered as a special case of that from $F1$ not only due to the dependency between the data structures of both frameworks but also by the fact that granules in $F2$ are constructed using the conjunction operator only.

In $F3$, an information granule is constructed based on the general description (i.e. generalized related set) of one target object. Such a granule corresponds to target objects that have the same descriptions as the object used during granule construction. In contrast to the remaining frameworks, $F3$ uses relational language to express granules, thanks to this, granules themselves express not only features that occur in particular tables but also connections among the tables.

The way of construction of an information granule in $F4$ is similar to that in $F1$. Namely, granules defined for each particular information system are used to construct a granule in the whole system. However, filtering of combinations of granules is done in another way. Thanks to extending the attribute-value language, connections among tables are explicitly shown in the construction of granules.

3. Relational pattern construction.
   The task of information granule construction can be seen as an intermediate step in the transformation of data into knowledge. A general granular representation of relational data can facilitate the process of discovering patterns of different types. Information granules can be viewed as components for constructing patterns. Namely, each granule shows some properties and is associated with objects sharing them. Therefore, information granules can directly be used or adapted to construct a pattern descriptor.
   The basis for discovering relational patterns in $F1$ such as association or classification rules is the construction of elementary patterns understood as conjunctions of conditions defined in particular information systems. Such patterns can be treated as information granules in the (constrained) sum of information systems that are adapted to a given data mining task. A condition of a pattern can be constructed twofold: based on an attribute of a particular information system, or based on attributes that can join two systems (sum of information systems) or additionally filter granules that relate to two systems (constrained sum of information systems).
   Association rules in $F2$ are constructed based on granules from two tables and on the relations between them. In terms of the construction, rules can be seen as a special case of patterns defined in $F1$. However, their meaning makes them a different being. Advanced similarity measures (partial and complete match) enable to interpret the meaning of granules in a variety of way.
   In $F3$ granules can be seen as elementary patterns, and they are used to construct proper ones, i.e. patterns of a required quality. Information granules in $F3$ enable to construct patterns expressed in relational language. Namely, the syntax of granules is defined using relational language, therefore, information granules can easily be transformed into standard relational patterns.
   Like in $F3$, information granules in $F4$ form a basis for constructing proper patterns. They are defined in an extended attribute-value language, thanks to this they are more similar to propositional ones. However, thanks to using the language richer than the attribute-value one, essential connections among tables are preserved and patterns can directly be applied to relational data.

In spite of the fact the all the frameworks were defined in a general granular computing environment, they are considerably influenced by rough set theory. Namely, all of them use information system—the typical rough set data structure—for storing relational data. Furthermore, three of them (i.e. $F1$, $F2$, and $F4$) apply an indiscernibility relation to form granules, whereas the remaining one divides the universe into possibly overlapping granules (similarity relation), i.e. an object may belong to different granules constructed using generalized related sets.

Table 1 summarizes the above-described approaches in terms of the data type to be mined, the language used to express knowledge, the data mining task to be performed, and the dedicated application.

Table 1 Characteristics of granular computing based frameworks for mining relational data

Framework	
Data type	
$F1$	Propositional database; relational database where join tables links two tables only
$F2$	Many-to-many case where the join table consists of two foreign keys only
$F3$	Relational database
$F4$	Propositional database, relational database
Language	
$F1$	Attribute-value language
$F2$	Attribute-value language
$F3$	Relational language
$F4$	Extended attribute-value language
Task	
$F1$	Hierarchical modeling of complex pattern
$F2$	Association discovery
$F3$	Multi-task (association discovery, classification, clustering)
$F4$	Multi-task (association discovery, classification, clustering)
Application	
$F1$	Multi-agent system based problem solving, e.g. failure diagnosis of the space robotic arm
$F2$	Recommender system, e.g. cold-start system
$F3$	General-purpose, e.g. building a decision support system
$F4$	General-purpose, e.g. building a decision support system

7.2 Towards building a granular computing based system form mining relational data

Each of the four frameworks can be used to build a granular computing based system for mining relational data (granular-relation data mining system for short). Such a system can include three modules as shown in Fig. 6.

