

A Multiview Approach for Intelligent Data Analysis based on Data Operators

Yaohua Chen and Yiyu Yao

*Department of Computer Science, University of Regina
Regina, Saskatchewan, Canada S4S 0A2
E-mail: {chen115y, yyao}@cs.uregina.ca*

Abstract

Multiview intelligent data analysis explores data from different perspectives to reveal various types of structures and knowledge embedded in the data. Each view may capture a specific aspect of the data and hence satisfy the needs of a particular group of users. Collectively, multiple views provide a comprehensive description and understanding of the data. In this paper, we propose a multiview framework of intelligent data analysis based on modal-style data operators. The classes of the data operators include basic set assignment, sufficiency, dual sufficiency, necessity and possibility operators. They demonstrate various types of data relationships and characterize various features and granulated views of the data. It is shown that different structures of the data can also be constructed based on the different data operators.

Key words: Multiview, Intelligent Data Analysis, Modal-style Data Operators, Concept Lattice, Granular Computing.

1 Introduction

Huge data sets and various data types lead to new types of problems and require the development of new types of techniques for modern intelligent data analysis [21]. An important objective of intelligent data analysis is to reveal and indicate diverse non-trivial features or views of a large amount of data. Many techniques and models in data mining, machine learning, pattern recognition, statistics, and other fields have been proposed. Each technique or model focuses on one particular view of the data and discovers a specific type of knowledge embedded in data.

In this paper, we propose a multiview approach that provides a unified framework for integrating multiple views of intelligent data analysis. It is easy to

know that an integrated and unified framework that allows a multiple view approach on the understanding, computation, and interpretation of data provides not only a platform to explore different aspects of the data, but also a tool to integrate many techniques of intelligent data analysis.

Modal-style data operators can be used to define, represent and analyze various types of data relations and structures [6,53]. Therefore, these modal-style data operators can provide a unified way to examine, characterize and construct different types of knowledge. In this paper, the multiview approach are defined based on modal-style data operators in a formal context.

The rest of the paper is organized as follows. In the next section, discussions about motivations and some related works are provided. Section 3 introduces formal contexts and modal-style data operators. In Section 4, we show that these data operators can be used to define different relations and granular structures of a universe. Section 5 examines several types of hierarchical structures defined based on the modal-style data operators. These granular and hierarchical structures are always be viewed as different types of knowledge embedded in the data set [11,6,39,53]. The conclusion of this study is given in Section 6.

2 Motivations and Related Work

In this section, we give motivations of this proposed research work and discuss some related work.

2.1 Motivations

Many techniques and models have been proposed for intelligent data analysis. In data mining, association rule mining is to detect the association relationship between two sets of items [1]. An association rule expresses that a customer buying one set of items tends to buy another set of items. Association rules can be considered as a description of association relationships among the data. In machine learning and pattern recognition, the techniques of classification and cluster learning focus on prediction of future data, plan future action, etc. [16,23,28]. They can be viewed as seeking for different types of relationships among the data. Each technique is designed with respect to the specific properties of the relationships [23]. Some techniques are proposed to detect the degree of similarity or commonality, such as template matching models [5,19], nearest neighbor models [13], kernel density models [40]. Some techniques are designed to construct decision boundaries or a partition based on commonality

of entities, such as decision tree [9,44,45], neural networks [2,3,35], and support vector machine [48]. Bayesian networks have been proposed and studied to classify entities into the class with respect to the maximum posterior probabilities of the entities [23,41]. Clustering is to directly partition unlabeled data based on some properties of the relationships among the data [22,27]. Many types of relationships can be used to determine which class the entities belong to, such as similarity, correlation, dependency, association, and so on [4]. Many techniques for data clustering have been proposed, such as K-means clustering [33], hierarchical clustering [26,46,49], nearest neighbor clustering [32], and density based clustering [15].

Most proposed techniques and models overemphasize the automation and efficiency of the data analysis systems, but neglect the effectiveness of the systems. To a large extent, the effectiveness determines the quality of the systems and satisfiability of human users [38]. There are two aspects of the effectiveness of data analysis systems. On the one hand, from a system or data set, a user is able to explore different types of knowledge, different features of data, and different interpretations of data with respect to different techniques, models, and user preferences. On the other hand, multiple aspects of data understanding, multiple angles of data summarizations or data descriptions, and multiple types of discovered knowledge are useful for satisfying a wide range of needs of a large diversity of users.

Therefore, a system providing multiple views of a data set is crucial and necessary for intelligent data analysis. In this paper, we propose an approach for multiview intelligent data analysis and demonstrate that one can discover different types of knowledge from a data set.

Modal-style data operators have been studied by many researchers for intelligent data analysis [11,14,17,18,50,51,54,55]. These data operators can be used to define, represent and analyze various types of relations and structures embedded in the data, which are generally viewed as different types of knowledge [6,53]. We use these modal-style data operators defined based on a formal context to examine, characterize and construct different types of data structures and knowledge.

For example, classifications of the universe induced by the relations produce various types of granular structures of the universe. Two types of structures, partition and covering, can be analyzed and discussed with respect to different modal-style data operators. Some hierarchical structures embedded in the data can also be defined based on modal-style data operators. Each hierarchical structure explores a particular organization of the data.

2.2 *Related Work*

Multiple view strategies and approaches have been proposed and studied in many fields, including social sciences, software design, machine learning, and data analysis [7,8,24,34,62]

Jeffries and Ransford argued that the study of social stratification should not focus on isolated and fragmented views, but holistic and unified views [24]. The study of social stratification identifies and investigates the characteristics of social societies. It focuses primarily on the hierarchies of social classes. The traditional single hierarchy approach based on social classes limits one's understanding of the complexities of modern societies. The authors proposed a multiple hierarchy model to integrate class, ethnicity, gender, and age social inequality hierarchies. These four hierarchies are considered as separated but interrelated. The study shows that a multiple hierarchy approach could increase one's understanding and be more comprehensive and valid for studying social inequality.

Belkhouche and Lemus-Olalde studied the formal foundations of an abstract interpretation of multiple views at the software design stage [7,8]. In the process of modeling a system, the designer always generates a set of designs, such as functional, behavioral, structural and data designs. Each design focuses on a view that describes a subset of relevant features of a system and is expressed by one or more notations. Different designs show different viewpoints of the system. Functional designs describe what the system does in terms of its tasks. Behavioral designs describe the causal links between events and system responses during execution. Structural designs concern the essentially static aspects of the system. Data designs concern the data used in system and relationships between them. The authors argued that a multiple view analysis framework can be used to systematically compare, identify and analyze the discrepancies among different views, enhance design quality and provide a multi-angled understanding of a problem or a project.

Several researchers proposed frameworks of multistrategy learning to integrate a wide range of learning strategies [34]. They argued that the research on multistrategy learning systems is significantly relevant to study on human learning since human learning is clearly multistrategy. Multistrategy systems have a potential to be more versatile and more powerful to solve a much wider range of learning problems than monostrategy systems. Some conceptual frameworks, such Inferential Theory of Learning and Plausible Justification Trees, are proposed to investigate the logical capabilities and relationships of various learning methods and processes. Based on the proposed frameworks, methodologies and applications for multistrategy task-adaptive learning have been developed [34].

Zhong *et al.* proposed an approach of multiple aspect analysis of human brain data for investigating human multiperception mechanism [61,62]. By using different methods and measurements, they can obtain various types of human brain data. They observed that each type of data extracted by a particular method or measure matches human ability in a specific aspect. Each approach to analyze the obtained data is also processed with respect to the specific state or the related part of a stimulus. They argued that every method or model of data acquisition and analysis has its own strength and weakness. Thus, designing psychological and physiological experiments for obtaining various data from human multiperception mechanism and analyzing such data from multiple aspects for discovering new models of human multiperception are the key issues. They proposed a framework of data-mining grid for multiple human brain data analysis by cooperatively using cognitive and data-mining techniques, and attempted to change the perspective of cognitive scientists from a single type of experimental data analysis toward a holistic view.

In data mining, some researchers have already recognized that integration of association rule mining and classification can produce a more accurate classifier [31,58,63]. Approaches on integrating clustering and classification techniques can also achieve better classification or clustering results [20,25,29,60].

3 Formal Contexts and Modal-style Data Operators

In this section, definitions for formal contexts and modal-style data operators are given. Some connections between modal-style data operators are investigated.

3.1 Formal Contexts

We assume that a data set is given in terms of a formal context [50] or a binary table [39]. It provides a convenient way to describe a finite set of objects by a finite set of attributes. For simplicity, we only consider a finite set of objects and a finite set of attributes in this paper. The results may not be true for the infinite cases.

Let U and V be any two finite sets. Elements of U are called objects and elements of V are called attributes or properties. The relationships between objects and attributes are described by a binary relation R between U and V , which is a subset of the Cartesian product $U \times V$. For a pair of elements $x \in U$ and $y \in V$, if $(x, y) \in R$, written as xRy , we say that x has the attribute y , or the attribute y is possessed by object x . The triplet (U, V, R) is

Table 1
 A Formal Context From [47]

	PO	LM	SM	VM	VC	AT	CS	DM	IC	AD
1. Picture Completion	✓	✓		✓						✓
2. Picture Arrangement	✓					✓	✓	✓	✓	✓
3. Block Design	✓				✓	✓		✓		✓
4. Object Assembly	✓				✓	✓		✓		✓
5. Digit Symbol			✓	✓	✓					

PO: Perceptual Organization, LM: Long-term Memory, SM: Short-term Memory,
 VM: Visual Memory, VC: Visual-motor Coordination, AT: Abstract Thought,
 CS: Common Sense, DM: Decision Making, IC: Intellectual Curiosity, AD: Attention to Detail.

called a formal context [17,50] or a binary information table [39]. In general, a multi-valued formal context can be transformed into a binary formal context through scaling [17]. That is, every information table can be represented by a formal context [10].

