Hindawi
Computational Intelligence and Neuroscience
Volume 2020, Article ID 5784209, 11 pages
https://doi.org/10.1155/2020/5784209

Research Article

Layered Concept Lattice Model and Its Application to Build Rapidly Concept Lattice

Xia Wu,[1] Jialu Zhang ,[1] and Jiaming Zhong[2]

[1]*College of Mathematics and Finance, Xiangnan University, Chenzhou 423000, China*
[2]*College of Economic and Management, Xiangnan University, Chenzhou 423000, China*

Correspondence should be addressed to Jialu Zhang; zjl0735@163.com

Received 15 February 2020; Revised 25 April 2020; Accepted 1 May 2020; Published 11 June 2020

Academic Editor: Mario Versaci

When some attributes of a formal context can be decomposed into some subattributes a model of layered concept lattice to improve the efficiency of building concept lattice with complex structure attribute data is studied, the relationship between concept lattice and layered concept is discussed. Two algorithms are proposed: one is the roll-up building algorithm in which the upper concepts are built by the lower concept and the other is the drill-down algorithm in which the lower concepts are built by the upper concept. The examples and experiments show that the layered concept lattice model can be used to model complex structure attribute data, and the roll-up building algorithm and the drill-down algorithm are effective. The layered concept lattice model expands the scope of the research and application of concept lattice, the roll-up building algorithm, and drill-down algorithm of layered concept lattice to improve the efficiency for building concept lattice.

1. Introduction

Humans usually describe and recognize objective things from different levels and different granularity. There is a process of deepening the attribute characteristics of objective things, and hence the knowledge concepts at different levels or at different granularity are obtained [1, 2]. Granular computing is a kind of useful mathematical method for processing complex structure data, and the idea of granular computing fits perfectly with the hierarchical and granular thinking mode of "from coarse to fine, from whole to part" in the process of human cognition [3–5]. The idea of granular computing originated from professor Zadeh [6]. Since Lin summarized relevant studies and introduced the term granular computing in 1998 [7], the thinking and methods of granular computing have appeared in many fields, such as rough set, fuzzy set, evidence theory, cluster analysis, machine learning, data mining, and knowledge discovery. In recent years, research studies on granular computing have been extensive and many meaningful results have been obtained. For example, Yao studied the basic problems and methods of granular computing [3], Lin studied the granular structure and representation [7], Pedrycz studied the granular computing methodology, mathematical framework, and information granulation algorithm [8], and Zhang and Zhang studied the quotient space [9].

Concept lattice is the key data structure of formal context analysis, and it is also a kind of basic data structure suitable for knowledge representation and knowledge discovery [10]. There have been many studies on basic theory of concept lattice [11] and algorithms to accelerate the construction of concept lattice [12–14], and concept lattice has been widely applied in many fields including data mining [15], knowledge discovery [16], information retrieval, and information extraction [17, 18]. Theory and application research of concept lattice are inseparable from the structure analysis of concept lattice and the model extension of concept lattice. The existing concept lattice models and their extension models are established on the relation between objects and attributes, they handle only the simple type of data, and there is no model that has been able to handle data with complex structures. For example, by using a fuzzy set on the basic language set [19], Wolff represented attribute values with fuzzy language variable values in a formal context, classified objects according to the scale, and then built concept lattice based on the scale [20]. Burusco and Fuentes discussed the concept lattice structure based on L-fuzzy concept set and proposed a method to build concept lattice in this case [21]. Considering that Burusco and Fuentes' models cannot handle continuous membership values and have high computational complexity, Qiang et al. proposed a fuzzy conceptual lattice model that can handle continuous membership values, and based on this model, they discussed the extraction of fuzzy rules and the clustering of fuzzy concepts [22]. Ali and Samir introduced the real interval formal context, established the real interval concept lattice, and built the classifier on the real interval concept lattice for data classification [23]. Qu et al. aimed to establish the relationship between formal concept analysis and rough set theory, obtained derivative formal context that can be induced by the notion of nominal scale and plain scaling and the technique of plain scaling in an information system, and proved that some core notions in rough theory such as partition, upper and lower approximation, independence, dependence, and reduction can be reinterpreted in derivative formal context. They pointed out that the rough set model can be extended by using plain scaling, but they have not further studied the concept lattice structure in the derivative formal context [24].

Due to the complexity of research objects, complex structural data are widely used in practical applications. For example, in the field of network intelligent information processing, it is necessary to analyze the correlation among characters, time, action, environment, and other elements through text, image, audio, video, etc. These elements are usually expressed in different types of data [25]. In particular, under big data environment, the remarkable feature of data is the diversity of types and the complexity of structures. An application often deals with both structured data and semistructured, unstructured data such as text, images, audio, video, video, and Web. In order to apply formal concept analysis to study knowledge representation and knowledge discovery in big data environment, the first thing to do is to extend the existing concept lattice model. In the process of extending the concept lattice model in big data environment, the attribute values used to describe the characteristics of unstructured data, such as text, images, audio, video, and Web, include word values, text values, vector values, and their composite values, in addition to the usual number value and character value [26, 27]. On formal context analysis on complex structural data, Zhi proposed a generalized concept lattice model for heterogeneous data analysis by discussing the partial order composition on heterogeneous datasets [28, 29]. However, Zhi's concept lattice model for heterogeneous data is only considered at the same level and at the same granularity, which does not reflect the idea of hierarchy and granularity of human thinking. Therefore, to describe human thinking process from a mathematical perspective or to apply the concept lattice theory to describe the human thinking process, we need to build the concept lattice model at different levels and different granularity.

In order to satisfy the need for describing in detail the knowledge concept on unstructured dataset, the thinking of granular computing is applied to the formal concept analysis of data with complex structure. The layered structure of attribute data is fully considered, and a layered concept lattice model with layered and three-dimensional structure is built in the complex structure data environment. The relation between concept lattice of original formal context and layered concept lattice of layered formal context is discussed. The roll-up building algorithm in which the upper concept is built by the lower concept and the drill-down algorithm in which the lower concept is built by the upper concept are proposed. The examples and experiments show that the roll-up and the drill-down algorithms are effective.

2. Preliminary

In this section, we summarize some basic notions and conclusions on concept lattices. For more details of these notions and conclusions, we refer the reader to Ganter and Wille's works "Formal Concept Analysis" [10].