1. Control module.
   This module is entirely constructed using knowledge provided by an expert. It includes standard components of the process of knowledge discovery such as language bias (e.g. information on relations that can be used and how can be used during pattern generation), search bias (e.g. the maximal depth level to be used during pattern generation), validation bias (e.g. quality of patterns). All the biases can be defined using a relational language.
2. Model module.
   This module is entirely constructed based on a given granular computing framework. It takes as the input a relational data model, uses a granular representation of the data (see steps relational data representation and relational information granule construction), and finally returns a granular representation of relational patterns (see step relational pattern construction).

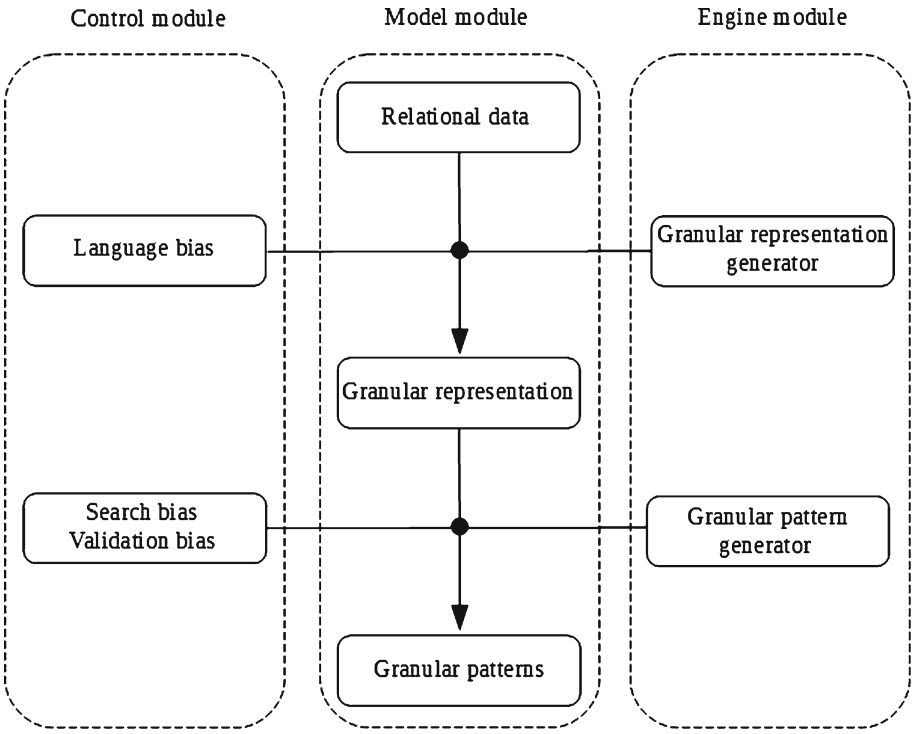

Fig. 6 A schema of a granular-relation data mining system

3. Engine module.
   This module is partially constructed based on a given granular computing framework and partially by adapting data mining tools. The first component (granular representation generator) uses a method for forming information granules based on relational data. This process is only navigated by the language bias. The second component (granular pattern generator) uses a method for forming patterns based on information granules as well as adapts a pattern generation algorithm to induce required patterns. This process is navigated by both: pattern generation strategy provided by the used algorithm, and the search and validation biases.

The proposed schema shows on a general level the incorporation of a granular framework to a whole system. The stage of transforming relational data into granular representation is in fact performed in each framework two-step: transformation of relational data into an information system based structure and generation of information granules from the transformed relational data. It means that a part of language bias instructions can be used at the intermediate step, e.g. some unneeded relations can be omitted during the construction of the information system based structure.

Furthermore, search bias instructions do not have to be associated with the stage of pattern generation only. Some of them can be used at the previous stage, depending on how refined a granule representation is to be generated. For example, in $F3$ the limitation of depth level is used during the construction of information granules. However, the level is possible to be additionally limited during the generation of patterns themselves.

