Chapter 2
Information System for Relational Data

2.1 Introduction

The goal of this chapter is to develop a general granular computing based framework for mining relational data. It is based on an information system defined for relational data [38]. Information granules derived from the information system are defined based on the notion of related sets, that is sets of objects related (i.e. joined) to the objects to be analyzed. Such granules are the basis for discovering relational knowledge.

The crucial task of the general framework is to process relational data for discovering patterns of different types. Namely, information granules obtained in the framework can be viewed as an abstract representation of relational data. Such a representation is treated as the search space for discovering relational patterns. Thanks to this, the size of the search space may be significantly limited.

The framework is independent on the way the language bias is specified, thereby biases from existing frameworks can be adapted. Furthermore, the framework, unlike others (i.e. ILP, RDB), unifies not only the way the data and patterns are expressed and specified, but also partially the process of discovering patterns from the data. Namely, the patterns can directly be obtained from the information granules or constructed based on them.

Applying the granular computing idea makes it possible to switch between different levels of granularity of the same universe (i.e. the set of objects), thereby one can choose an appropriate granularity of the data for a given task.

In the framework, one can define new methods as well as redefined existing ones for performing popular relational data mining tasks.

P. Hońko, *Granular-Relational Data Mining*, Studies in Computational Intelligence 702, DOI 10.1007/978-3-319-52751-2_2

The remaining of the chapter is organized as follows. Section 2.2 constructs an information system for relational data. Section 2.3 defines a granular description of relational objects that is based on the notion of generalized related sets. Section 2.4 shows how to construct relational patterns based on introduced granules. Section 2.5 provides concluding remarks.

2.2 Relational Data

It is assumed that we are given relational data that resides in a relational database; however, the framework can also be defined for data stored in a deductive database.

Definition 2.1 (*Relational database*) A relational database can be defined in the context of MRDM by the following notions.

- A relation schema is an expression of the form $R(a_1, a_2, \ldots, a_n)$, where R is a relation name, and a_i $(1 \leq i \leq n)$ are the attributes.
- A relation is a subset of the Cartesian product $V_{a_1} \times V_{a_2} \times \cdots \times V_{a_n}$, where V_{a_i} $(1 \leq i \leq n)$ are the value sets of attributes a_i.
- A relational database $D = T \cup B$ is a collection of logically connected relations, where $T = \{R_1^T, R_2^T, \ldots, R_{n_T}^T\}$ and $B = \{R_1^B, R_2^B, \ldots, R_{n_B}^B\}$ consist of target and background relations, respectively.

The target table (i.e. relation[1]) includes objects to be analyzed, e.g. objects for which association rules are mined. Such objects may reside in more than one table; for example, each target table includes the objects of one class. Background tables include additional objects which are directly or indirectly joined to the objects of the target table. The same terms are used for the objects of the target and background tables, i.e. the target and background objects.

Example 2.1 Given a database $D = \{customer\} \cup \{product, purchase\}$ for the customers of a grocery store.

customer

id	name	age	gender	income	class
1	Adam Smith	36	male	1500	yes
2	Tina Jackson	33	female	2500	yes
3	Ann Thompson	30	female	1800	no
4	Susan Clark	30	female	1800	yes
5	Eve Smith	26	female	2500	yes
6	John Clark	29	male	3000	yes
7	Jack Thompson	33	male	1800	no

married_to

id	cust_id_1	cust_id_2
1	5	1
2	6	4
3	3	7

[1]The notions of relation and table are used in this monograph interchangeably.

purchase

id	cust_{id}	prod_{id}	amount	date
1	1	1	1	24/06
2	1	3	2	24/06
3	2	1	1	25/06
4	2	3	1	26/06
5	4	6	1	26/06
6	4	2	3	26/06
7	6	5	3	27/06
8	3	4	1	27/06

product

id	name	price
1	bread	2.00
2	butter	3.50
3	milk	2.50
4	tea	5.00
5	coffee	6.00
6	cigarettes	12.00

The target table *customer* includes basic data about customers. The data is divided into two groups according to the values of the attribute *class*. The background tables include information on marriage couples (*married_to*) and that on products purchased by the customers (*product* and *purchase*).

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

Definition 2.2 (*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.

For example, the first tuple of table *customer* from Example 2.1 is represented by the object *customer*(1, *Adam Smith*, 36, *male*, 1500, *yes*).