Definition 1. A formal context is a triplet $K = (G, M, I)$, where $I \subseteq G \times M$ is a binary relation between G and M. The elements in G and M are called objects and attributes, respectively. $(g, m) \in I$ or gIm indicates the object g has the attribute m.

Definition 2. Let $K = (G, M, I)$ be a formal context. For a set $A \subseteq G$ of objects, we define a set of attributes common to all objects in A as

$$\alpha(A) = \{m \in M \mid gIm, \forall g \in A\}. \tag{1}$$

Correspondingly, for a set $B \subseteq M$ of attributes, we define the set of objects that have all attributes in B as

$$\beta(B) = \{g \in G \mid gIm, \forall m \in B\}. \tag{2}$$

In brief, we also denote $\alpha(\{a\}), \beta(\{m\})$ as $\alpha(a), \beta(m)$.

Proposition 1 (see [10]). *For $A, A_1, A_2 \subseteq G$ and $B, B_1, B_2 \subseteq M$,*

(1) $A_1 \subseteq A_2 \Longrightarrow \alpha(A_2) \subseteq \alpha(A_1), B_1 \subseteq B_2 \Longrightarrow \beta(B_2) \subseteq \beta(B_1)$

(2) $A \subseteq \beta(\alpha(A)), B \subseteq \alpha(\beta(B))$

(3) $A \subseteq \beta(B) \Longleftrightarrow B \subseteq \alpha(A)$

(4) $\alpha(A_1 \cup A_2) = \alpha(A_1) \cap \alpha(A_2)$, $\beta(B_1 \cup B_2) = \beta(B_1) \cap \beta(B_2)$

(5) $\alpha(A_1 \cap A_2) \supseteq \alpha(A_1) \cup \alpha(A_2)$, $\beta(B_1 \cap B_2) \supseteq \beta(B_1) \cup \beta(B_2)$

(6) $(\beta(\alpha(A)), \alpha(A))$ *and* $(\beta(B), \alpha(\beta(B)))$ *are concepts*

Definition 3. Let $K = (G, M, I)$ be a formal context. For $A \subseteq G$ and $B \subseteq M$, if $\alpha(A) = B$ and $\beta(B) = A$, then a pair (A, B) is called a formal concept of formal context K, and A and B are called the extent and the intent of (A, B), respectively. The set of all concepts of formal context K is denoted as $C(K)$.

Definition 4. Let (A_1, B_1) and (A_2, B_2) be two formal concepts of a given formal context K. (A_1, B_1) is called a subconcept of (A_2, B_2) if $A_1 \subseteq A_2$(or equivalently $B_1 \supseteq B_2$), which can be denoted by $(A_1, B_1) \leq (A_2, B_2)$.

Obviously, the subconcept relation $\leq$ is a partial order on $C(K)$. Since $\leq$ is a partial order, we can adopt the definition of neighboring nodes of order theory here. Let C_1 and C_2 be two concepts of a given formal context K. We say C_1 is a lower neighbor (or a child) of C_2 and C_2 is an upper neighbor (or a parent) of C_1, if $C_1 \leq C_2$ and there is no other concept C_3 with $C_3 \neq C_1$, $C_3 \neq C_2$, and $C_1 \leq C_3 \leq C_2$.

Theorem 1 (see [10]). *The set $C(K)$ of all formal concepts of context K together with partial order $\leq$ makes a complete lattice, which is called a concept lattice of K and is denoted as $(C(K), \leq)$; the upper and lower bound operations are*

$$\begin{aligned} &\wedge_{t\in T}(A_t, B_t) = (\cap_{t\in T} A_t, \alpha(\beta(\cup_{t\in T} B_t))), \\ &\vee_{t\in T}(A_t, B_t) = (\beta(\alpha(\cup_{t\in T} A_t)), \cap_{t\in T} B_t). \end{aligned} \tag{3}$$

3. Layered Concept Lattice Model and Its Roll-Up Algorithm to Build the Upper Concept Lattice

According to the analysis in the introduction, when we discuss the concept lattice theory and application of complex data structure, the problem of data attribute stratification is often encountered. For example, on the basis of the discussing new energy vehicles at coarse-grained, we often need to discuss the problem of new energy vehicles at a finer granularity to meet the needs of practical application: hybrid vehicles and pure electric vehicles. In terms of the concept lattice, when the new energy attribute is decomposed into lower hybrid power and pure electric power attributes, we need to analyze the formal concept in the layered formal context.

In this section, we give the definition of layered formal context and layered concept lattice, discuss their properties, and propose a roll-up building algorithm of layered concept lattice in a normal layered formal context.

3.1. Layered Concept Lattice Model and Roll-Up Algorithm Theory

Definition 5. Let $K = (G, M, I)$ be a formal context. If every attribute in M can be represented as a subset, which means that

$$\begin{aligned} &M = \{m_1, m_2, \cdots, m_l\}, \\ &m_k = \{m_{k1}, m_{k2}, \cdots, m_{ki_k}\}, \quad k = 1, 2, \cdots, l, \end{aligned} \tag{4}$$

then $K = (G, M, I)$ is called a layered formal context and m_k is called a layered attribute. In a layered formal context $K = (G, M, I)$, the lower attributes corresponding to the layered attribute m_k are denoted as

$$\overline{m_k} = \{(m_k, m_{k1}), (m_k, m_{k1}), \cdots, (m_k, m_{ki_k})\}. \tag{5}$$

If $i_k = 1$, then $\overline{m_k} = m_k$, which is actually not a layered attribute.

By using the formal context $K = (G, M, I)$, we can introduce a new layered formal context $\overline{K} = (G, N, J)$, where $N = \cup_{k=1}^{l} \overline{m_k}$, and the relation J between objects and attributes is defined by $gJ(m_k, m_{ki}) \Longleftrightarrow gIm_k$. For the convenience of expression, we also call K the upper formal context, $\overline{K}$ the lower formal context, $C(K)$ the upper concept lattice, and $C(\overline{K})$ the lower concept lattice accordingly.

A layered formal context $K = (G, M, I)$ is normal if for every layered attribute gIm_k iff there is a unique iff there is a unique lower attribute m_{ki} such that $gJ(m_k, m_{ki})$.

In this paper, we only discuss the normal layered formal context. The following conclusion is obvious.