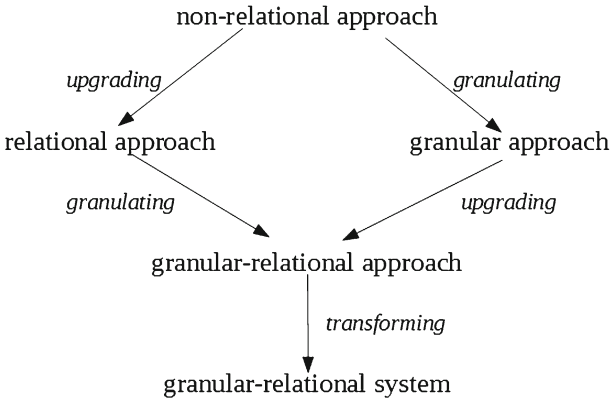

Fig. 7 A schema of transformation a non-relational approach into the granular-relation data mining system

The scheme given in Fig. 6 suggests that the granular-relation data mining system is devoted to data mining tasks that require the generation of patterns (e.g. frequent pattern and association rule discovery, decision rule discovery). However, granular representation generated under the system can be also used for other types of tasks, e.g. clustering. Namely, granular pattern generation and granular patterns can be generalized to a granular result generation, and granular results, respectively. For example, we could adapt a clustering algorithm to operate on information granules and to produce higher level granules, i.e. clusters of information granules.

Despite the fact that the system is defined based on the four previously described granular computing frameworks it can be also used to redefine or expand other data mining approaches such as those based on graph or formal concept analysis.

A data mining approach can be transformed into a granular-relation data mining system according to the scheme given in Fig. 7.

The transformation can be done for a data mining approach of any stage, i.e. non-relational, relational, granular, or granular-relational. The particular stages can be obtained using the following methodologies:

- *Non-relational* → *relational* Upgrading a data mining algorithm to a relational case (Van Laer and De Raedt 2001).
- *Non-relational* → *granular* Constructing a granule description language (Skowron and Stepaniuk 2001).
- *relational* → *granular-relational* Constructing a relational granule description language (Hońko 2015a).
- *Granular* → *granular-relational* Upgrading a granular data mining framework to a relational case (Hońko 2014).

For example, to transform the approaches described in Sects. 2.1.1 and 2.1.2, which are relational ones, we mainly need to apply a granulation procedure (relational → granular-relational). Granules in graph can be defined based of vertices, edges, or both (see, e.g. Chiaselotti et al. 2016). For instance, a set of vertices can be partitioned into subsets (elementary granules) according to a given a relation showing indiscernibility of vertices. In formal concept analysis an elementary granule can be formed based on a concept, i.e. the extension of the granule is the set of objects of the concept, whereas the intension is defined by the attributes of the concept (see, e.g. Yao 2001).

8 Concluding remarks

This paper has introduced and discussed recent developments in constructing granular computing frameworks for mining relational data. The frameworks unify the way the data and patterns are expressed and specified. They also partially standardize the process of discovering patterns from the data. Namely, the patterns can directly be obtained from the information granules or constructed based on them. In conjunction with pattern generation algorithms controlled by expert knowledge, they can build a complex system for mining relational data.

The frameworks can be summarized as follows.

1. Constrained sums of information systems.
   It is intended to mine complex data that is structured as separate universes, which can alternatively interact with one another (e.g. multi-agent system). The interaction can be defined using any relation over the universes. The framework can produce patterns expressed in an attribute-value language with additional expressions showing interactions.
2. Granular association rule approach.
   It is dedicated to mine associations occurring between two universes that are dependent on each other (e.g. recommender systems). The dependency is expressed using a characteristic function. It can be viewed a specialized version of the first approach in terms of data representation and pattern expression.
3. Generalized related set based approach.
   It is oriented toward constructing essential descriptions of target objects (e.g. descriptions distinguishing objects from different classes) using background knowledge that is hidden in tables directly or indirectly joined with the target one (e.g. decision support system). Patterns in this approach are expressed in the native language, i.e. relational one.
4. Description language based approach.
   It is a modified version of the first approach and is dedicated to mine data coming from typical relational databases (e.g. decision support system). Patterns are expressed in an extended attribute-value language that enables to define conditions on two key attributes (i.e. attribute-attribute condition).

From the theoretical viewpoint the introduced frameworks fill the gap between two research areas: relational data mining and granular computing. Unlike other existing approaches, they comprehensively define relational data, information, and knowledge in the context of granular computing.

In practice, the approaches provide more unified frameworks for mining relational data. The common granular representation of data is the basis for performing different relational data mining tasks.