A relational database is represented by an information system that is constructed based on the standard information system [71].[2]

Definition 2.3 (*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.

The information system for storing relational data is constructed as follows. Consider a database $D = T \cup B$. Let $U_{D_T} = T$, $U_{D_B} = B$, $A_{D_T} = \bigcup_{R \in T} A_R$,[3] and $A_{D_B} = \bigcup_{R \in B} A_R$.

Definition 2.4 (*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

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

[2] The standard information system is understood as the Pawlak information system.

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

Example 2.2 Database D of Example 2.1 can be represented by 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} = {*customer*(1, *Adam Smith*, 36, *male*, 1500, *yes*), ..., *customer*(7, *Jack Thompson*, 33, *male*, 1800, *no*)},
U_{D_B} = {*married_to*(1, 5, 1) ..., *married_to*(3, 3, 7), *purchase*(1, 1, 1, 1, 24/06), ..., *purchase*(8, 3, 4, 1, 27/06), *product*(1, *bread*, 2.00), ..., *product*(6, *cigarettes*, 12.00)},
A_{D_T} = {*customer.id*, *customer.name*, *customer.age*, *customer.gender*, *customer.income*, *customer.class*},
A_{D_B} = {*married_to.id*, *married_to.cust_id*$_1$, *married_to.cust_id*$_2$, *purchase.id*, *purchase.cust_id*, *purchase.prod_id*, *purchase.amount*, *purchase.date*, *product.id*, *product.name*, *product.price*}.[4]

2.3 Relational Information

Essential information acquired from relational data is expressed by descriptions of target objects. The descriptions are used in a sense to identify the objects, i.e. the objects are compared to each other or to patterns (e.g. classification rules) based on their descriptions. For each target object its description is constructed based on background relations. To construct such descriptions, the notion of related set is introduced [36].

Definition 2.5 (*Related objects*) Object o is related to object o', denoted by $o \sim o'$, if and only if there exists a key attribute joining o with o'.[5]

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 description is expressed by a set of background objects joined with the target object. More precisely.

Definition 2.6 (*Related set*) 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.

Each target object in this approach is processed along with its related set.

Example 2.3 Consider the target objects o_1 = *customer*(1, *Adam Smith*, 36, *male*, 1500, *yes*), o_2 = *customer*(2, *Tina Jackson*, 33, *female*, 2500, *yes*) from the information system of Example 2.2.

[4] It is assumed that the value of an attribute is specified for a given object if and only if the object belongs to the relation whose schema includes the attribute.

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

The related sets of o_1 and o_2 are $rlt(o_1) = \{married\_to(1, 5, 1), purchase(1, 1, 1, 1, 24/06), purchase(2, 1, 3, 2, 24/06), product(1, bread, 2.00), product(3, milk, 2.50)\}$ and $rlt(o_2) = \{purchase(3, 2, 1, 1, 25/06), purchase(4, 2, 3, 1, 26/06), product(1, bread, 2.00), product(3, milk, 2.50)\}$, respectively.
The objects of relation *purchase* (*product*) are directly (indirectly) related to the target objects by attribute *c_id* (by relation *purchase* and attribute *p_id*).

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 description (i.e. the related set) can be tuned by its depth level. To define a related set of a given depth level, Definition 2.5 is generalized.

Definition 2.7 (*n-related objects*) Object o_0 is n-related to object o_n, denoted by $o_0 \overset{n}{\sim} o_n$, if and only if there exists o_{i+1} such that $o_i \sim o_{i+1}$, where $n > 0$ and $0 \leq i \leq n - 1$.

One can note that for $n = 1$ Definitions 2.5 and 2.7 are equivalent.
A related set of a given depth level is defined as follows.

Definition 2.8 (*n-related set*) The nth 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 $m \leq n$.

It is assumed that for each $o \in U_{D_T}$ we have $rlt^0(o) = \emptyset$. It is reasonable to consider a target object without its related set (i.e. the related set is empty) when the object itself includes information, i.e. descriptive attributes occur in the target relation (e.g. attribute *class* in relation *customer*).

Example 2.4 Consider the target object o_2 from Example 2.3.
We can obtain two different non-empty descriptions of o, namely $rlt^1(o_2) = \{purchase(3, 2, 1, 1, 25/06), purchase(4, 2, 3, 1, 26/06)\}$ and $rlt^2(o_2) = rlt(o_2)$.

A target object with its related sets can be presented in the form of a graph.