Proposition 2. *Suppose that $K = (G, M, I)$ is a layered formal context. Then,*

$$\{\beta(m_k, m_{k1}), \beta(m_k, m_{k2}), \cdots, \beta(m_k, m_{ki_k})\}, \tag{6}$$

is a partition of $\beta(m_k)$.

Example 1. Table 1 shows a simplified layered formal context $K = (G, M, I)$, where a ="off-road vehicle," b = "midrange vehicle," c = "new energy power vehicle" which is a layer attribute, c_1 = "hybrid power," and c_2 = "pure electric power."

The formal context $\overline{K}$ induced by K is shown in Table 2.

When a formal concept K contains layered attributes, the upper concept lattice $C(K)$ is closely related to the lower concept lattice $C(\overline{K})$. Hence, we can start with the lower concept lattice $C(\overline{K})$ by rolling up the lower attributes $\overline{m_k}$ of some lower concepts to an upper attribute m_k and then building a upper concept; this is easier than building the concept lattice directly from the formal context K.

Theorem 2. *Suppose that $K = (G, M, I)$ is a layered formal context, $\overline{K} = (G, N, J)$ is the corresponding lower formal context of K, and $C(K)$ and $C(\overline{K})$ are upper concept lattice and lower concept lattice, respectively. If $\{(A_1, \overline{B_1}), \cdots, (A_\alpha, \overline{B_\alpha})\} \subseteq C(\overline{K})$ contain the same layered attribute $\overline{m_k}$ and are the nearest subconcepts of a node concept, then*

(1) $(A_1, \overline{B_1}), \cdots, (A_\alpha, \overline{B_\alpha})$ can be rolled up as a new concept $(A^{(k)}, B^{(k)})$, where $A^{(k)} = A_1 \cup \cdots \cup A_\alpha$ and $B^{(k)} = ((\overline{B_1} - \overline{m_k}) \cap \cdots \cap (\overline{B_\alpha} - \overline{m_k})) \cup \{m_k\}$.

(2) The lower concepts $(A_i, \overline{B_i})(i = 1, \cdots, \alpha)$ are all updated into $(A_i, (\overline{B_i} - \overline{m_k}) \cup \{m_k\})$, respectively.

Proof. It is worth pointing out before proving that a concept $(A_i, \overline{B_i})$ in $C(\overline{K})$ may contain both nonlayered attributes and lower attributes in the form of (m_k, m_{kl}). In the following, we prove the theorem by four steps.

(1) Prove equation $\alpha(A^{(k)}) = \{m \in M \mid gIm, \forall g \in A^{(k)}\} = B^{(k)}$.

Let $m_l \in \alpha(A^{(k)})$. Then, for each $g \in A^{(k)} = A_1 \cup \cdots \cup A_\alpha$, we have gIm_l. If $m_l = m_k$, then

Table 1: Layered formal context K.

G	a	b	c	
			c_1	c_2
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1
5	1	1	1	0
6	1	1	0	1

Table 2: Formal context $\overline{K}$ induced by K.

G	a	b	(c, c_1)	(c, c_2)
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1
5	1	1	1	0
6	1	1	0	1

obviously $m_l \in B^{(k)}$. If $m_l \neq m_k$, then because $(A_i, \overline{B_i})(i = 1, \cdots, \alpha)$ are concepts and gIm_l for each $g \in A_i$, we have that $m_l \in \overline{B_i}(i = 1, \cdots, \alpha)$. So, $m_l \in (\overline{B_1} - \overline{m_k}) \cap \cdots \cap (\overline{B_{i_k}} - \overline{m_k})$. This means that $m_l \in B^{(k)}$. Hence,

$$\alpha\left(A^{(k)}\right) = \left\{m \in M \mid gIm, \forall g \in A^{(k)}\right\} \subseteq B^{(k)}. \tag{7}$$

Suppose that $m_l \in B^{(k)}$. If $m_l \in \{m_k\}$, then $m_l = m_k$. Hence, m_k are the layered attributes of concepts $(A_1, \overline{B_1}), \cdots, (A_{i_k}, \overline{B_{i_k}})$, so $\alpha(A_i) = \{m \in N \mid \ gJm, \forall g \in A_i\} = \ \overline{B_i}, (i = 1, \cdots, \alpha)$. Thus, for each $g \in A^{(k)} = A_1 \cup \cdots \cup A_\alpha$, there exists i_0 such that $g \in A_{i_0}$, so $gJ(m_k, m_{ki_0})$, and gIm_k. This shows that gIm_k for each $g \in A^{(k)} = A_1 \cup \cdots \cup A_\alpha$. This means that $m_k \in \alpha(A^{(k)})$. If $m_l \in (\overline{B_1} - \overline{m_k}) \cap \cdots \cap (\overline{B_\alpha} - \overline{m_k})$, then $m_l \neq m_k$ is not a layered attribute and $m_l \in \overline{B_i}, (i = 1, \cdots, \alpha)$. Since $(A_i, \overline{B_i}), (i = 1, \cdots, \alpha)$ are concepts, we have gIm_l for each $g \in A^{(k)} = A_1 \cup \cdots \cup A_\alpha$. Hence, $m_l \in \alpha(A^{(k)})$. Thus, $B^{(k)} \subseteq \alpha(A^{(k)}) = \{m \in M \mid gIm, \forall g \in A^{(k)}\}$.

(2) Prove equation $\beta(B^{(k)}) = \{g \in G \mid gI\,m, \forall m \in B^{(k)}\} = A^{(k)}$.