Example 1 *Table 1 is an example of a formal context taken from [47]. The attributes describe the abilities of human brain. The objects represent the realistic actions that need human brain to perform. A binary relation determine which actions should be performed by which abilities. Thus, objects and attributes in a formal context are described and determined by each other based on the binary relation.*

Based on the binary relation R , we can associate a set of attributes with an object. An object $x \in U$ has the set of attributes:

$$xR = \{y \in V \mid xRy\} \subseteq V.$$

The set of attributes xR can be viewed as a description of the object x . In other words, object x is described or characterized by the set of attributes xR . Similarly, an attribute y is possessed by the set of objects:

$$Ry = \{x \in U \mid xRy\} \subseteq U.$$

By extending these notations, we can establish relationships between sets of objects and sets of attributes. This leads to two types of data operators, one from 2^U to 2^V and the other from 2^V to 2^U , where 2^U and 2^V are the power sets of U and V , respectively.

3.2 Modal-style Data Operators

With a formal context, we can define different types of modal-style data operators. Each of them leads to a different type of rules summarizing the relationships between objects and attributes.

Basic Set Assignments: For a set of objects $A \subseteq U$ and a set of attributes $B \subseteq V$, we can define a pair data operators, ${}^b : 2^U \rightarrow 2^V$ and ${}^b : 2^V \rightarrow 2^U$, called basic set assignments, as follows [52]:

$$\begin{aligned} A^b &= \{y \in V \mid Ry = A\}, \\ B^b &= \{x \in U \mid xR = B\}. \end{aligned}$$

For simplicity and clarity, the same symbol is used for both operators. The operators b associate a set of objects with a set of attributes and a set of attributes with a set of objects. They satisfy the following properties: for $A, A_1, A_2 \subseteq U$,

- (b1) $\emptyset^b = \emptyset$,
- (b2) $V = \bigcup \{A^b \mid A \subseteq U\}$,
- (b3) $A_1 \neq A_2 \implies A_1^b \cap A_2^b = \emptyset$.

Property (b1) states that the assigned set for an empty set is the empty set. It should be pointed out that property (b1) holds only if the formal context satisfies the condition: $Ry \neq \emptyset$, for all $y \in V$. Property (b2) shows that the union of assigned sets of attributes for all subsets of objects is the entire set of attributes V . Property (b3) means the assignments for different subsets of objects are distinct and non-overlapping. Properties (b2) and (b3) hold for any binary relation. The property (b1) holds if an object must have an attribute or an attribute must be possessed by an object.

Sufficiency Operators: For a set of objects $A \subseteq U$ and a set of attributes $B \subseteq V$, we can define a pair data operators, ${}^* : 2^U \rightarrow 2^V$ and ${}^* : 2^V \rightarrow 2^U$, called sufficiency operators, as follows [17,50]:

$$\begin{aligned} A^* &= \{y \in V \mid A \subseteq Ry\} \\ &= \bigcap_{x \in A} xR, \end{aligned} \tag{1}$$

$$\begin{aligned} B^* &= \{x \in U \mid B \subseteq xR\} \\ &= \bigcap_{y \in B} Ry. \end{aligned} \tag{2}$$

The same symbol is used for both operators. By definition, $\{x\}^* = xR$ is the set of attributes possessed by the object x , and $\{y\}^* = Ry$ is the set of objects having attribute y . For a set of objects A , A^* is the maximal set of properties shared by all objects in A . Similarly, for a set of attributes B , B^* is the maximal set of objects that have all attributes in B .

Dual Sufficiency Operators: For a set of objects $A \subseteq U$ and a set of attributes $B \subseteq V$, we can define a pair data operators, $\# : 2^U \rightarrow 2^V$ and $\# : 2^V \rightarrow 2^U$, called dual sufficiency operators, as follows [55]:

$$\begin{aligned} A^\# &= \{y \in V \mid A \cup Ry \neq U\}, \\ B^\# &= \{x \in U \mid B \cup xR \neq V\}. \end{aligned}$$

The same symbol is again used for both operators. For a subset $A \subseteq U$, an attribute in $A^\#$ is not possessed by at least one object not in A .

The following properties hold for the sufficiency operators [17,50]: for $A, A_1, A_2 \subseteq U$,

- (s1) $\emptyset^* = U$,
- (s2) $A^* = A^{c\#c}$,
- (s3) $A^{***} = A^*$,
- (s4) $A_1 \subseteq A_2 \implies A_1^* \supseteq A_2^*$,
- (s5) $(A_1 \cup A_2)^* = A_1^* \cap A_2^*$.

where A^c presents the complement of the set of A . The dual sufficiency operators have the following properties [55]: for $A, A_1, A_2 \subseteq U$,

- (d1) $U^\# = \emptyset$,
- (d2) $A^\# = A^{c*c}$,
- (d3) $A^{\#\#\#} = A^\#$,
- (d4) $A_1 \subseteq A_2 \implies A_1^\# \subseteq A_2^\#$,
- (d5) $(A_1 \cap A_2)^\# = A_1^\# \cup A_2^\#$,

The properties (s1) and (d1) mean that the operation $*$ on the empty set will be the universe, and the operation $\#$ on the universe will be the empty set. The properties (s2) and (d2) show that the operators $*$ and $\#$ are dual to each other. Property (s4) shows that the operator $*$ is monotonic decreasing with respect to set inclusion while property (d4) shows that the operator $\#$ is monotonic increasing. Similar properties hold for the operators $*$ and $\#$ on sets of attributes.

A pair of dual mappings, $*$: $2^U \rightarrow 2^V$ and $*$: $2^V \rightarrow 2^U$, is indeed a Galois connection between two partially ordered sets $\langle 2^U, \subseteq \rangle$ and $\langle 2^V, \subseteq \rangle$ [14,51].

Necessity Operators: For a set of objects $A \subseteq U$ and a set of attributes $B \subseteq V$, we can define a pair data operators, \square : $2^U \rightarrow 2^V$ and \square : $2^V \rightarrow 2^U$, called necessity operators, as follows [14,18,54,55]:

$$\begin{aligned} A^\square &= \{y \in V \mid Ry \subseteq A\}, \\ B^\square &= \{x \in U \mid xR \subseteq B\}. \end{aligned}$$

The same symbol is used for both operators. By definition, an object having an attribute in A^\square is necessarily in A . The operators are referred to as the necessity operators.

Possibility Operators: For a set of objects $A \subseteq U$ and a set of attributes $B \subseteq V$, we can define a pair data operators, \diamond : $2^U \rightarrow 2^V$ and \diamond : $2^V \rightarrow 2^U$, called possibility operators, as follows [14,18,54,55]:

$$\begin{aligned} A^\diamond &= \{y \in V \mid Ry \cap A \neq \emptyset\} \\ &= \bigcup_{x \in A} xR, \\ B^\diamond &= \{x \in U \mid xR \cap B \neq \emptyset\} \\ &= \bigcup_{y \in B} Ry. \end{aligned}$$

The same symbol is again used for both operators. An object having an attribute in A^\diamond is only possibly in A . The operators are referred to as the possibility operators.

The necessity operators have the following properties [54]: for $A, A_1, A_2 \subseteq U$,

- (n1) $U^\square = V$,
- (n2) $A^\square = A^{c \diamond c}$,
- (n3) $A_1 \subseteq A_2 \implies A_1^\square \subseteq A_2^\square$,
- (n4) $(A_1 \cap A_2)^\square = A_1^\square \cap A_2^\square$.

The possibility operators have the following properties [54]: for $A, A_1, A_2 \subseteq U$,

- (p1) $U^\diamond = V$,
- (p2) $A^\diamond = A^{c \square c}$,
- (p3) $A_1 \subseteq A_2 \implies A_1^\diamond \subseteq A_2^\diamond$,
- (p4) $(A_1 \cup A_2)^\diamond = A_1^\diamond \cup A_2^\diamond$.

The properties (n1) and (p1) state that the universe U are mapped to the entire set of attribute V . The properties (n2) and (p2) show that the operators \square and \diamond are dual to each other. The properties (n3) and (p3) say that these two operators are monotonic increasing. Property (n4) states that the operator \square is distributive over set intersection \cap , while property (p4) states that the operator \diamond is distributive over set union \cup .

3.3 Connections between Data Operators

Since the modal-style data operators are defined based on the binary relation R of a formal context, there exist connections among these modal-style data operators [55]. In terms of the basic set assignment, we can re-express operators $*$, \square and \diamond as:

- (1) $A^* = \bigcup\{X^b \mid X \subseteq U, A \subseteq X\}$,
- (2) $A^\# = \bigcup\{X^b \mid X \subseteq U, X \cup A \neq U\}$,
- (3) $A^\square = \bigcup\{X^b \mid X \subseteq U, X \subseteq A\}$,
- (4) $A^\diamond = \bigcup\{X^b \mid X \subseteq U, A \cap X \neq \emptyset\}$.

In other words, the sufficiency, dual sufficiency, necessity and possibility operations on a set A can be expressed by a union of some particular basic assigned sets for A .

Conversely, one can re-express operator b by using operators $*$ and \square as:

- (5) $A^b = A^* \cap A^\square$,
- (6) $A^b = A^* - \bigcup\{X^* \mid A \subset X\}$,
- (7) $A^b = A^\square - \bigcup\{X^\square \mid X \subset A\}$.

The basic set assignment on a set A is the intersection of sufficiency and necessity operations on A . Moreover, for a set A , the result of removing the union of the sufficiency operations on all proper subset of A from the sufficiency operation on set A is the basic set assignment on A . Similarly, the result of removing the union of the necessity operations on all proper superset of A from the necessity operation on set A is also the basic set assignment on A .