Definition 2.9 (*Directed graph of related set*) Given a target object o. Let $dl(o')$ be the depth level of an object $o' \in \{o\} \cup rlt(o)$. A target object o with its related set $rlt(o)$ can be presented in the form of the directed graph $G_o = (V, E)$ where $V = \{o\} \cup rlt(o)$ and $E = \{(o', o'') \in V \times V : o' \sim o'', dl(o') < dl(o'')\}$.

The directed graph illustrates how a related set of a given target object is formed (see Fig. 2.1). If the way the object description is formed is not essential, a target object with its related set can be presented using an undirected graph.

Definition 2.10 (*Undirected graph of related set*) A target object o with its related set $rlt(o)$ can be presented in the form of the undirected graph $G_o = (V, E)$ where $V = \{o\} \cup rlt(o)$ and $E = \{\{o', o''\} \subseteq V : o' \sim o''\}$.

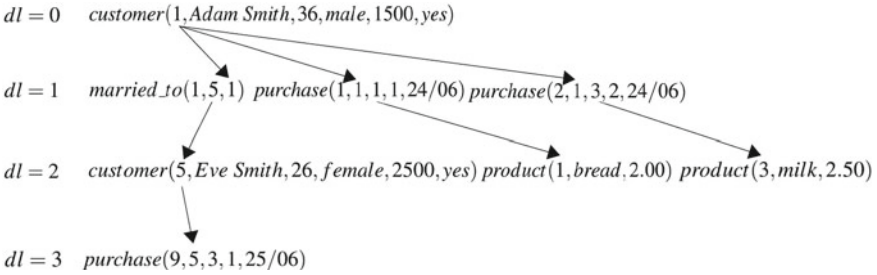

Fig. 2.1 Directed graph for the first customer from Example 2.1 (For illustrative purposes table *purchase* is extended by the tuple (9, 5, 3, 1, 25/06).)

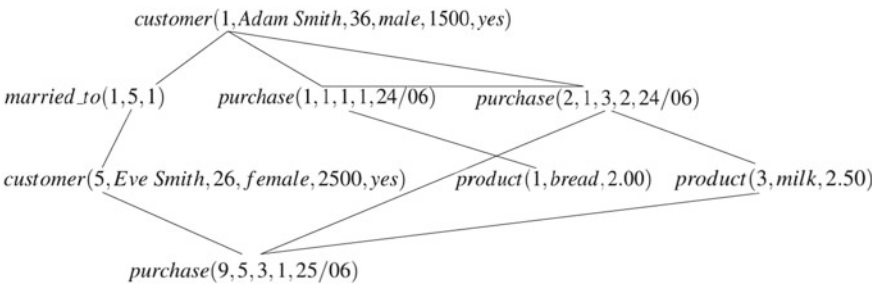

Fig. 2.2 Undirected graph for the first customer from Example 2.1

An undirected graph enables to check if two object are *n*-related.

Proposition 2.1 *Given a target object o and its related set rlt*(o)*. Objects* o' *and* o'' *such that* $o', o'' \in \{o\} \cup rlt(o)$ *are n-related if and only if there exists in* G_o *a path of length n joining* o' *and* o''*.*

As it can be observed in Fig. 2.2, two objects can be related in more than one way. For example, objects *customer*(1, *Adam Smith*, 36, *male*, 1500, *yes*) and *purchase*(2, 1, 3, 2, 24/06) are 1-related and 2-related.

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

Definition 2.11 (*Generalized target object*) A generalized target object o, denoted by o_{gen}, is the target object with certain components replaced according to a given substitution.[6]

[6] A component of an object can be replaced with either a variable, a set of constants, or symbol "_" if the component is not important for the consideration.

Definition 2.12 (*Generalized related set*) A generalized related set of a target object o, denoted by $rlt_{gen}(o)$, is the related set with certain components replaced according to the substitution (partially) constructed during generalization of the target object.

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 [36]). A method for generalization can be developed taking into consideration a language bias.

Example 2.5 Consider again the target object $o = customer(2, Tina\ Jackson, 33, female, 2500, yes)$ from Example 2.3 and its related set $rlt^2(o) = \{purchase(3, 2, 1, 1, 25/06), purchase(4, 2, 3, 1, 26/06), product(1, bread, 2.00), product(3, milk, 2.50)\}$.