At first, we prove $\beta(B^{(k)}) \subseteq A^{(k)}$. Let us assume $g \in \beta(B^{(k)})$; then, for every $m \in B^{(k)} = ((\overline{B_1} - \overline{m_k}) \cap \cdots \cap (\overline{B_\alpha} - \overline{m_k})) \cup \{m_k\}$, we have gIm. Hence by gIm_k, we know that there are some i such that $gJ(m_k, m_{ki})$, so $gJ\overline{m_k}$, and let i_0 be the subscript of biggest concept $(A_{i_0}, \overline{B_{i_0}})$ in these concepts, where the biggest concept means that the concept contains as many object elements as possible and as fewer attribute elements as possible. Thus, from $(A_{i_0}, \overline{B_{i_0}})$ being the biggest concept, we obtain that

$$g \in \beta\left(\overline{B_{i_0}}\right) = \left\{g \in G \mid gJ(m_k, m_{ki}), \forall (m_k, m_{ki}) \in \overline{B_{i_0}}\right\} = A_{i_0}. \tag{8}$$

This means that $g \in A^{(k)}$. Therefore,

$$\beta\left(B^{(k)}\right) = \left\{g \in G \mid gIm, \forall m \in B^{(k)}\right\} \subseteq A^{(k)}. \tag{9}$$

Secondly, we prove $A^{(k)} \subseteq \beta(B^{(k)})$. Let $g \in A^{(k)}$; then, there are some i such that $g \in A_i$, and let i_0 be the subscript of biggest concept $(A_{i_0}, \overline{B_{i_0}})$ in these concepts. Hence, $g \in A_{i_0} = \beta(\overline{B_{i_0}})$, and for each $m \in \overline{B_{i_0}}$, gJm. This shows that $g \in \{g \in G \mid gIm, \forall m \in B^{(k)}\}$, and then $g \in \beta(B^{(k)})$. Therefore,

$$A^{(k)} \subseteq \beta\left(B^{(k)}\right) = \left\{g \in G \mid gIm, \forall m \in B^{(k)}\right\}. \tag{10}$$

From (1) and (2) we know that $(A^{(k)}, B^{(k)})$ is a concept.

(3) Prove that the concepts $(A_i, \overline{B_i}) \in C(\overline{K})(i = 1, \cdots, \alpha)$, which are not rolled up, are updated new concepts $(A_i, (\overline{B_i} - \overline{m_k}) \cup \{m_k\})(i = 1, \cdots, \alpha)$.

At first, we prove equation $\alpha(A_i) = \{m \in M \mid gI\,m, \forall g \in A_i\} = (\overline{B_i} - \overline{m_k}) \cup \{m_k\}$.

Let $m_l \in \alpha(A_i)$; then, for every $g \in A_i$, we have gIm_l. If $m_l = m_k$, then certainly $m_l \in (\overline{B_i} - \overline{m_k}) \cup \{m_k\}$. If $m_l \neq m_k$, then from that $(A_i, \overline{B_i})$ is a concept and gIm_q for every $g \in A_i$, we have $m_l \in \overline{B_i}$. Hence, $m_l \in \overline{B_i} - \overline{m_k}$. This shows that $\alpha(A_i) = \{m \in M \mid gIm, \forall g \in A_i\} \subseteq (\overline{B_i} - \overline{m_k}) \cup \{m_k\}$.

Conversely, let $m_l \in (\overline{B_i} - \overline{m_k}) \cup \{m_k\}$. If $m_l \in \{m_k\}$, namely, $m_l = m_k$ is a layered attribute, then by $(A_i, \overline{B_i})$ being a concept, we obtain that

$$\overline{B_i} = \alpha(A_i) = \{(m_k, m_{ki}) \in N \mid gJ(m_k, m_{ki}), \forall g \in A_i\}. \tag{11}$$

Hence, by $gJ(m_k, m_{ki}) \Longleftrightarrow gIm_k$, we know $m_q \in \alpha(A_i)$. If $m_q \in \overline{B_i} - \overline{m_k}$, then $m_q \neq m_k$ and $m_q \in \overline{B_i}$. Since $(A_i, \overline{B_i})$ is a concept, we have gIm_q for every $g \in A_i$; hence, $m_q \in \alpha(A_i)$. Therefore,

$$\alpha(A_i) = \{m \in M \mid gIm, \forall g \in A_i\} \supseteq (\overline{B_i} - \overline{m_k}) \cup \{m_k\}. \tag{12}$$

Secondly, we prove the following equation:

$$\beta((\overline{B_i} - \overline{m_k}) \cup \{m_k\}) = \{g \in G \mid gIm, \forall m \in (\overline{B_i} - \overline{m_k}) \cup \{m_k\}\} = A_i. \tag{13}$$

Let $g \in \beta((\overline{B_i} - \overline{m_k}) \cup \{m_k\})$; then, gIm for every $m \in (\overline{B_i} - \overline{m_k}) \cup \{m_k\}$. Since $(A_i, \overline{B_i})$ is a concept, we know $\beta(\overline{B_i}) = \{g \in G \mid gJm, \forall m \in \overline{B_i}\} = A_i$. Hence, by $gJ(m_k, m_{ki}) \Longleftrightarrow gIm_k$, we have

$$\beta((\overline{B_i} - \overline{m_k}) \cup \{m_k\}) = \{g \in G \mid gIm, \forall m \in (\overline{B_i} - \overline{m_k}) \cup \{m_k\}\} \subseteq A_i. \tag{14}$$

Conversely, let $g \in A_i$. Then, since $(A_i, \overline{B_i})$ is a concept, we have $\beta(\overline{B_i}) = \{g \in G \mid gJm, \forall m \in \overline{B_i}\} = A_i$. If $m \in \overline{B_i}$ is not a layered attribute, then the relation between g and m does not change before and after rolling up. If $m \in \overline{B_i}$ is a layered attribute and its form is (m_k, m_{kj}), then by $gJ(m_k, m_{ki}) \Longleftrightarrow gIm_k$, we have gIm for each $m \in (\overline{B_i} - \overline{m_k}) \cup \{m_k\}$. Therefore,

$$A_i \subseteq \beta((\overline{B_i} - \overline{m_k}) \cup \{m_k\}) = \{g \in G \mid gIm, \forall m \in (\overline{B_i} - \overline{m_k}) \cup \{m_k\}\}. \tag{15}$$

(4) Prove that the partial order among $(A_i, (\overline{B_i} - \overline{m_k}) \cup \{m_k\})(i = 1, \cdots, \alpha)$ is the same $(A_i, \overline{B_i})(i = 1, \cdots, \alpha)$.