Acknowledgements The project was partially funded by the National Science Center awarded on the basis of the decision number DEC-2012/07/B/ST6/01504.

References

Al-Hmouz R, Pedrycz W, Balamash AS (2015) Description and prediction of time series: a general framework of granular computing. Expert Syst Appl 42(10):4830–4839

Antonelli M, Ducange P, Lazzerini B, Marcelloni F (2016) Multi-objective evolutionary design of granular rule-based classifiers. Granul Comput 1(1):37–58

Apolloni B, Bassis S, Rota J, Galliani GL, Gioia M, Ferrari L (2016) A neurofuzzy algorithm for learning from complex granules. Granul Comput 1(4):225–246

Azmeh Z, Huchard M, Napoli A, Hacene MR, Valtchev P (2011) Querying relational concept lattices. In: Napoli A, Vychodil V (eds) Proceedings of The eighth international conference on concept lattices and their applications, Nancy, France, October 17–20, 2011, CEUR-WS.org, vol 959, pp 377–392

Bargiela A, Pedrycz W (2003) Granular computing: an introduction. Kluwer Academic Publishers, Boston

Bargiela A, Pedrycz W (2008) Toward a theory of granular computing for human-centered information processing. IEEE Trans Fuzzy Syst 16(2):320–330

Blaťák J (2005) First-order frequent patterns in text mining. In: Carlos Bento GD, Amilcar Cardoso (ed) 2005 Portuguese conference on artificial intelligence. IEEE, pp 344–350

Chiaselotti G, Ciucci D, Gentile T (2016) Simple graphs in granular computing. Inf Sci 340–341:279–304

Ciucci D (2016) Orthopairs and granular computing. Granul Comput 1(3):159–170

Cook DJ, Holder LB (2000) Graph-based data mining. IEEE Intell Syst 15(2):32–41

Dolques X, Le Ber F, Huchard M, Nebut C (2016) Relational concept analysis for relational data exploration. In: Guillet F, Pinaud B, Venturini G, Zighed DA (eds) Advances in knowledge discovery and management, vol 5. Springer, Berlin, pp 57–77

Dubois D, Prade H (2016) Bridging gaps between several forms of granular computing. Granul Comput 1(2):115–126

Džeroski S, Lavrač N (2001b) Relational data mining. Springer, Berlin

Džeroski S, Lavrač N (2001a) An introduction to inductive logic programming. In: Džeroski and Lavrač (2001b). Springer, pp 48–71

Eissa MM, Elmogy M, Hashem M (2016) Roughgranular computing knowledge discovery models for medical classification. Egypt Inform J 17(3):265–272

Ferré S, Ridoux O, Sigonneau B (2005) Arbitrary relations in formal concept analysis and logical information systems. In: Dau F, Mugnier ML, Stumme G (eds) Conceptual structures: common semantics for sharing knowledge: 13th international conference on conceptual structures, ICCS 2005. Springer, Berlin, pp 166–180

Ganivada A, Dutta S, Pal SK (2011) Fuzzy rough granular neural networks, fuzzy granules, and classification. Theor Comput Sci 412(42):5834–5853

Ganter B, Wille R (1999) Formal concept analysis: mathematical foundations. Springer, Berlin

Ganter B, Stumme G, Wille RE (2005) Formal concept analysis, foundations and applications, vol 3626. Springer, Berlin

Guo J, Zheng L, Li T (2007) An efficient graph-based multi-relational data mining algorithm. In: Computational intelligence and security, international conference, CIS 20077. IEEE Computer Society, pp 176–180

Holder LB, Cook DJ, Coble J, Mukherjee M (2005) Graph-based relational learning with application to security. Fundam Inform 66(1–2):83–101

Hońko P (2010) Simialrity-based classification in relational databases. Fundam Inform 101(3):187–213

Hońko P (2013a) Association discovery from relational data via granular computing. Inform Sci 234:136–149

Hońko P (2013b) Granular computing for relational data classification. J Intell Inf Syst 41(2):187–210

Hońko P (2014) Upgrading a granular computing based data mining framework to a relational case. Int J Intell Syst 29(5):407–438

Hońko P (2015a) Description languages for relational information granules. Fundam Inform 137(3):323–340

Hońko P (2015b) Relation-based granules to represent relational data and patterns. Appl Soft Comput 37(C):467–478