The generalized target object and its related set can be of the following forms $o_{gen} = customer(A, \_, \_, \_, \_, yes)$ and $rlt^2_{gen}(o) = \{purchase(B, A, C, \_, \_), product(C, \{bread, milk\}, \_)\}$,[7] respectively.

An object of the relation *customer* can be generalized according to the following language bias constraint $mode(customer(+type(c\_id), \_, \_, \_, \#[yes, no]))$, which means that the first argument of the relation *customer* has to be replaced with an input variable of a type that is the same as that of attribute *c_id*, the last one can be replaced with *yes* or *no* (i.e. the class label), and the remaining arguments are omitted. Object o is generalized according to the substitution $\{2/A, Tina\ Jackson/\_, 33/\_, female/\_, income/\_\}$.

As presented above, each target object is represented by the set of background objects related to the target object. It is natural to treat such a set as *a granule of objects drawn together by their relationships with the target object*. Therefore, we consider a granule defined by the pair $(o, rlt(o))$, where o is a target object from a given information system.

For generalized related sets, information granules are defined by their syntax and semantics. For this purpose, the method for constructing information granules [83] is extended to a relational case.

In the approach, an elementary granule is defined by a conjunction of relational descriptors, i.e. expressions of the form $R(t_1, t_2, \ldots, t_n)$, where R is a relation name, and t_i $(1 \leq i \leq n)$ are the terms (constants or variables).

Given information system $IS_D = (U_D, A_D)$.

- A generalized target object o_{gen} of object o from IS_D is a trivial elementary granule, i.e. a single relational descriptor.
  The meaning (i.e. semantics) of the granule, denoted by $SEM_{IS_D}(o_{gen})$, is the set of target objects that satisfy the descriptor.

[7] The denotation $\{v_1, v_2, \ldots, v_n\}$ that occurs in an object argument list means that the corresponding attribute may take any of the values $v_1, v_2, \ldots, v_n$. We assume that sets are formed for attributes that take on a relatively small number of values. Otherwise, the attributes are previously discretized.

- A generalized related set $rlt_{gen}(o)$ of target object o from IS_D is an elementary granule where each descriptor is constructed based on a background relation.
  The meaning of the granule, denoted by $SEM_{IS_D}(rlt_{gen}(o))$, is the set of target objects for each of which there exists a substitution such that each descriptor under the substitution is satisfied.
- A generalized target object o_{gen} with its generalized related set $rlt_{gen}(o)$ is represented by the granule $(o_{gen}, rlt_{gen}(o))$.
  The meaning of the granule is $SEM_{IS_D}\left((o_{gen}, rlt_{gen}(o))\right) = (SEM_{IS_D}(o_{gen}), SEM_{IS_D}(rlt_{gen}(o)))$.

Example 2.6 Consider the generalized target object from Example 2.5: $o_{gen} = customer(A, \_, \_, \_, \_, yes)$ and $rlt^2_{gen}(o) = \{purchase(B, A, C, \_, \_), product(C, \{bread, milk\}, \_)\}$.
The meaning of the granule $(o_{gen}, rlt^2_{gen}(o))$ is $SEM_{IS_D}\left((o_{gen}, rlt^2_{gen}(o))\right) = (\{o_1, o_2, o_4, o_5, o_6\}, \{o_1, o_2\})$ (o_i stands for the i-th customer of database D).

Information granules defined as above can be viewed as an abstract representation of relational data. The accuracy level of the representation can easily be changed by taking other depth level of related sets. Furthermore, a representation constructed based on the information granules obtained for all target objects is treated in the approach as the search space for discovering patterns. Thanks to this, the size of the search space may significantly be limited.

A granularity of the universe is defined by the set $\{SEM_{IS_D}(rlt^n_{gen}(o)) : o \in U_{D_T}\}$. Thus different depth levels of related sets correspond to different levels of information granulation. As the depth level increases, a lower-level granularity is obtained.

2.4 Relational Knowledge

The information granules defined in the previous section are the basis for the discovery of relational knowledge. Thanks to constructing such granules we are able to obtain knowledge of different types. Therefore, we can consider as granules, e.g. frequent patterns and relational association rules, relational classification rules, and relational clusters and their descriptions.

Firstly, basic definitions will be restated (cf. [25]).