By the structure of $(A^{(k)}, B^{(k)})$, for $i = 1, \cdots, \alpha$, we have

$$\left(A_i, \left(\overline{B_i} - \overline{m_k}\right) \cup \{m_k\}\right) \le \left(A^{(k)}, B^{(k)}\right). \tag{16}$$

If two concepts $(A_i, \overline{B_i})$ and $(A_j, \overline{B_j})$ are in $C(\overline{K})$ with order $(A_i, \overline{B_i}) \le (A_j, \overline{B_j})$, then the following also holds:

$$\left(A_i, \left(\overline{B_i} - \overline{m_k}\right) \cup \{m_k\}\right) \le \left(A_j, \left(\overline{B_j} - \overline{m_k}\right) \cup \{m_k\}\right). \tag{17}$$

□

3.2. The Description and Analysis of Roll-Up Algorithm. In order to simplify the presentation below, we introduce the following two terms: parallel lower subconcepts and linear lower subconcepts. Let $(A_1, \overline{B_1}), \cdots, (A_\alpha, \overline{B_\alpha})$ be the lower subconcepts under node concept (A, B). If there is no partial order relation between each other, then $(A_1, \overline{B_1}), \cdots, (A_\alpha, \overline{B_\alpha})$ is called parallel lower subconcepts. If there is linear partial order relation among them, then $(A_1, \overline{B_1})$, $(A_1, \overline{B_1}), \cdots, (A_\alpha, \overline{B_\alpha})$ is called linear lower subconcepts.

Because this paper only discusses the normal layered formal context and in a normal layered formal context an object will not have two lower attributes of one layer attribute, a concept will not contain more than two lower attributes except the empty concept (object set is empty set). Hence, every concept in the linear lower subconcepts only contains the same lower attribute, and different attributes among different subconcepts are nonlayer attributes. On the basis of this analysis, we can present a roll-up building algorithm of building a concept lattice based on attribute fusion. The process of rolling up from $C(\overline{K})$ to $C(K)$ involves the fusion of attributes, and the partial order in the concept lattice is defined according to the negative inclusion of the attribute subset. Thus, the rolling up is carried out from top to bottom, that is, start with the concept of having fewer attributes. The method of rolling up is to roll up the lower concept of node concept (A, B) in the lower concept lattice $C(\overline{K})$ in a linear or parallel lower subconcepts, insert the new concept obtained from rolling up into the bottom of node (A, B), and update the node concept (A, B) and its related node concept accordingly (given in Algorithm 1).

The time complexity of Algorithm 1 is related to the number of layered attributes and the number of lower attributes in the layered context. In the following, we assume that the number of layered attributes in K is l, the number of node in $C(\overline{K})$ is s, and the maximum number of lower attributes of all layered attributes is r. The computing number at Step 1 is r, and its time complexity is $O(r)$. The time complexity at Step 2 is $O(r \log_2 r)$; it is equal to the time complexity of r concept sorting. At Step 3, in the worst case, the number of linear lower subconcepts under a node in $C_k(\overline{K})$ is l, and the number of computing in rolling up is l, but the total number of computing of all nodes in rolling up is not more than s. Hence, the time complexity at Step 3 is $O(s)$. The time complexity of Step 4 and Step 5 is the same as that of Step 3, which is also $O(s)$. The computing number at Step 6 does not exceed $s - r$, so the time complexity at Step 6 is $O(s - r)$. The time complexity analysis at Step 7 has been included from Step 3 to Step 6. Therefore, the time complexity of Algorithm 1 is $O((r + r\log_2 r + \ s + (s - r))l) = O(l(2s + r\log_2 r))$.

The following example illustrates the application of the roll-up building algorithm in building concept lattice.

Example 2. Consider the layered formal context K as shown in example 1; the induced formal context by K is shown in Figure 1.

Steps 1 and 2. The concepts containing lower attributes in $C(\overline{K})$ are arranged in ascending order according to the number of attributes of concepts, $C_k(\overline{K}) = \{(35, (c, c_1)), (46, (c, c_2)), (5, ab(c, c_1)), (6, ab(c, c_2)), (\{\}, ab(c, c_1)(c, c_2))\}$.

Step 3. There are no linear lower subconcepts in this example.

Step 4. The concept of parallel lower concepts $(35, (c, c_1))$, $(46, (c, c_2))$ below node concept $(U, \{\})$ is rolled up into a new concept $(3456, c)$ and inserted below node $(U, \{\})$. The upper bound node of the new node concept $(3456, c)$ and the node concepts $(156, a)$, $(256, b)$ which do not contain lower attributes are also $(U, \{\})$.

The parallel lower concept $(5, ab(c, c_1))$, $(6, ab(c, c_2))$ below node concept $(56, ab)$ is rolled up into a new concept $(56, abc)$.

Step 5. After Step 4 is completed, there is no lower concept to be rolled up. There is only one layered attribute in this example, and the algorithm ends.

Figure 2 shows the concept lattice $C(K)$ obtained after the rolling up.

4. Drill-Down Algorithm to Build the Lower Concept Lattice

In this section, a drill-down algorithm to build lower layered concept lattice $C(\overline{K})$ based on the upper concept lattice $C(K)$ is presented.

4.1. Drill-Down Algorithm Theory

Definition 6. Let $K = (G, M, I)$ be a layered formal context and $b \in M$. Then, $(\beta(b), \alpha(\beta(b)))$ is called an attribute concept.

Proposition 3. *$(\beta(b), \alpha(\beta(b)))$ is the largest concept containing attribute b.*

Proof. Obviously, $(\beta(b), \alpha(\beta(b)))$ is a concept containing attribute b. If (A, B) is another concept containing attribute b, then by $\{b\} \subseteq B$ and Proposition 1, we have $\beta(b) \supseteq \beta(B) = A$. Hence, $(A, B) \le (\beta(b), \alpha(\beta(b)))$. □

Theorem 3. *Let $K = (G, M, I)$ be a layered formal context and $\overline{K} = (G, N, J)$ be the lower formal context. Suppose that m_k is a layered attribute; then,*

$$\overline{m_k} = \left\{(m_k, m_{k1}), (m_k, m_{k2}), \cdots, \left(m_k, m_{ki_k}\right)\right\}, \tag{18}$$