Hu X, Pedrycz W, Wang X (2015) Comparative analysis of logic operators: a perspective of statistical testing and granular computing. Int J Approx Reason 66:73–90

Huchard M, Hacene MR, Roume C, Valtchev P (2007) Relational concept discovery in structured datasets. Ann Math Artif Intell 49(1–4):39–76

Kavurucu Y, Mutlu A, Ensari T (2016) Graph-based concept discovery in multi relational data. In: 6th international conference on cloud system and big data engineering. IEEE, pp 274–278

Ketkar NS, Holder LB, Cook DJ (2005) Comparison of graph-based and logic-based multi-relational data mining. SIGKDD Explor Newsl 7(2):64–71

Knobbe AJ, Siebes A, Blockeel H, Wallen DVD (2000) Multi-relational data mining, using UML for ILP. In: Principles of data mining and knowledge discovery, pp 1–12

Knobbe AJ (2006) Multi-relational data mining. IOS Press, Amsterdam
Kötters J (2011) Object configuration browsing in relational databases. In: Valtchev P, Jäschke R (eds) Formal concept analysis—9th international conference, ICFCA 2011, Nicosia, Cyprus, May 2–6, 2011. Proceedings, Springer, Lecture Notes in Computer Science, vol 6628, pp 151–166
Kramer S, Lavrač N, Flach P (2001) Propositionalization approaches to relational data mining. In: Džeroski and Lavrač (eds) Springer, pp 262–291
Kreinovich V (2016) Solving equations (and systems of equations) under uncertainty: how different practical problems lead to different mathematical and computational formulations. Granul Comput 1(3):171–179
Kundu S, Pal SK (2015) FGSN: fuzzy granular social networks model and applications. Inf Sci 314(Supplement C):100–117
Kuželka O, Železný F (2008) HiFi: Tractable propositionalization through hierarchical feature construction. In: Železný F, Lavrač N (eds) Late breaking papers, the 18th international conference on inductive logic programming, pp 1–6
Lan S, Xiangzhi H (2007) Rough set model with double universe of discourse. In: Proceedings of the IEEE international conference on information reuse and integration. IEEE Systems, Man, and Cybernetics Society, pp 492–495
Lavrač N, Džeroski S, Grobelnik M (1991) Learning nonrecursive definitions of relations with LINUS. In: Kodratoff Y (ed) Machine learning—EWSL-91: European working session on learning Porto, Portugal, March 6–8, 1991 Proceedings. Springer, Berlin, pp 265–281
Li J, Mei C, Xu W, Qian Y (2015) Concept learning via granular computing: a cognitive viewpoint. Inf Sci 298:447–467
Lin TY (2008) Granular computing: common practices and mathematical models. In: Proceedings of the IEEE international conference on fuzzy systems (FUZZ-IEEE 2008). IEEE Computer Society, pp 2405–2411
Lin TY (2005) Introduction to special issues on data mining and granular computing. Int J Approx Reason 40(1–2):1–2
Lin TY, Zadeh LA (2004) Special issue on granular computing and data mining. Int J Intell Syst 19(7):565–566
Lingras P, Haider F, Triff M (2016) Granular meta-clustering based on hierarchical, network, and temporal connections. Granul Comput 1(1):71–92
Livi L, Sadeghian A (2016) Granular computing, computational intelligence, and the analysis of non-geometric input spaces. Granul Comput 1(1):13–20
Mendel JM (2016) A comparison of three approaches for estimating (synthesizing) an interval type-2 fuzzy set model of a linguistic term for computing with words. Granul Comput 1(1):59–69
Milton RS, Maheswari VU, Siromoney A (2005) Studies on rough sets in multiple tables. In: Slezak D, Wang G, Szczuka MS, Duentsch I, Yao Y (eds) RSFDGrC (1), Lecture Notes Computer Science, vol 3641. Springer, pp 265–274
Min F, Hu Q, Zhu W (2012) Granular association rules with four subtypes. In: 2012 IEEE international conference on granular computing. IEEE, pp 353–358
Nagao M, Seki H (2016) On mining quantitative association rules from multi-relational data with FCA. In: 9th IEEE international workshop on computational intelligence and applications, IWCIA 2016. IEEE, pp 81–86
Pal SK, Banerjee R (2013) Context granulation and subjective-information quantification. Theor Comput Sci 488(Supplement C):2–14
Pal SK, Shankar BU, Mitra P (2005) Granular computing, rough entropy and object extraction. Pattern Recogn Lett 26(16):2509–2517
Pal SK, Meher SK, Dutta S (2012) Class-dependent rough-fuzzy granular space, dispersion index and classification. Pattern Recogn 45(7):2690–2707
Pal SK, Meher SK, Skowron A (2015) Data science, big data and granular mining. Pattern Recogn Lett 67:109–112
Pawlak Z (1991) Rough sets. Theoretical aspects of reasoning about data. Kluwer Academic, Dordrecht
Pedrycz W, Skowron A, Kreinovich V (2008) Handbook of granular computing. Wiley, New York
Peters G, Weber R (2016) DCC: a framework for dynamic granular clustering. Granul Comput 1(1):1–11
Ray SS, Ganivada A, Pal SK (2016) A granular self-organizing map for clustering and gene selection in microarray data. IEEE Trans Neural Netw Learn Syst 27(9):1890–1906
Skowron A, Stepaniuk J (2001) Information granules: towards foundations of granular computing. Int J Intell Syst 16(1):57–85
Skowron A, Stepaniuk J (2004) Constrained sums of information systems. In: Tsumoto S, Slowinski R, Komorowski HJ, Grzymala-Busse JW (eds) Rough sets and current trends in computing, lecture notes in computer science, vol 3066. Springer, Berlin, pp 300–309
Skowron A, Stepaniuk J, Swiniarski R (2012) Modeling rough granular computing based on approximation spaces. Inf Sci 184(1):20–43