Example 2 *A simple example is provided to illustrate the semantic meanings of each modal-style data operator.*

The formal context in Table 1 describes the relationships between actions and human brain abilities. In this table, the objects represent the actions, such

as picture completion, picture arrangement, block design, object assembly and digit symbol. The attributes represent the brain abilities needed to complete the actions.

The relation xRy , where $x \in U$ and $y \in V$ is interpreted as “Action x is performed by using the brain ability y ”. Suppose A denotes a behavior, a set of actions, and y denotes a brain ability. The data operators can be interpreted as follows:

- $y \in A^b \iff$ The ability y is needed for performing exactly the set of actions A ,
- $y \in A^* \iff$ The ability y is sufficient to perform all actions in A ,
- $y \in A^\# \iff$ Not all actions in A^c can be performed by y ,
- $y \in A^\square \iff$ The actions performed by using y are necessarily in A ,
- $y \in A^\diamond \iff$ The ability y is possibly used to perform some action in A .

Suppose B denotes a set of abilities, and x denotes an action. The data operators can be interpreted as follows:

- $x \in B^b \iff$ The action x is exactly performed by all abilities in B ,
- $x \in B^* \iff$ The action x is sufficiently performed by all abilities in B ,
- $x \in B^\# \iff$ Not all abilities in B^c can perform x ,
- $x \in B^\square \iff$ The ability to perform the action x is necessarily in B ,
- $x \in B^\diamond \iff$ The action x is possibly performed by some ability in B .

Based on these modal-style data operators, one can compose new data operators to express various features of the data. In the next two sections, we will show that the modal-style data operators can not only be used to define various types of relations between objects and induced granular structures of the universe, but also be used to define different types of hierarchical structures of the data.

4 Granular Structures of the Universe

In this section, we investigate various relations between objects and granular structures induced by the relations based on modal-style data operators.

4.1 Granulation and Granular structure

Granular computing has emerged as a multi-disciplinary study of problem solving and information processing [6,30,36,39,42,43,53,59]. It provides a general, systematic and natural way to analyze, understand, represent, and solve real world problems. Granulation and granular structure of the universe are fundamental issues in granular computing. Granular structures are used to classify and characterize the data [53,59].

Granulation deals with the grouping of individual elements of a universe into classes or sets called granules. A family of granules provides a granulated view of the universe. Elements in a granule are grouped together by indistinguishability, similarity, proximity or functionality [36,59]. There are both semantics and algorithmic aspects of granulation. A semantic interpretation explains why two objects are put into the same granule and an algorithm determines how two objects are related with each other. The granules in a granulated view can be either disjoint or overlapping and are regarded as a level of understanding the subject matter of the study. The different granulated views can be viewed as different granular structures of a universe. Generally, there are two types of granulations, and the corresponding granular structures are partition and covering of the universe.

A simple way for the granulation of a universe U is through a binary relation \mathfrak{R} on U . A binary relation \mathfrak{R} over a universe U is a subset of the Cartesian product $U \times U$. Based on \mathfrak{R} , a 1-neighborhood system of objects can be constructed [56]. For two objects $x, x' \in U$, if $x\mathfrak{R}x'$, we say that x is a predecessor of x' , and x' is a successor of x . The set of predecessors of x is called predecessor neighborhood and denoted by $\mathfrak{R}x = \{x' \in U \mid x'\mathfrak{R}x\}$, and the set of successors of x is called successor neighborhood and denoted by $x\mathfrak{R} = \{x' \in U \mid x\mathfrak{R}x'\}$. A binary relation \mathfrak{R} on U is called an equivalence relation if it is reflexive, symmetric and transitive:

- Reflective: for all $x \in U$, $x\mathfrak{R}x$,
- Symmetric: for all $x \in U$, $x\mathfrak{R}x' \implies x'\mathfrak{R}x$,
- Transitive: for all $x \in U$, $[x\mathfrak{R}x', x'\mathfrak{R}x''] \implies x\mathfrak{R}x''$.

An equivalence relation partitions the universe U into a family of disjoint subsets. This partition of the universe is denoted by U/E [57]. The elements of the partition U/E are called granules and satisfy the following properties:

- (I) $X_i \cap X_j = \emptyset$, where $X_i, X_j \in U/E, i \neq j$,
- (II) $U = \bigcup\{X \mid X \in U/E\}$.

Conversely, given a partition of the universe, we can construct uniquely an equivalence relation that produces the same partition. There is a one-to-one correspondence between partitions of a universe and equivalence relations on the universe.

The notion of a covering of a universe is a generalization of a partition by removing the non-overlapping condition (I). Let \mathbf{C} be a family of subsets of U . It is called a covering of U if the following condition holds:

$$(III) \quad U = \bigcup \{X \mid X \in \mathbf{C}\}.$$

Unlike the case of equivalence relations, we cannot establish a one-to-one correspondence between coverings and a class of binary relations. Nevertheless, for a reflexive relation \mathfrak{R} , the family of successor neighborhoods $\{x\mathfrak{R} \mid x \in U\}$ is a covering of the universe.

The binary relations between objects in turn can induce different granular structures. Modal-style data operators can be used to define different relations between objects.

4.2 Granular Structures Induced by Modal-style Operators

Many types of binary relations between objects have been studied [37,56]. The relations between objects can be defined based on the relations between their descriptions, i.e. sets of attributes. In other words, one can use the relations between objects and attributes, $R : U \times V$, to define the relations between objects, $\mathfrak{R} : U \times U$. In this paper, we consider four types of relations: equivalence relation, partial order relation, similarity relation and negative similarity relation.

Equivalence Relation: Two objects may be viewed as being equivalent if they have the same description, i.e. they share the same set of attributes [39]. An equivalence relation can be defined by: for $x, x' \in U$,

$$x \equiv_U x' \iff xR = x'R.$$

With equivalence relation \equiv_U , an object has the same predecessor and successor neighborhood. For an object $x \in U$, the set of objects that are equivalent to x is called an equivalence class of x and defined by:

$$\equiv_U x = \{x' \in U \mid x' \equiv_U x\},$$

$$\begin{aligned}
 &= \{x' \in U \mid x \equiv_U x'\}, \\
 &= x \equiv_U, \\
 &= [x].
 \end{aligned}$$

The equivalence classes can be re-expressed by using modal-style data operators. In a formal context (U, V, R) , for an object $x \in U$, its equivalence class can be re-expressed by:

$$\begin{aligned}
 [x] &= \{x' \in U \mid x' \equiv_U x\}, \\
 &= \{x' \in U \mid xR = x'R\}, \\
 &= \{x' \in U \mid x'R = \{x\}^*\}, \\
 &= \{x\}^{*b}.
 \end{aligned}$$

The objects in $[x]$ are considered to be indistinguishable from x . One is therefore forced to consider $[x]$ as a whole. In other words, under an equivalence relation, equivalence classes are the smallest non-empty observable, measurable, or definable subsets of U .

A subset of objects $A \subseteq U$ is called a focal set of objects if $A^b \neq \emptyset$. Similarly, we can define a focal set of attributes. By the properties of the basic set assignment, the partition U/E is indeed the assignments of the focal sets of attributes:

$$U/E \equiv_U = \{Y^b \neq \emptyset \mid Y \subseteq V\}.$$

This can be easily seen from the fact that each object $x \in Y^b$ possesses exactly the set of attribute Y .

Similarity Relation: If two objects x and x' have some overlapping attributes, they are regarded as being similar to each other. This type of relation is defined by [37]: for $x, x' \in U$,

$$x \approx_U x' \iff x'R \cap xR \neq \emptyset.$$

The relation is reflexive and symmetric, but not necessarily transitive.

Based on the similarity relation \approx_U , for an object, its predecessor and successor neighborhoods are the same because of the symmetric property of \approx_U . In a formal context (U, V, R) , for an object $x \in U$, the \approx_U predecessor and successor neighborhood are given by:

$$\begin{aligned}
 \approx_U x &= x \approx_U, \\
 &= \{x' \in U \mid x \approx_U x'\},
 \end{aligned}$$

$$\begin{aligned}
 &= \{x' \in U \mid x' \approx_U x\}, \\
 &= \{x' \in U \mid x'R \cap xR \neq \emptyset\}, \\
 &= \{x\}^{\diamond\diamond}.
 \end{aligned}$$

The objects in $\{x\}^{\diamond\diamond}$ must share some attributes of x . The union of all such sets of objects is the universe U , that is, $U = \bigcup\{\{x\}^{\diamond\diamond} \mid x \in U\}$. Thus, the family of all sets of objects induced by similarity relation \approx_U is a covering of the universe U .

Negative Similarity Relation: For two objects x and x' , if the union of their attributes is not the whole set of attributes, we can consider them as being negatively similar [37]. This type of relation is defined by: for $x, x' \in U$,

$$\begin{aligned}
 x \succsim_U x' &\iff xR^c \cap x'R^c \neq \emptyset, \\
 &\iff xR \cup x'R \neq V,
 \end{aligned}$$

where R^c is the complement of the relation R . The negative similarity relation is only symmetric.

For the negative similarity relation \succsim_U , the predecessor and successor neighborhoods for an object are the same. For an object $x \in U$, the \succsim_U predecessor and successor are:

$$\begin{aligned}
 \succsim_U x &= x \succsim_U, \\
 &= \{x' \in U \mid x \succsim_U x'\}, \\
 &= \{x' \in U \mid x' \succsim_U x\}, \\
 &= \{x' \in U \mid xR \cup x'R \neq V\}, \\
 &= \{x\}^{*\#}.
 \end{aligned}$$

For an object x' in $\{x\}^{*\#}$, there must exist at least one attribute that is not possessed by both x and x' . The family of all sets of objects $\{x\}^{*\#}$ induced by the negative similarity relation \succsim_U may not be necessarily a partition or covering of the universe U .