Definition 2.13 (*Relational pattern*) A relational pattern is an expression of the form[8]

$$R_1(t^1_1, t^1_2, \ldots, t^1_{n_1}) \wedge R_2(t^2_1, t^2_2, \ldots, t^2_{n_2}) \wedge \cdots \wedge R_m(t^m_1, t^m_2, \ldots, t^m_{n_m}),$$

where R_i $(1 \leq i \leq m)$ are relations, and indexed t are the terms (constants or variables).

[8]One of relations R_i is usually considered as the target one. However, such a relation, as in this approach, may be determined externally, i.e. it occurs in the database but not in the pattern.

For simplicity's sake we denote a relational pattern as α.
The frequency of a pattern α is the ratio between the number of objects that satisfy α and the number of all objects under consideration.

Definition 2.14 (*Relational frequent pattern*) A relational frequent pattern is a relational pattern that occurs in a given database with the frequency not less than a given threshold.

Definition 2.15 (*Relational association rule*) An association rule is an expression of the form $\alpha \rightarrow \beta$, where α and β are relational (frequent) patterns and α is more general than β.

The frequency of an association rule $\alpha \rightarrow \beta$ is the frequency of β. The confidence of association rule $\alpha \rightarrow \beta$ is the ratio between the frequency of β and that of α.

Definition 2.16 (*Relational classification rule*) A relational classification rule is an expression of the form[9]

$$R(t_1, t_2, \ldots, t_n) \leftarrow R_1(t_1^1, t_2^1, \ldots, t_{n_1}^1) \wedge R_2(t_1^2, t_2^2, \ldots, t_{n_2}^2) \wedge \cdots \wedge R_m(t_1^m, t_2^m, \ldots, t_{n_m}^m),$$

where R is a target relation, R_i $(1 \leq i \leq m)$ are background relations, and indexed t are terms.

For simplicity, a relational classification rule is denoted as $\alpha \leftarrow \beta$.
The accuracy (coverage) of the rule $\alpha \leftarrow \beta$ is the ratio between the number of objects that satisfy $\alpha \wedge \beta$ and the number of objects that satisfy β (α).

Example 2.7 Assume that we discover associations involving the customers from database D of Example 2.1. Table *customer* is therefore the target one, however the division into classes is not taken into account.
Given patterns $\alpha = customer(A, \_, \_, \_, \_, \_) \wedge purchase(B, A, C, \_, \_)$ and $\beta = \alpha \wedge product(C, \{bread, milk\}, \_)\}$. Patterns α and β are satisfied by objects o_1, o_2, o_3, o_4, o_6 and o_1, o_2, respectively. Hence, the frequencies of α and β are 5/7 and 2/7, respectively.
Since α is more general than β we can build the following association rule $\alpha \rightarrow \beta$. The frequency and confidence of $\alpha \rightarrow \beta$ are 2/7 and 2/5, respectively.

Consider information system $IS_D = (U_D, A_D)$. Relational patterns are represented by granules as follows.

- A relational (frequent) pattern α in IS_D is represented by the granule $(o_{gen}, rlt_{gen}(o))$. The meaning of the granule is $SEM_{IS_D}(\alpha) = (SEM_{IS_D}(o_{gen}), SEM_{IS_D}(rlt_{gen}(o)))$.
  The pattern's frequency can be calculated by $freq_{IS_D}(\alpha) = \frac{card(SEM_{IS_D}(rlt_{gen}(o)))}{card(SEM_{IS_D}(o_{gen}))}$.

[9] One can also consider rules including negated descriptors or conditions formed based on arguments of descriptors previously added.

- A set of relational (frequent) patterns is represented by the set of granules $\{\alpha_i : 1 \leq i \leq k\}$, where k is the cardinality of the set of rules.
  The meaning of the granule is $\{SEM_{IS_D}(\alpha_i) : 1 \leq i \leq k\}$.
- 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 $SEM_{IS_D}(rlt_{gen}(o)) \subseteq SEM_{IS_D}(rlt'_{gen}(o))$.
  The meaning of the granule is $SIM_{IS_D}((\alpha, \beta)) = (SIM_{IS_D}(\alpha), SIM_{IS_D}(\beta))$.
  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, $SIM_{IS_D}((\alpha, \beta)) = (SEM_{IS_D}(o_{gen}), SEM_{IS_D}(rlt'_{gen}(o)), SEM_{IS_D}(rlt_{gen}(o)))$.
  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.
- A set of relational association rules is represented by the set of granules $\{(\alpha_i, \beta_i) : 1 \leq i \leq k\}$, where k is the cardinality of the set of rules.
  The meaning of the granule is $\{SEM_{IS_D}((\alpha_i, \beta_i)) : 1 \leq i \leq k\}$.
- A relational classification rule $\alpha \leftarrow \beta$ in IS_D is represented by the granule (α, β), where α and β correspond to o_{gen} and $rlt_{gen}(o)$, respectively.
  The meaning of the granule is $SIM_{IS_D}((\alpha, \beta)) = (SIM_{IS_D}(\alpha), SIM_{IS_D}(\beta))$.
  The rule's accuracy and coverage can be computed 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.
- A set of relational classification rules is represented by the set of granules $\{(\alpha_i, \beta_i) : 1 \leq i \leq k\}$, where k is the cardinality of the set of rules.
  The meaning of the granule is $\{SEM_{IS_D}((\alpha_i, \beta_i)) : 1 \leq i \leq k\}$.