Skowron A, Jankowski A, Dutta S (2016) Interactive granular computing. Granul Comput 1(2):95–113
Stepaniuk J (2008) Rough-granular computing in knowledge discovery and data mining. Stud Comp Intell 152. Springer, Berlin
Van Laer W, De Raedt L (2001) How to upgrade propositional learners to first order logic: a case study. In: Džeroski and Lavrač (2001b), Springer, pp 235–261
Washio T, Motoda H (2003) State of the art of graph-based data mining. SIGKDD Explor Newsl 5(1):59–68
Wilke G, Portmann E (2016) Granular computing as a basis of human-data interaction: a cognitive cities use case. Granul Comput 1(3):181–197
Yan R, Zheng J, Liu J, Zhai Y (2010) Research on the model of rough set over dual-universes. Knowl Based Syst 23(8):817–822
Yang Q, Wu X (2006) 10 challenging problems in data mining research. Int J Inf Techol Decis Mak 5(4):597–604
Yao JT (2005) Information granulation and granular relationships. In: Hu X, Liu Q, Skowron A, Lin TY, Yager RR, Zhang B (eds) Proceedings of the IEEE conference on granular computing. IEEE Computer Society, pp 326–329
Yao Y (2001) On modeling data mining with granular computing. In: 25th International computer software and applications conference (COMPSAC 2001), invigorating software development, 8–12 October 2001, Chicago, IL, USA, p 638
Yao YY (2000) Granular computing: basic issues and possible solutions. In: Wang P (ed) Proceedings of the 5th joint conference on information sciences (JCIS). Association for Intelligent Machinery, pp 186–189
Yao YY (2004) A comparative study of formal concept analysis and rough set theory in data analysis. In: Tsumoto S, Slowinski R, Komorowski HJ, Grzymala-Busse JW (eds) Rough sets and current trends in computing, lecture notes computer sciences, vol 3066. Springer, Berlin, pp 59–68
Yao Y (2016) A triarchic theory of granular computing. Granul Comput 1(2):145–157
Yao JT, Vasilakos AV, Pedrycz W (2013) Granular computing: perspectives and challenges. IEEE Trans Cybernet 43(6):1977–1989
Yu PS, Yin X, Yang J, Han J (2006) Efficient classification across multiple database relations: a crossmine approach. IEEE Trans Knowl Data Eng 18:770–783
Zadeh LA (1997) Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. Fuzzy Set Syst 90(2):111–127
Železný F, Lavrač N (2006) Propositionalization-based relational subgroup discovery with RSD. Mach Learn 62(1):33–63