Partial Order Relation: A partial order relation on objects can be defined based on set inclusion: for $x, x' \in U$,

$$x \preceq_U x' \iff xR \subseteq x'R.$$

It is reflexive and transitive.

A partial order relation is not necessarily symmetric. In this case, we have different predecessor and successor neighborhoods for an object. For an object $x \in U$, the \preceq_U successor neighborhood is given by:

$$\begin{aligned}
 x \preceq_U &= \{x' \in U \mid x \preceq_U x'\}, \\
 &= \{x' \in U \mid xR \subseteq x'R\}, \\
 &= \{x' \in U \mid \{x\}^* \subseteq x'R\}, \\
 &= \{x\}^{**}.
 \end{aligned}$$

For an object $x \in U$, the \preceq_U predecessor neighborhood is given by:

$$\begin{aligned}
 \preceq_U x &= \{x' \in U \mid x' \preceq_U x\}, \\
 &= \{x' \in U \mid x'R \subseteq xR\}, \\
 &= \{x' \in U \mid x'R \subseteq \{x\}^\diamond\}, \\
 &= \{x\}^{\diamond\Box}.
 \end{aligned}$$

The set $\{x\}^{**}$ is the *maximal* set of objects that have *all* attributes in $\{x\}^*$. In other words, any objects in $\{x\}^{**}$ has at least all attributes of x . In fact, each set $\{x\}^{**}$ is the extension of an object concept in formal concept analysis [17,50]. The union of all such sets of objects is the universe U , that is, $U = \bigcup\{\{x\}^{**} \mid x \in U\}$. Thus, the family of all sets of objects $\{x\}^{**}$ induced by the partial order relation \preceq is a covering the universe U . The set $\{x\}^{\diamond\Box}$ is the *maximal* set of objects whose attributes are subsets of $\{x\}^\diamond$. In other words, any object in $\{x\}^{\diamond\Box}$ has at most all attributes of x . The union of all such sets of objects is the universe U , that is, $U = \bigcup\{\{x\}^{\diamond\Box} \mid x \in U\}$. Thus, the family of all sets of objects $\{x\}^{\diamond\Box}$ induced by the partial order relation \preceq_U is a covering the universe U .

4.3 Connections between Granular Structures

These types of data operators induce different types of granulated views of the universe. They can be considered as different types of strategies to divide the universe or classify the objects.

The connections among various types of granular structures can be established based on the connections of the data operators. In fact, the equivalence classes can be used to re-express other types of sets of objects.

$$\begin{aligned}
 \{x\}^{**} &= \bigcup\{[x'] \mid xR \subseteq x'R\}, \\
 \{x\}^{*\#} &= \bigcup\{[x'] \mid xR \cup x'R \neq V\}, \\
 \{x\}^{\diamond\Box} &= \bigcup\{[x'] \mid xR \supseteq x'R\}, \\
 \{x\}^{\diamond\Diamond} &= \bigcup\{[x'] \mid xR \cap x'R \neq \emptyset\}.
 \end{aligned}$$

Similar to the granulations on objects, we can also define the various granulations on attributes by using modal-style data operators. The detailed definitions and formulas of these granulations on attributes are given in the Appendix.

5 Hierarchical Structures Embedded in Data

In the classical view, a concept is determined by its extension and intension. The extension consists of all objects belonging to the concept. The intension includes all attributes that are used to measure the objects belonging to the concept. A pair of extension and intension is considered as a representation of a concept. Lattice theory provides a proper vocabulary for hierarchical structures and can be applied to construct concept lattices [17,50,54,55].

In this section, the modal-style data operators are examined to construct different types of concept lattices. Several types of concept lattices are derived from a same formal context and reflect various features of the data and to represent a diversity of knowledge embedded in the data.

5.1 Class-oriented Formal Concept Lattice

An equivalence relation between objects can induce a partition of the universe. Each element in the partition is called an elementary definable set [39]. One can construct a definable set by taking a union of some equivalence classes. The family of all definable sets on the universe U is an σ -algebra $\sigma(U/\equiv_U) \subseteq 2^U$ with basis U/\equiv_U , where 2^U is the power set of U [57]. The following properties hold:

- (E1) $\emptyset \in \sigma(U/\equiv_U)$,
- (E2) $U \in \sigma(U/\equiv_U)$,
- (E3) $X \in \sigma(U/\equiv_U) \Rightarrow X^c \in \sigma(U/\equiv_U)$,
- (E4) $A_1, A_2 \in \sigma(U/\equiv_U) \Rightarrow A_1 \cap A_2 \in \sigma(U/\equiv_U)$,
- (E5) $A_1, A_2 \in \sigma(U/\equiv_U) \Rightarrow A_1 \cup A_2 \in \sigma(U/\equiv_U)$.

The properties (E1) and (E2) state that the empty set \emptyset and the universe U are in $\sigma(U/\equiv_U)$. The properties (E3) - (E5) express that the complement, intersection and union of some definable sets in $\sigma(U/\equiv_U)$ are still definable sets in $\sigma(U/\equiv_U)$. In other words, $\sigma(U/\equiv_U)$ is closed under set complement, intersection and union operations. Similarly, the family of all definable sets on the universe V is also an σ -algebra $\sigma(U/\equiv_U) \subseteq 2^U$ with basis V/\equiv_U .

Example 3 In Table 1, the equivalence classes over the universe of objects are:

- | | |
|-------------------------|---------------------------------------------------------|
| $[1] = \{1\},$ | assigned to the focal set $\{PO, LM, VM, AD\},$ |
| $[2] = \{2\},$ | assigned to the focal set $\{PO, AT, CS, DM, IC, AD\},$ |
| $[3] = [4] = \{3, 4\},$ | assigned to the focal set $\{PO, VC, AT, DM, AD\},$ |
| $[5] = \{5\},$ | assigned to the focal set $\{SM, VM, VC\}.$ |

The equivalence classes over the universe of attributes are:

- | | |
|-----------------------------|---------------------------------------------|
| $[PO] = [AD] = \{PO, AD\},$ | assigned to the focal set $\{1, 2, 3, 4\},$ |
| $[AT] = [DM] = \{AT, DM\},$ | assigned to the focal set $\{2, 3, 4\},$ |
| $[CS] = [IC] = \{CS, IC\},$ | assigned to the focal set $\{2\},$ |
| $[LM] = \{LM\},$ | assigned to the focal set $\{1\},$ |
| $[SM] = \{SM\},$ | assigned to the focal set $\{5\},$ |
| $[VM] = \{VM\},$ | assigned to the focal set $\{1, 5\},$ |
| $[VC] = \{VC\},$ | assigned to the focal set $\{3, 4, 5\}.$ |

A pair (A, B) , $A \subseteq U, B \subseteq V$, is called a class-oriented formal concept if $A \in \sigma(U/\equiv_U)$ and $B = A^*$. The set of objects A is called the extension of the concept (A, B) , and the set of attributes B is called the intension.

For two class-oriented concepts (A_1, B_1) and (A_2, B_2) , we say that (A_1, B_1) is a sub-concept of (A_2, B_2) , and (A_2, B_2) is a super-concept of (A_1, B_1) , if and only if $A_1 \subseteq A_2$.

The family of all class-oriented formal concepts forms a complete lattice called class-oriented formal concept lattice. This lattice gives a hierarchical structure of the elements in $\sigma(U/\equiv_U)$ and their corresponding attributes. The meet \wedge and the join \vee are defined by:

$$\begin{aligned} (A_1, B_1) \wedge (A_2, B_2) &= ((A_1 \cap A_2), (A_1 \cap A_2)^*), \\ (A_1, B_1) \vee (A_2, B_2) &= ((A_1 \cup A_2), (B_1 \cap B_2)). \end{aligned}$$

Example 4 The lattice derived from Table 1 is illustrated in Figure 1 in which each rectangle represents a class-oriented formal concept, i.e., a definable set in the system of $\sigma(U/\equiv)$ and its correspondent attributes. The lines associating the rectangles is considered as the partial order relations between class-oriented formal concepts.

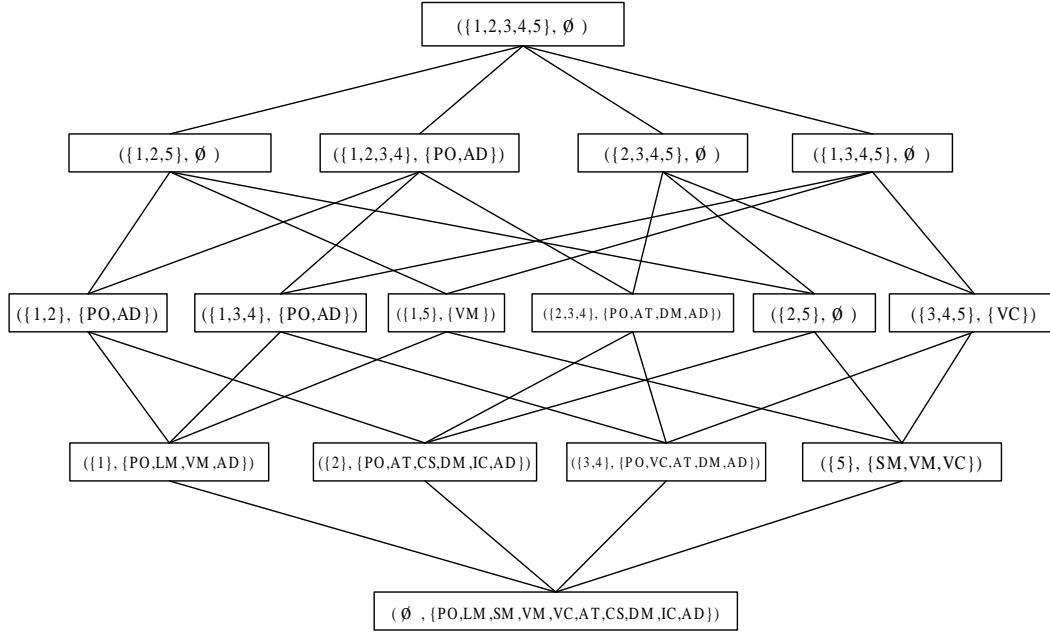


Fig. 1. A Class-oriented Formal Concept Lattice

5.2 Formal Concept Lattice

Wille and Ganter use efficiency operators to define a lattice of formal concepts [17,50]. A pair (A, B) , $A \subseteq U, B \subseteq V$, is called a formal concept if $A = B^*$ and $B = A^*$. The set of objects A is referred to as the extension of the concept (A, B) , and the set of attributes B is referred to as the intension. Objects in A share all attributes B , and only attributes in B are possessed by all objects in A .