Input: a lower concept lattice $C(\overline{K})$ of layered formal context $\overline{K}$ induced by formal context K.
Output: an upper concept lattice.
Step 1. For $k = 1, 2, \ldots, l$, select all lower concepts containing layered attribute $\overline{m_k}$ and form lower concept set $C_k(\overline{K})$ to be rolled up.
Step 2. The concepts in $C_k(\overline{K})$ are sorted in ascending order according to the number of lower attributes contained within it. When the number of lower attributes is the same, the concepts containing fewer nonlower attributes are ranked ahead.
Step 3. From top to bottom, roll up the lower concepts in $C(\overline{K})$. For the linear lower subconcepts under node concept (A, B) in $C_k(\overline{K})$, the rolling-up method is as follows: combine the linear lower concepts $(A_1, \overline{B_1}), \ldots, (A_\beta, \overline{B_\beta})$ containing $\overline{m_k}$ into a new concept $(A^{(k)}, B^{(k)})$, where $A^{(k)} = A_1 \cap \cdots \cap A_\beta$, $B^{(k)} = ((\overline{B_1} - \overline{m_k}) \cup \cdots \cup (\overline{B_\beta} - \overline{m_k})) \cup \{m_k\}$, and insert $(A^{(k)}, B^{(k)})$ to node (A, B) below. If node concept (A, B) contains subconcepts with nonlower attributes, then node (A, B) is updated to be the common upper bound of $(A^{(k)}, B^{(k)})$ and other concepts (excluding lower attributes in $\overline{m_k}$) under node (A, B).
Step 4. For the parallel lower subconcepts and lower subconcepts under node concept (A, B) in $C_k(\overline{K})$, the rolling-up method is as follows: combine the parallel lower concepts $(A_1, \overline{B_1}), \ldots, (A_\beta, \overline{B_\beta})$ containing $\overline{m_k}$ into a new concept $(A^{(k)}, B^{(k)})$, where $A^{(k)} = A_1 \cup \cdots \cup A_\alpha$, $B^{(k)} = ((\overline{B_1} - \overline{m_k}) \cap \cdots \cap (\overline{B_\alpha} - \overline{m_k})) \cup \{m_k\}$, and insert $(A^{(k)}, B^{(k)})$ to node (A, B) below. If node concept (A, B) contains subconcepts with nonlower attributes, then node (A, B) is updated to be the common upper bound of $(A^{(k)}, B^{(k)})$ and other concepts (excluding lower attributes in $\overline{m_k}$) under node (A, B).
Step 5. If the node concept (A, B) has both the parallel lower subconcepts and the linear lower subconcepts, then linear lower subconcepts are fused firstly; the method of fusion is to replace the linear lower subconcepts with the minimum concept in these linear lower subconcepts and then roll up the parallel lower subconcepts by Step 4 secondly.
Step 6. Update the subconcept $(C_j, \overline{D_j})$, which is not rolled up from Step 3 to Step 5 but it contains the lower attribute of $\overline{m_k}$, to a new concept $(C_j, (\overline{D_j} - \overline{m_k}) \cup \{m_k\})$.
Step 7. Repeat from Step 3 to Step 6 until the concept in $C(\overline{K})$ does not contain the lower attribute of $\overline{m_k}$ (namely, $C_k(\overline{K})$ is an empty set).
Step 8. Repeat from Step 1 to Step 7 until $C(\overline{K})$ does not contain any layered attribute. Algorithm is completed.

ALGORITHM 1: Roll-up building algorithm of layered concept lattice.

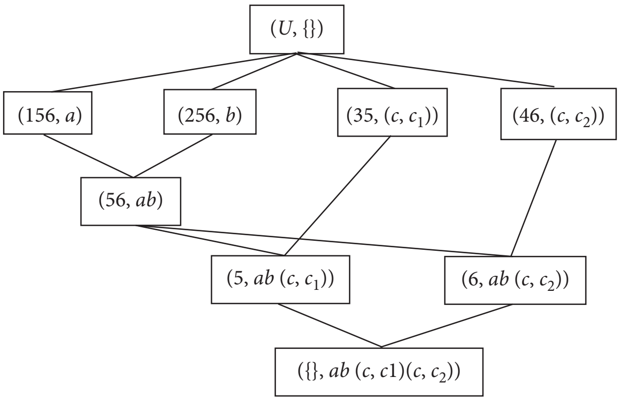

FIGURE 1: Concept lattice $C(\overline{K})$ of formal context $\overline{K}$.

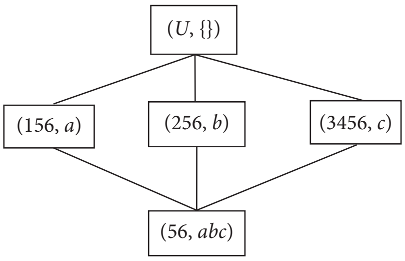

FIGURE 2: Concept lattice $C(K)$ after rolling up.

is the lower attribute corresponding to m_k, and $(\beta(m_k, m_{kj}), \alpha(\beta(m_k, m_{kj})))$ are attribute concepts corresponding to the lower attributes $(m_k, m_{kj})\ (j = 1, 2, \cdots, i_k)$. $(A, B) \in C(K)$ is a concept and for every $j \in \{1, 2, \cdots, i_k\}$, there is no child (X, Y) of (A, B) with $A \cap \beta(m_k, m_{kj}) \subseteq X$; then, for every $A \cap \beta(m_k, m_{kj})$, $j \in \{1, 2, \cdots, i_k\}$,

(1) If $A \cap \beta(m_k, m_{kj}) = A$, then (A, B) is updated to $(A, (B - \{m_k\}) \cup \{(m_k, m_{kj})\})$

(2) If $A \cap \beta(m_k, m_{kj}) \neq A$, then a new concept $(A \cap \beta(m_k, m_{kj}), (B - \{m_k\}) \cup \{(m_k, m_{kj})\})$ is added as a child of (A, B)

Proof

(1) is a special case of (2), so we only prove (2).