Example 2.8 Given information system IS_D from Example 2.2 and patterns $\alpha = customer(A, \_, \_, \_, \_, \_) \wedge purchase(B, A, C, \_, \_)$ and $\beta = \alpha \wedge product(C, \{bread, milk\}, \_)$.
Consider the following generalizations of the object $o = customer(2, Tina\ Jackson, 33, female, 2500, yes)$: $o_{gen} = customer(A, \_, \_, \_, \_, \_)$, $rlt^1_{gen}(o) = \{purchase(B, A, C, \_, \_)\}$, $rlt^2_{gen}(o) = \{purchase(B, A, C, \_, \_), product(C, \{bread, milk\}, \_)\}$.
Patterns α and β can be represented, respectively, by granules $(o_{gen}, rlt^1_{gen}(o))$ and $(o_{gen}, rlt^2_{gen}(o))$ with the meanings $SEM_{IS_D}(\alpha) = (\{o_1, \ldots, o_7\}, \{o_1, o_2, o_3, o_4, o_6\})$ and $SEM_{IS_D}(\beta) = (\{o_1, \ldots, o_7\}, \{o_1, o_2\})$.
The frequencies of α and β are $freq_{IS_D}(\alpha) = \frac{|SEM_{IS_D}(rlt^1_{gen}(o))|}{|SEM_{IS_D}(o_{gen})|} = 5/7$ and $freq_{IS_D}(\beta) = \frac{|SEM_{IS_D}(rlt^2_{gen}(o))|}{|SEM_{IS_D}(o_{gen})|} = 2/7$.
Consider also the association rule $\alpha \rightarrow \beta$. The meaning of the rule is $SEM_{IS_D}(\alpha \rightarrow \beta) = (\{o_1, \ldots, o_7\}, \{o_1, o_2, o_3, o_4, o_6\}, \{o_1, o_2\})$. The frequency and confidence of $\alpha \rightarrow \beta$ are $freq_{IS_D}(\alpha \rightarrow \beta) = freq_{IS_D}(\beta) = 2/7$ and $conf_{IS_D}(\alpha \rightarrow \beta) = \frac{freq_{IS_D}(\beta)}{freq_{IS_D}(\alpha)} = 2/5$.

2.5 Conclusions

This chapter has introduced a general framework for mining relational data. The structure for storing relational data in this framework is an information system that is constructed by adapting the notion of the standard information system. Information granules derived from the information system are used to construct relational patterns such as frequent patterns, association rules, and classification rules.

The introduced framework can be summarized as follows.

1. The framework can be helpful when a given database consists of many tables and some background objects are joined with the target ones through a number of tables. In this case, there arises the problem of how deeply one should search the database for background objects that are joined with the target ones. In the framework the search level can easily be changed so as to adjust the target object representation to a given data mining task.
2. The framework can also be useful when the search space limitation achieved by a language bias is not sufficient. The search space can additionally be limited since this is given as a set of information granules derived from the data.
3. The framework has an advantage over the ILP and RDB frameworks in terms of generation of patterns. Namely, the framework, unlike others, partially unifies the process of discovering patterns from data. This is done by constructing the search space based on information granules. The patterns can thus directly be obtained from such granules or constructed based on them.

Springer

http://www.springer.com/978-3-319-52750-5

Granular-Relational Data Mining
How to Mine Relational Data in the Paradigm of Granular Computing?

Hońko, P.

2017, XV, 123 p. 4 illus., Hardcover
ISBN: 978-3-319-52750-5