For two formal concepts (A_1, B_1) and (A_2, B_2) , we say that (A_1, B_1) is a sub-concept of (A_2, B_2) , and (A_2, B_2) is a super-concept of (A_1, B_1) , if and only if $A_1 \subseteq A_2$, or equivalently, if and only if $B_2 \subseteq B_1$.

The family of all formal concepts is a complete lattice called a formal concept lattice [17,50]. The meet \wedge and the join \vee in formal concept lattice are defined by:

$$\begin{aligned} (A_1, B_1) \wedge (A_2, B_2) &= (A_1 \cap A_2, (B_1 \cup B_2)^{**}), \\ (A_1, B_1) \vee (A_2, B_2) &= ((A_1 \cup A_2)^{**}, B_1 \cap B_2). \end{aligned}$$

Example 5 Figure 2 is the formal concept lattice formed based on Table 1. Each rectangle represents a formal concept in the lattice. Lines between rectangles represent the partial order relations between the concepts.

Consider an operator $^{**} : 2^U \rightarrow 2^U$, which is closure operator on 2^U [14,17,18].

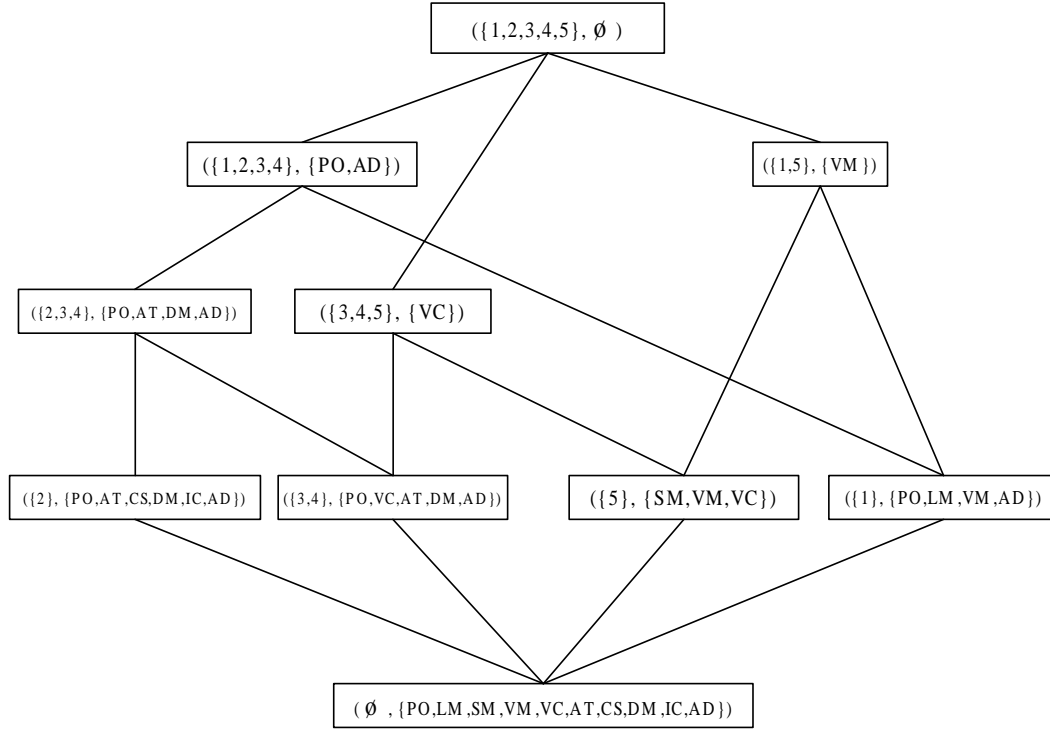


Fig. 2. A Formal Concept Lattice

The properties of the operator $**$ are: for $A, A_1, A_2 \subseteq U$,

- (SI) $U^{**} = U$,
- (SII) $A \subseteq A^{**}$,
- (SIII) $A^{**} = A^{****}$,
- (SIV) $A_1 \subseteq A_2 \implies A_1^{**} \subseteq A_2^{**}$.

Property (SI) states that the operation $**$ on the universe U will be the universe itself. Property (SII) and (SIV) mean that $**$ is a monotonically increasing operator. Property (SIII) shows that $**$ is an idempotent operator. The operator $** : 2^V \rightarrow 2^V$ is a closure operator on 2^V and has similar properties of the operator $** : 2^U \rightarrow 2^U$ over the universe U .

Let $(U, **)$ denote the family of sets $(U, **) = \{A^{**} \mid A \subseteq U\}$. In fact, $(U, **) \subseteq 2^U$ containing the universe U is the set of the extensions of all formal concepts. The system $(U, **)$ is closed under the operator $**$ and set intersection operator, but not set union operator. That is, the system $(U, **)$ is a closure system defined by the following properties [12]:

- (S1) $U \in (U, **)$,
- (S2) $A_1, A_2 \in (U, **) \implies A_1 \cap A_2 \in (U, **)$.

In fact, the system $(U, **)$ consists of all fixed points of operator $**$, that is,

$(U, **) = \{A^{**} = A \mid A \subseteq U\}$. Furthermore, for a non-empty set $A \in (U, **)$ and the corresponding formal concept (A, A^*) , we have [17]:

$$(A, A^*) = \bigvee \{(\{x\}^{**}, \{x\}^*) \mid x \in A\}.$$

That is, $\{x\}^{**}$ can be used to re-express the sets in the system $(U, **)$. The whole system $(U, **)$ can be generated by the sets of $\{x\}^{**}$, $x \in U$. In fact, a pair $(\{x\}^{**}, \{x\}^*)$, $x \in U$, is a formal concept called an object concept [17]. $\{x\}^{**}$ is the extension of an object concept. A formal concept in the lattice can be generated, composed or re-expressed by some object concepts. In other words, the whole formal concept lattice can be generated by object concepts.

5.3 Dual Formal Concept Lattice

Similar to the efficiency operators for formal concept lattice, the dual efficiency operators can be used to define a lattice of dual formal concepts. A pair (A, B) , $A \subseteq U, B \subseteq V$, is called a dual formal concept if $A = B^\#$ and $B = A^\#$. The set of objects A is referred to as the dual extension of the concept (A, B) , and the set of attributes Y is referred to as the dual intension. It can be easily seen that a pair (A, B) is a dual formal concept if and only if (A^c, B^c) is a standard formal concept. For a dual concept (A, B) , only objects not in A share all attributes not in B , and only attributes not in B are possessed by all objects not in A .

For two dual formal concepts (A_1, B_1) and (A_2, B_2) , we say that (A_1, B_1) is a sub-concept of (A_2, B_2) , and (A_2, B_2) is a super-concept of (A_1, B_1) , if and only if $A_1 \subseteq A_2$, or equivalently, if and only if $B_2 \subseteq B_1$.

The family of all dual formal concepts forms a complete lattice called dual formal concept lattice. The meet \wedge and the join \vee in dual formal concept lattice are defined by:

$$\begin{aligned} (A_1, B_1) \wedge (A_2, B_2) &= ((A_1 \cap A_2)^{\#\#}, (B_1 \cup B_2)), \\ (A_1, B_1) \vee (A_2, B_2) &= ((A_1 \cup A_2), (B_1 \cap B_2)^{\#\#}). \end{aligned}$$

Because of the property of duality of the operators $*$ and $\#$, one can re-express one operator with the other operator as $\# = c^*c$ and hence $\#\# = c^{**}c$. In other words, a dual formal concept can be obtained by taking the set complement of the extension and intension of the formal concept.