Firstly, we prove that if $A \cap \beta(m_k, m_{kj}) \neq \varphi$, then $(A \cap \beta(m_k, m_{kj}), (B - \{m_k\}) \cup \{(m_k, m_{kj})\})$ are concepts $(j = 1, 2, \cdots, i_k)$; hence, we need to prove two equations:

$$\begin{aligned}\alpha\big(A \cap \beta(m_k, m_{kj})\big) &= \{m \in N \mid gJm, \forall g \in A \cap \beta(m_k, m_{kj})\} \\ &= (B - \{m_k\}) \cup \{(m_k, m_{kj})\}),\end{aligned} \tag{19}$$

$$\begin{aligned}\beta\big((B - \{m_k\}) \cup \{(m_k, m_{kj})\}\big) &= \{g \in G \mid gJm, \forall m \in (B \\ &\quad - \{m_k\}) \cup \{(m_k, m_{kj})\}\} \\ &= A \cap \beta(m_k, m_{kj}).\end{aligned} \tag{20}$$

Since (A, B) is a concept, we know $\alpha(A) = B$. By Proposition 1, we have that

$$\begin{aligned}\alpha\big(A \cap \beta(m_k, m_{kj})\big) &\supseteq \alpha(A) \cup \alpha\big(\beta(m_k, m_{kj})\big) \\ &= B \cup \alpha\big(\beta(m_k, m_{kj})\big) \\ &\supseteq (B - \{m_k\}) \cup \alpha\big(\beta(m_k, m_{kj})\big).\end{aligned} \tag{21}$$

Conversely, let $m_q \in \alpha(A \cap \beta(m_k, m_{kj}))$. Then, for each $g \in A \cap \beta(m_k, m_{kj})$, we have gJm_q. In the following, we discuss it in two cases:

(i) $m_q \in \overline{m_k}$: if $m_q \neq (m_k, m_{kj})$, then there is $i \neq j$ such that $m_q = (m_k, m_{ki})$. By Proposition 2 $\beta(m_k, m_{k1}), \cdots, \beta(m_k, m_{ki_k})\}$ is a partition of $\beta(m_k)$, we know that for every $g \in \beta(m_k, m_{kj})$, g and (m_k, m_{ki}) have no relation J. This contradicts gJm_q for each $g \in A \cap \beta(m_k, m_{kj})$. Thus, $m_q \in \{(m_k, m_{kj})\}$.

(ii) $m_q \notin \overline{m_k}$: if $m_q \notin B - \{m_k\}$, then obviously $m_q \neq m_k$, and hence $m_q \notin B$. Because (A, B) is a concept containing attribute m_k, there exists $g \in A$ such that g and m_q have no relation J. The set of all elements of A that has no relation J with m_q is denoted as A_0. If $A_0 \cap \beta(m_k, m_{kj})$ is an empty set, then for any $g \in \beta(m_k, m_{kj})$ and $g \in A$, g and m_q have no relation J. This contradicts gJm_q for each $g \in A \cap \beta(m_k, m_{kj})$. Hence, there is $g_0 \in A_0 \cap \beta(m_k, m_{kj})$ such that g_0 and m_q have no relation J. This also contradicts gJm_q for each $g \in A \cap \beta(m_k, m_{kj})$. Therefore, if $m_q \notin \overline{m_k}$, then $m_q \in B - \{m_k\}$.

This proves that

$$\alpha\left(A \cap \beta\left(m_k, m_{kj}\right)\right) = \left(B - \{m_k\}\right) \cup \left\{\left(m_k, m_{kj}\right)\right\}. \tag{22}$$

Since (A, B) is a concept, we have $\beta(B) = A$. By Proposition 1 we obtain

$$\begin{aligned}\beta\left(\left(B - \{m_k\}\right) \cup \left\{\left(m_k, m_{kj}\right)\right\}\right) &= \beta\left(B - \{m_k\}\right) \cap \beta\left(m_k, m_{kj}\right) \\ &\supseteq \beta(B) \cap \beta\left(\left(m_k, m_{kj}\right)\right) \\ &= A \cap \beta\left(m_k, m_{kj}\right).\end{aligned} \tag{23}$$

Conversely, let

$$g \in \beta\left(\left(B - \{m_k\}\right) \cup \left\{\left(m_k, m_{kj}\right)\right\}\right) = \beta\left(B - \{m_k\}\right) \cap \beta\left(m_k, m_{kj}\right). \tag{24}$$

From $g \in \beta(m_k, m_{kj})$, we know $gJ(m_k, m_{kj})$. By $gIm_k \Longleftrightarrow gJ(m_k, m_{kj})$, we have gIm_k. From $g \in \beta(B - \{m_k\})$, we know gIm for each $m \in B - \{m_k\}$. Hence, gIm for every $m \in B$. This shows that $g \in \beta(B) = A$, and hence $g \in A \cap \beta(m_k, m_{kj})$. Therefore,

$$\beta\left(\left(B - \{m_k\}\right) \cup \left\{\left(m_k, m_{kj}\right)\right\}\right) \subseteq A \cap \beta\left(m_k, m_{kj}\right). \tag{25}$$

This proves that

$$\beta\left(\left(B - \{m_k\}\right) \cup \left\{\left(m_k, m_{kj}\right)\right\}\right) = A \cap \beta\left(m_k, m_{kj}\right). \tag{26}$$

Next, we prove that the partial order among concepts $(A_i, \overline{B_i})(i = 1, \cdots, \alpha)$ is the same as the partial order among $(A_i, (\overline{B_i} - \overline{m_k}) \cup \{m_k\})(i = 1, \cdots, \alpha)$.

At first, for $i = 1, \cdots, \alpha$, by the construction of $(A^{(k)}, B^{(k)})$, we have

$$\left(A_i, \left(\overline{B_i} - \overline{m_k}\right) \cup \{m_k\}\right) \leq \left(A^{(k)}, B^{(k)}\right). \tag{27}$$

Secondly, if $(A_i, \overline{B_i}) \leq (A_j, \overline{B_j})$ in $C(\overline{K})$, then we also have

$$\left(A_i, \left(\overline{B_i} - \overline{m_k}\right) \cup \{m_k\}\right) \leq \left(A_j, \left(\overline{B_j} - \overline{m_k}\right) \cup \{m_k\}\right). \tag{28}$$

□

4.2. The Description and Analysis of Drill-Down Algorithm. According to Theorem 3, started with the upper concept lattice $C(K)$, inserted some new concepts, modified some concepts and deleted some concepts, we can obtained lower concept lattice $C(\overline{K})$. In the following, we propose a drill-down algorithm to build lower concept lattice based on attribute decomposition (given in Algorithm 2).

The time complexity of Algorithm 2 is analyzed below. In the following, we assume that the number of layered attributes in K is l, the number of node concepts in $C(K)$ is s, and the maximum number of lower attributes is r. The Bordat algorithm building concept lattice of formal context (G, M, I) is improved by Chen in [12], and the time complexity of the improved algorithm is less than $O(|G||M|^2)$. Then, in the worst case, the time complexity of Step 1 is $O(|G|r^2)$. The time complexity of Step 2 is $O(sr)$. The time complexity of Step 3 is $O(r^2)$. Therefore, the time complexity of algorithm 2 is $O((|G|r^2 + sr + r^2)l) = O(l((|G| + 1)r^2 + sr))$.