Example 6 A dual formal concept lattice derived from Table 1 is illustrated in Figure 3. Each rectangle represents a dual formal concept. A line describes

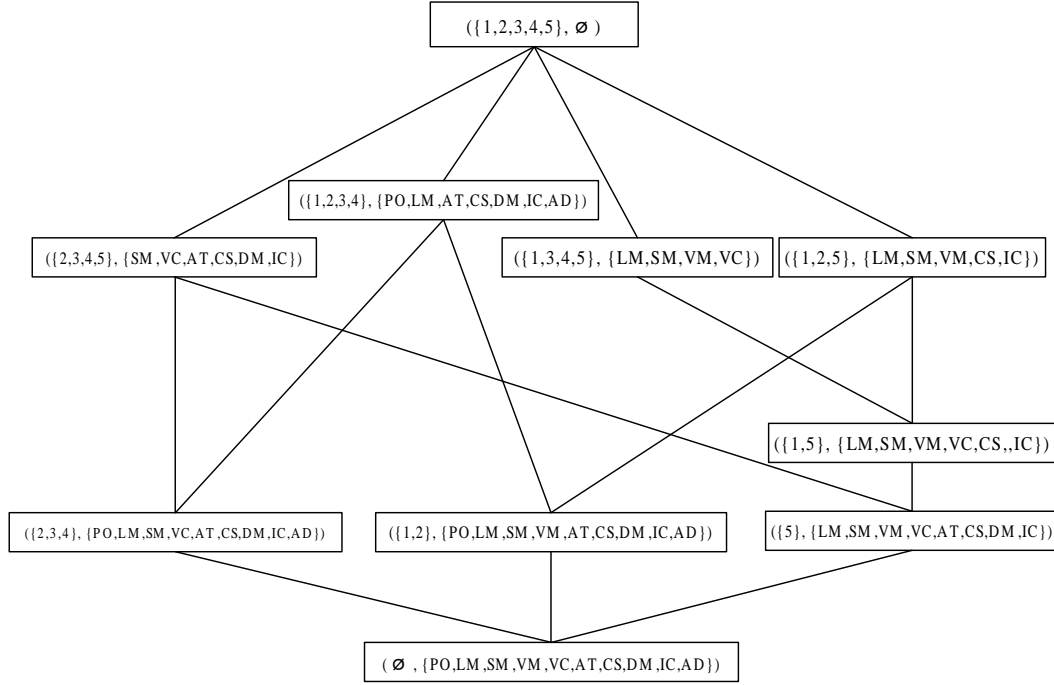


Fig. 3. A Lattice of Dual Formal Concepts

a partial order relation between the concepts. From Figure 3 and Figure 2, one can reveal that a dual formal concept in the lattice can be obtained by taking the set complements of the formal concept in the formal concept lattice, and vice versa.

Consider the operator $\#\# : 2^U \rightarrow 2^U$, which is an interior operator on 2^U . The properties of this operator are: for $A, A_1, A_2 \subseteq U$.

- (NI) $\emptyset^{\#\#} = \emptyset$,
- (NII) $A^{\#\#} \subseteq A$,
- (NIII) $A^{\#\#} = A^{\#\#\#}$,
- (NIV) $A_1 \subseteq A_2 \implies A_1^{\#\#} \subseteq A_2^{\#\#}$.

Property (NI) states that the application of operator $\#\#$ on the empty set is the empty set. The properties (NIII) - (NIV) mean that $\#\#$ is an idempotent and monotonically decreasing operator.

Let $(U, \#\#)$ denote the family of sets $(U, \#\#) = \{A^{\#\#} \mid A \subseteq U\}$. In fact, $(U, \#\#) \subseteq 2^U$ contains the empty set \emptyset and the set of the extensions of all dual formal concepts. The system $(U, \#\#)$ has the following properties:

- (N1) $\emptyset \in (U, \#\#)$,
- (N2) $A_1, A_2 \in (U, \#\#) \implies A_1 \cup A_2 \in (U, \#\#)$.

That is, the system $(U, \#\#)$ is closed under the operator $\#\#$ and set union \cup , but not closed under the set intersection \cap . The system $(U, \#\#)$ consists of all fixed points of operator $\#\#$, that is, $(U, \#\#) = \{A^{\#\#} = A \mid A \subseteq U\}$.

5.4 Property-oriented Concept Lattice

The approximation operators \diamond and \square can be used to form two different types of lattices, property-oriented formal concept lattice and object-oriented formal concept lattice [18,51,54].

A pair (A, B) , $A \subseteq U, B \subseteq V$, is called a property-oriented formal concept if $A = B^\square$ and $B = A^\diamond$. The set of objects A is referred to as the extension of the concept (A, B) , and the set of attributes B is referred to as the intension. If an attribute is possessed by an object in A then the attribute must be in B . Moreover, only attributes B are possessed by objects in A .

For two property-oriented formal concepts (A_1, B_1) and (A_2, B_2) , we say that (A_1, B_1) is a sub-concept of (A_2, B_2) , and (A_2, B_2) is a super-concept of (A_1, B_1) , if and only if $A_1 \subseteq A_2$, or equivalently, if and only if $B_1 \subseteq B_2$.

The family of all property-oriented formal concepts is a complete lattice called property-oriented formal concept lattice [54]. The meet \wedge and the join \vee in property-oriented formal concept lattice are defined by:

$$\begin{aligned} (A_1, B_1) \wedge (A_2, B_2) &= ((A_1 \cap A_2), (B_1 \cap B_2)^{\square\diamond}), \\ (A_1, B_1) \vee (A_2, B_2) &= ((A_1 \cup A_2)^{\diamond\square}, (B_1 \cup B_2)). \end{aligned}$$

Example 7 *The property-oriented concept lattice of Table 1 is illustrated in Figure 4. Each rectangle stands for a property-oriented formal concept in the lattice. The lines between rectangles demonstrate partial order relations between the property-oriented formal concepts.*

Consider the operator $\diamond\square : 2^U \rightarrow 2^U$, which is a closure operator on 2^U [14,18]. The properties of this operator are: for $A, A_1, A_2 \subseteq U$.

$$\begin{aligned} \text{(PI)} \quad & U^{\diamond\square} = U, \\ \text{(PII)} \quad & A \subseteq A^{\diamond\square}, \\ \text{(PIII)} \quad & A^{\diamond\square} = A^{\diamond\square\diamond\square}, \\ \text{(PIV)} \quad & A_1 \subseteq A_2 \implies A_1^{\diamond\square} \subseteq A_2^{\diamond\square}. \end{aligned}$$

The property (PI) and (PII) state that the universe U will not be changed under the operator $\diamond\square$. The properties (PIII) - (PIV) mean that $\diamond\square$ is an idempotent and monotonically increasing operator.

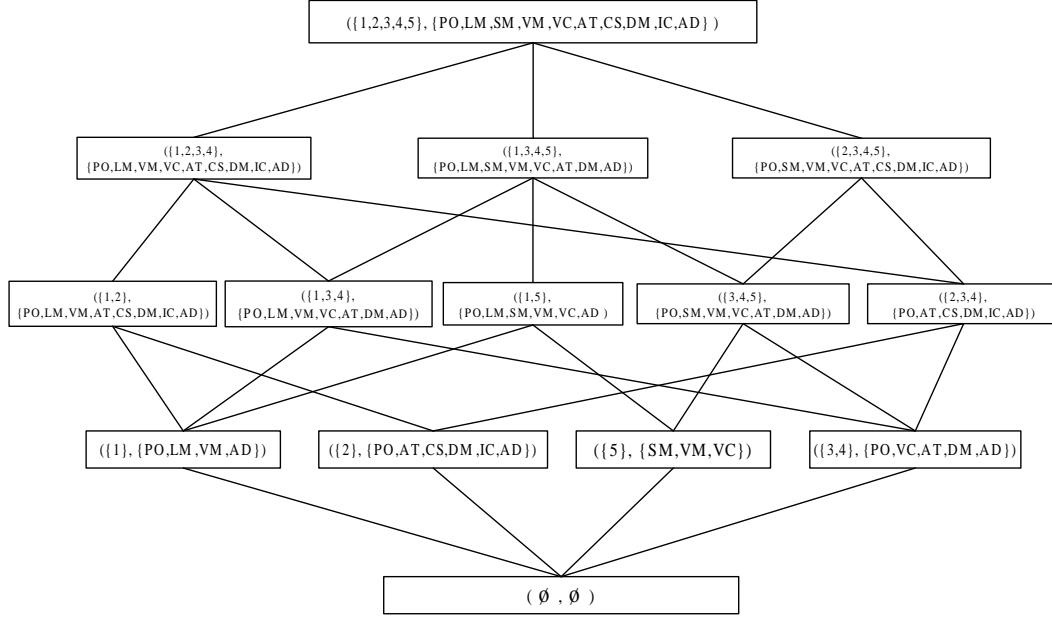


Fig. 4. The Lattice of Property-oriented Concepts

Let (U, \diamond^{\square}) denote the family of sets $(U, \diamond^{\square}) = \{A^{\diamond^{\square}} \mid A \subseteq U\}$. It contains the universe U and the empty set \emptyset and is the set of the extensions of all property-oriented formal concepts in the lattice. The system (U, \diamond^{\square}) is closed under the operator \diamond^{\square} and the set intersection \cap , but not closed under the set union \cup . That is, the system (U, \diamond^{\square}) is a closure system with the following properties:

- (P1) $\emptyset \in (U, \diamond^{\square})$,
- (P2) $A_1, A_2 \in (U, \diamond^{\square}) \implies A_1 \cap A_2 \in (U, \diamond^{\square})$.

Furthermore, for a non-empty set $A \in (U, \diamond^{\square})$, we can have:

$$(A, A^{\diamond}) = \bigvee \{(\{x\}^{\diamond^{\square}}, \{x\}^{\diamond}) \mid x \in A\}.$$

That is, the $\{x\}^{\diamond^{\square}}$ can be used to re-express the sets in the system (U, \diamond^{\square}) . The whole system (U, \diamond^{\square}) can be generated by the sets of $\{x\}^{\diamond^{\square}}$, $x \in U$. The pair $(\{x\}^{\diamond^{\square}}, \{x\}^{\diamond})$ is a property-oriented formal concept and can be used to generate other property-oriented formal concepts in the lattice.

5.5 Object-oriented Concept Lattice

Another type of concept lattice formed by approximation operators \diamond and \square is object-oriented concept lattice [18,51,54]. A pair (A, B) , $A \subseteq U, B \subseteq V$, is called an object-oriented formal concept if $A = B^{\diamond}$ and $B = A^{\square}$. The set of

objects A is referred to as the extension of the concept (A, B) , and the set of attributes Y is referred to as the intension. If an object has an attribute in B then the object belongs to A . Moreover, only objects in A possess attributes in B .

For two object-oriented formal concepts (A_1, B_1) and (A_2, B_2) , we say that (A_1, B_1) is a sub-concept of (A_2, B_2) , and (A_2, B_2) is a super-concept of (A_1, B_1) , if and only if $A_1 \subseteq A_2$, or equivalently, if and only if $B_1 \subseteq B_2$.

The family of all object-oriented formal concepts is a complete lattice called object-oriented formal concept lattice [54]. The meet \wedge and the join \vee in object-oriented formal concept lattice are defined by:

$$\begin{aligned} (A_1, B_1) \wedge (A_2, B_2) &= ((A_1 \cap A_2)^{\square\Diamond}, (B_1 \cap B_2)), \\ (A_1, B_1) \vee (A_2, B_2) &= ((A_1 \cup A_2), (B_1 \cup B_2)^{\Diamond\square}). \end{aligned}$$

Moreover, because of the property of duality of the operators \square and \Diamond , one can re-express one operator with the other operator. That is, $\square\Diamond = c^{\Diamond\square}c$ and $c^{\square\Diamond}c = \Diamond\square$. In other words, the concepts in object-oriented formal concept lattice can be obtained by taking the set complements of the concepts in property-oriented formal concept lattice.

Example 8 *Based on Table 1, an object-oriented formal concept lattice can be derived and illustrated in Figure 5. Rectangles represent the object-oriented formal concepts in the lattice. Lines between rectangles express the partial order relations among the concepts. From Figure 4 and Figure 5, one can reveal that a concept in the object-oriented concept lattice can be obtained by taking the set complements of the concept in the property-oriented concept lattice, and vice versa.*

Consider the operator $\square\Diamond : 2^U \rightarrow 2^U$, which is an interior operator on 2^U [14,18]. The properties of this operator are: for $A, A_1, A_2 \subseteq U$.

- (LI) $\emptyset^{\square\Diamond} = \emptyset$,
- (LII) $A \subseteq A^{\square\Diamond}$,
- (LIII) $A^{\square\Diamond} = A^{\square\Diamond\square\Diamond}$,
- (LIV) $A_1 \subseteq A_2 \implies A_1^{\square\Diamond} \subseteq A_2^{\square\Diamond}$.

The properties (LI) and (LII) state that the empty set \emptyset will not be changed under the operator $\square\Diamond$. The properties (LIII) - (LIV) mean that $\square\Diamond$ is an idempotent and monotonically decreasing operator.

Let $(U, \square\Diamond)$ denote the family of sets $(U, \square\Diamond) = \{A^{\square\Diamond} \mid A \subseteq U\}$. It contains the universe U and the empty set \emptyset and is the family of the extension of all

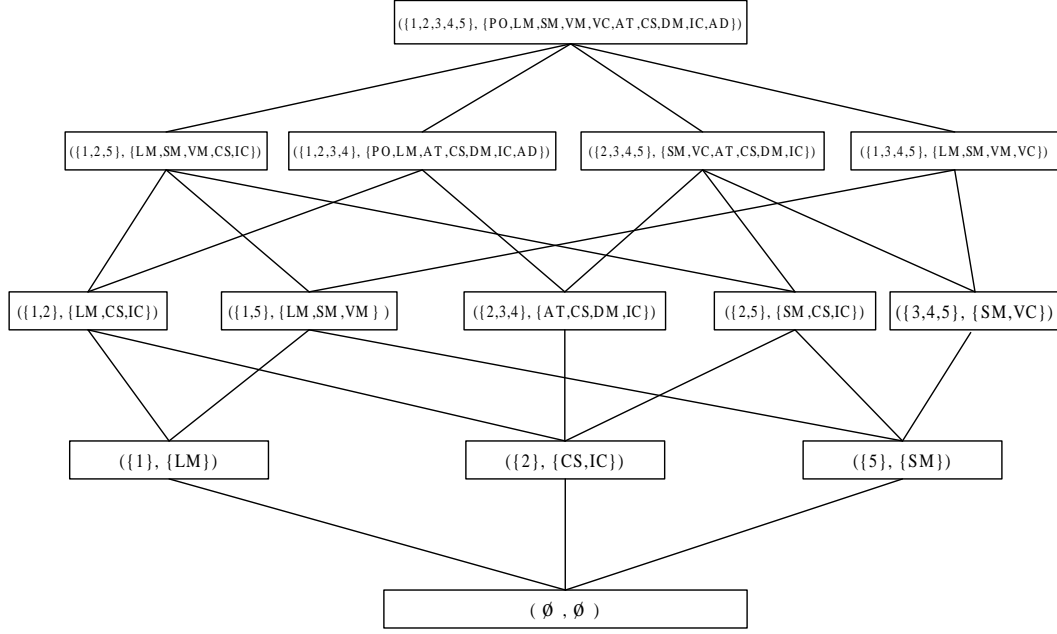


Fig. 5. The Lattice of Object-oriented Concepts

object-oriented formal concepts. The properties of the system (U, \square^\diamond) are:

- (L1) $U \in (U, \square^\diamond)$,
- (L2) $A_1, A_2 \in (U, \square^\diamond) \implies A_1 \cup A_2 \in (U, \square^\diamond)$.

These properties state that the system (U, \square^\diamond) is closed under the operator \square^\diamond and the set union \cup , but not closed under set intersection \cap .

5.6 Connections between Different Lattices

Based on the connections of modal-style data operators, the concepts in various lattices can be expressed by equivalence classes as follows. For $A \subseteq U, B \subseteq V$,

$$\begin{aligned}
 A^{**} &= \bigcup \{[x] \mid x \in U, A^* \subseteq xR\}, \\
 A^{\#\#} &= \bigcup \{[x] \mid x \in U, A^\# \cup xR \neq V\}, \\
 A^{\square^\diamond} &= \bigcup \{[x] \mid x \in U, A^\square \cap xR \neq \emptyset\}, \\
 A^{\diamond^\square} &= \bigcup \{[x] \mid x \in U, xR \subseteq A^\diamond\},
 \end{aligned}$$

and

$$B^{**} = \bigcup \{[y] \mid y \in V, B^* \subseteq Ry\},$$

$$\begin{aligned} B^{\#\#} &= \bigcup \{[y] \mid y \in V, B^{\#} \cup Ry \neq V\}, \\ B^{\square\lozenge} &= \bigcup \{[y] \mid y \in V, B^{\square} \cap Ry \neq \emptyset\}, \\ B^{\lozenge\square} &= \bigcup \{[y] \mid y \in V, Ry \subseteq B^{\lozenge}\}. \end{aligned}$$

It follows that different types of lattices can be connected and transformed from each other by using equivalence classes. The equivalence classes can be considered as a base to construct different types of lattices.

6 Conclusion

In this paper, we propose an approach to investigate multiple views of intelligent data analysis based on modal-style data operators. We provides a basic framework to explore the significant importance of multiview for the research on intelligent data analysis. The integration of these multiple views may bring more powerful and useful data analysis tools.

By defining various types of data operators, we present various data relationships and construct different types of hierarchical structures, which provides different granulated views of the data. A type of data relationships has a specific semantic interpretation and may be considered as a type of knowledge discovered from a data set. Different types of data relationships present different types of knowledge. Different hierarchical structures defined based on modal-style data operators represent various aspects or features of data and can be viewed as multiple types of knowledge embedded in the data.

For future work, we will study the connections and transformations between different data relationships or data structures. A general framework to study different views and their connections and transformations may bring more insights into intelligent data analysis.

References

- [1] Agrawal, R., Imielinski, T., and Swami, A. Mining association rules between sets of items in large databases, *Proceedings of International Conference on Mangement of Data (ACM SIGMOD)*, 207-216, 1993.
- [2] Aleksander, I. and Morton, H. *An Introduction to Neural Computing*, Chapman & Hall, London, 1990.
- [3] Amari, S. Mathematical foundations of neurocomputing, *Proceedings of the IEEE*, **78**, 1443-1463, 1990.

- [4] Anderberg, M.R. *Cluster Analysis for Application*, Academic Press, New York, 1973.
- [5] Bajcsy, R. and Kovacic, S. Multiresolution elastic matching, *Computer Vision Graphic Image Processing*, **46**, 1-21, 1989.
- [6] Bargiela, A. and Pedrycz, W. *Granular Computing: an Introduction*, Kluwer Academic Publishers, Boston, 2002.
- [7] Belkhouche, B. and Lemus-Olalde, C. Multiple view analysis of designs, *Foundations of Software Engineering, Joint proceedings of the 2nd International Software Architecture Workshop (ISAW-2) and International Workshop on Multiple Perspectives in Software Development, (SIGSOFT'96)*, 159-161, 1996.
- [8] Belkhouche, B. and Lemus-Olalde, C. Multiple views analysis of software designs, *International Journal of Software Engineering and Knowledge Engineering*, **10**, 557-579, 2000.
- [9] Breiman, L., Friedman, J.H., Olshen, R.A. and Stone, C.J. *Classification and Regression Trees*, Wadsworth, California, 1984.
- [10] Buszkowski, W. Approximation spaces and definability for incomplete information systems, *Rough Sets and Current Trends in Computing, Proceedings of the 1st International Conference (RSCTC'98)*, 115-122, 1998.
- [11] Chen, Y.H. and Yao, Y.Y. Multiview Intelligent Data Analysis based on Granular Computing, *Proceedings of 2006 IEEE International Conference on Granular Computing (Grc'06)*, 281-286, 2006.
- [12] Cohn, P.M. *Universal Algebra*, Harper and Row Publishers, New York, 1965.
- [13] Cover, T.M. and Hart, P.E. Nearest neighbor pattern classification, *IEEE Transactions on Information Theory*, **IT-13**, 21-27, 1967.
- [14] Düntsch, I. and Gediga, G. Approximation operators in qualitative data analysis, in: *Theory and Application of Relational Structures as Knowledge Instruments*, de Swart, H., Orłowska, E., Schmidt, G. and Roubens, M. (Eds.), Springer, Heidelberg, 216-233, 2003.
- [15] Ester, M., Kriegel, H.P., Sander, J., and Xu, X. A density-based algorithm for discovering clusters in large spatial data sets with noise. *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*. 226-231, 1996.
- [16] Freitas, A.A. Understanding the crucial differences between classification and discovery of association rules - A position paper, *SIGKDD Explorations*, **2**, 65-69, 2000.
- [17] Ganter, B. and Wille, R. *Formal Concept Analysis: Mathematical Foundations*, Springer-Verlag, New York, 1999.
- [18] Gediga, G. and Düntsch, I. Modal-style operators in qualitative data analysis, *Proceedings of the 2002 IEEE International Conference on Data Mining*, 155-162, 2002.

- [19] Grenander, U. *General Pattern Theory*, Oxford University Press, New York, 1993.
- [20] Han, H., Manavoglu, E., Lee, G.C. and Zha, H.Y. Rule-based word clustering for text classification. *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'03)*, 445-446, 2003.
- [21] Hand, D.J. Intelligent data analysis: issues and opportunities, *Intelligent Data Analysis*, **2**, 67-79, 1998.
- [22] Jain, A.K., Murty M.N., and Flynn P.J. Data clustering: a review, *ACM Computing Surveys*, **31**, 264-323, 1999.
- [23] Jain, A.K., Duin, R.P., and Mao, J. Statistical pattern recognition: a review, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**, 4-37, 2000.
- [24] Jeffries, V. and Ransford, H.E. *Social Stratification: A Multiple Hierarchy Approach*, Allyn and Bacon, Inc., Boston, 1980.
- [25] Jeon, B. and Landgrebe, D. Partially supervised classification using weighted unsupervised clustering, *IEEE Transactions on Geoscience and Remote Sensing*, **37**, 1073-1079, 1999.
- [26] King, B. Step-wise clustering procedures, *Journal of the American Statistical Association*, **69**, 86-101, 1967.
- [27] Kotsiantis, S. and Pintelas, P.E. Recent advances in clustering: a brief survey, *WSEAS Transactions on Information Science and Applications*, **1**, 73-81, 2004.
- [28] Kulkarni, S.R. Lugosi, G. and Venkatesh, S.S., Learning pattern classification - a survey, *IEEE Transactions on Information Theory*, **44**, 2178-2206, 1998.
- [29] Lin, C.R., Liu, K.H. and Chen, M.S. Dual clustering: integrating data clustering over optimization and constraint domains, *IEEE Transactions on Knowledge and Data Engineering*, **17**, 628-637, 2005.
- [30] Lin, T.Y. Granular computing, *announcement of the BISC Special Interest Group on Granular Computing*, 1997.
- [31] Liu, B., Hsu, W. and Ma, Y. Integrating classification and association rules mining. *Proceedings of 4th International Conference on Knowledge Discovery and Data Mining (KDD'98)*, 80-86, 1998.
- [32] Lu, S.Y. and Fu, K.S. A sentence-to-sentence clustering procedure for pattern analysis, *IEEE Transactions on System, Man and Cybernetics*, **8**, 381-389, 1978.
- [33] Mcqueen, J. Some methods for classification and analysis of multivariate observations. *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, 281-297, 1967.
- [34] Michalski, R. (Ed.) *Multistrategy Learning*, Kluwer Academic Publishers, Boston, 1993.

- [35] Minsky, M.L. and Papert, S.A. *Perceptrons*, MIT Press, Cambridge, 1988.
- [36] Nguyen, S.H., Skowron, A. and Stepaniuk, J. Granular computing: a rough set approach, *Computational Intelligence*, **17**, 514-544, 2001.
- [37] Orlowska, E. Introduction: what you always wanted to know about rough sets, in: *Incomplete Information: Rough Set Analysis*, Ewa Orlowska (Ed.), Physica-Verlag Heidelberg, New York, 1998.
- [38] Parasuraman, R. Humans and automation: use, misuse, disuse, abuse, *Human Factors*, **39**, 230-253, 1997.
- [39] Pawlak, Z. Rough sets, *International Journal of Computer Information Science*, **11**, 341-356, 1982.
- [40] Parzen, E. On the estimation of a probability density function and the mode, *Annals of Mathematical Statistics*, **33**, 1065-1076, 1962.
- [41] Pearl, J. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, San Mateo, California, 1988.
- [42] Pedrycz, W. Granular computing: an introduction, *Proceedings of Joint the 9th IFSA World Congress and the 20th NAFIPS International Conference, IEEE Press*, 1349-1354, 2001.
- [43] Pedrycz, W. and Vukovich, G. Granular computing with shadowed sets, *International Journal of Intelligent Systems*, **17**, 173-197, 2002.
- [44] Quinlan, J.R. Simplifying decision trees, *International Journal on Man-Machine Studies*, **27**, 221-234, 1987.
- [45] Quinlan, J.R. *C4.5: Programs for Machine Learning*, Morgan Kaufmann, San Mateo, California, 1993.
- [46] Sneath, P.H.A. and Sokal, R.R. *Numerical Taxonomy*, Freeman, London, 1973.
- [47] Sprandel, H.Z. *The Psychoeducational Use and Interpretation of The Wechsler Adult Intelligence Scale-Revised*, Charles C Thomas, U.S.A. 1985.
- [48] Vapnik, V.N. *Statistical Learning Theory*, John Wiley & Sons, New York, 1998.
- [49] Ward, J.H. Hierarchical grouping to optimize an objective function, *Journal of American Statistical Association*, **58**, 236-244, 1963.
- [50] Wille, R. Restructuring lattice theory: an approach based on hierarchies of concepts, in: *Ordered sets*, Ivan Rival (Ed.), Reidel, Dordrecht-Boston, 445-470, 1982.
- [51] Wolski, M. Galois connections and data analysis, *Fundamenta Informaticae CSP*, 1-15, 2003.
- [52] Wong, S.K.M, Wang, L.S. and Yao, Y.Y. On modeling uncertainty with interval structures, *Computational Intelligence*, **11**, 406-426, 1995.

- [53] Yao, Y.Y. Granular computing, *Computer Science (Ji Suan Ji Ke Xue), Proceedings of the 4th Chinese National Conference on Rough Sets and Soft Computing*, **31**, 1-5, 2004.
- [54] Yao, Y.Y. Concept lattices in rough set theory, *Proceedings of Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS'04)*, 796-801, 2004.
- [55] Yao, Y.Y. A comparative study of formal concept analysis and rough set theory in data analysis, *Proceedings of the International Conference on Rough Sets and Current Trends in Computing (RSCTC'04)*, 59-68, 2004.
- [56] Yao, Y.Y. Relational interpretations of neighborhood operators and rough set approximation operators, *Information Sciences*, **111**, 239-259, 1998.
- [57] Yao, Y.Y. On generalizing rough set theory, *Proceedings of the 9th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing (RSFDGrC'03)*, 44-51, 2003.
- [58] Yin, X. and Han, J. CPAR: Classification based on predictive association rules. *Proceedings of SIAM International Conference on Data Mining (SDM'03)*, 331-335, 2003.
- [59] Zadeh, L.A. Towards a theory of fuzzy information categorization and its centrality in human reasoning and fuzzy logic, *Fuzzy Sets and Systems*, **19**, 111-127, 1997.
- [60] Zeng, H.J., Wang, X.H., Chen, Z., Lu, H.J. and Ma, W.Y. CBC: clustering based text classification requiring minimal labeled data, *Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM'03)*, 443-450, 2003.
- [61] Zhong, N. Multi-aspect data analysis in brain informatics, *Proceedings of the 1st International Conference on Pattern Recognition and Machine Intelligence, (PReMI'05)*, LNCS 3776, 98-107, 2005.
- [62] Zhong, N., Yao, Y.Y., Wu, J.L. and Liu J. Grid based multi-aspect data analysis in brain informatics protal, *Proceedings of ICDM Workshop on Foundation of Semantic Oriented Data and Web Mining*, 130-137, 2005.
- [63] Zimmermann, A. and Raedt, L.D. CorClass: correlated association rule mining for classification. *Proceedings of the 7th International Conference on Discovery Science (DS'04)*, 60-72, 2004.

Appendix

The four binary relations on attributes are defined as follows.

Equivalence Relation. For $y, y' \in V$, $y \equiv_V y' \iff Ry = Ry'$, which is reflective, symmetric and transitive.

Partial Order Relation. For $y, y' \in V$, $y \preceq_V y' \iff Ry \subseteq Ry'$, which is reflective and transitive.

Similarity Relation: For $y, y' \in V$, $y \approx_V y' \iff Ry' \cap Ry \neq \emptyset$, which is reflective and symmetric.

Negative Similarity Relation: For $y, y' \in V$, $y \succsim_V y' \iff Ry^c \cap Ry'^c \neq \emptyset \iff Ry \cup Ry' \neq U$, which is symmetric.

Modal-style data operators defined for relations between attributes are listed as follows.

In a formal context (U, V, R) , for an attribute $y \in V$, its equivalence class can be defined by: $\equiv_V y = y \equiv_V = [y] = \{y\}^{*b}$.

In a formal context (U, V, R) , for an attribute $y \in V$, the \preceq_V successor neighborhood is given by: $y \preceq_V = \{y\}^{**}$.

In a formal context (U, V, R) , for an attribute $y \in V$, the \preceq_V predecessor neighborhood is given by: $\preceq_V y = \{y\}^{\diamond\Box}$.

In a formal context (U, V, R) , for an attribute $y \in V$, the \approx_V predecessor and successor neighborhood are given by: $\approx_V y = y \approx_V = \{y\}^{\diamond\diamond}$.

In a formal context (U, V, R) , for an attribute $y \in V$, the \succsim_V predecessor and successor is: $\succsim_V y = y \succsim_V = \{y\}^{*\#}$.