The following example illustrates the application of the drill-down building algorithm in building the concept lattice.

Example 3. A layered formal context $K = (G, M, I)$ is shown in Table 3, and the concept lattice of K is shown in Figure 3. The lower formal context $\overline{K} = (G, N, J)$ induced by K is shown in Table 4.

The concept lattice of $\overline{K}$ is shown in Figure 4.

Step 1. Layered attribute c has three lower attributes, and the corresponding lower attribute concepts are $(1256, (c, c_1))$, $(3, b(c, c_2))$, $(4, b(c, c_3))$.

Step 2. There are four concepts in $C(K)$ that contain the layered attribute c. According to Step 2 in Algorithm 2, (U, c) is updated into three new concepts $(1256, (c, c_1))$, $(3, b(c, c_2))$, $(4, b(c, c_3))$; $(34, bc)$ is updated into two new concepts $(3, b(c, c_2))$, $(4, b(c, c_3))$; $(156, ac)$ is updated into a new concept $(156, a(c, c_1))$.

Step 3. Two lower attribute concepts $(3, b(c, c_2))$, $(4, b(c, c_3))$ can be combined into a new concept $(34, b)$ and inserted above $(3, b(c, c_2))$, $(4, b(c, c_3))$.

Step 4. There is only one layered attribute, and the algorithm ends.

The purpose of the roll-up building algorithm is to build the upper concept lattice $C(K)$ on the basis of lower concept lattice $C(\overline{K})$, and the drill-down building algorithm is to build the lower concept lattice $C(\overline{K})$ on the basis of the upper concept lattice $C(K)$. From the point of view of granular computing, roll-up building algorithm helps us recognize concepts from fine-grained to coarse-grained, while drill-down building algorithm help us recognize concepts from coarse-grained to fine-grained. Therefore, from the human cognitive thinking process, roll-up and drill-own building algorithms are reversed, and then there is a question to be discussed: whether there is reducibility between these two reverse algorithms.

Now, we consider example 3.

Input: a concept lattice $C(K)$ of formal context K containing layered attributes.
Output: the lower concept lattice $C(\overline{K})$.
Step 1. For each lower attribute of a layered attribute m_k, $k \in \{1, 2, \ldots, l\}$: $\overline{m_k} = \{(m_k, m_{k1}), (m_k, m_{k2}), \ldots, (m_k, m_{ki_k})\}$, build all attribute concepts: $(A \cap \beta(m_k, m_{kj}), (B - \{m_k\}) \cup \alpha(\beta(m_k, m_{kj})))\,(j = 1, 2, \ldots, i_k)$.
Step 2. The concept (A, B) containing layered attribute m_k is updated to some new concepts $(A \cap \beta(m_k, m_{kj}), (B - \{m_k\}) \cup \alpha(\beta(m_k, m_{kj})))$, which contain lower attributes (m_k, m_{kj}), $j = 1, \ldots, i_k$, respectively, where $A \cap \beta(m_k, m_{kj})$ is not an empty set. The concepts that do not contain the layered attribute m_k remain the same.
Step 3. If some concepts with lower attributes obtained from Step 2 can be combined, then these concepts are combined and a concept and is inserted above them.
Step 4. The node concepts that are connected to (A, B) and are also not updated in $C(K)$ are connected to the updated node concepts.
Step 5. For each layered attribute $m_k\,(k = 1, \ldots, l)$, repeat from Step 1 to Step 4 above.
Step 6. End of the algorithm.

Algorithm 2: Drill-down algorithm to build the lower concept lattice.

Table 3: Layered formal context K.

G	a	b	c		
			c_1	c_2	c_3
1	1	0	1	0	0
2	0	0	1	0	0
3	0	1	0	1	0
4	0	1	0	0	1
5	1	0	1	0	0
6	1	0	1	0	0

Table 4: Formal context $\overline{K}$ induced by K.

G	a	b	(c, c_1)	(c, c_2)	(c, c_3)
1	1	0	1	0	0
2	0	0	1	0	0
3	0	1	0	1	0
4	0	1	0	0	1
5	1	0	1	0	0
6	1	0	1	0	0

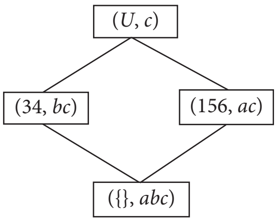

Figure 3: Concept lattice $C(K)$ of context K.

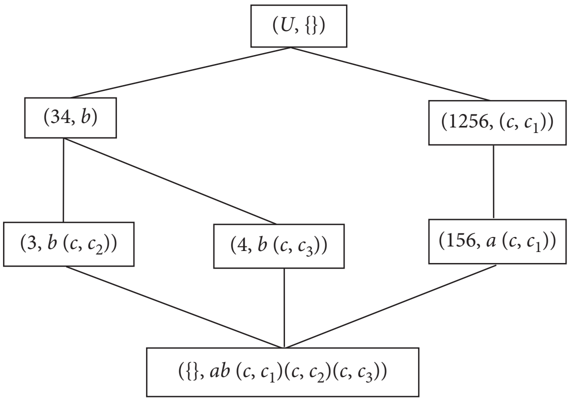

Figure 4: Concept lattice $C(\overline{K})$ of formal context $\overline{K}$.

Example 4. In the layered formal context of example 3, by using the drill-down algorithm, we obtain the lower concept lattice $C(\overline{K})$ as shown in Figure 4. Now, we run the roll-up building algorithm on $C(\overline{K})$.

Steps 1 and 2. The concepts containing lower attributes in $C(\overline{K})$ are arranged in ascending order according to the number of attributes contained in the concepts as follows: $C'(\overline{K}) = \{1256, (c, c_1), (3, b(c, c_2)), (4, b(c, c_3)), (156, a\ \ (c, c_1)), (\{\}, ab(c, c_1)(c, c_2)(c, c_3))\}$.

Step 3. The linear lower subconcepts $(1256, (c, c_1))$, $(156, a(c, c_1))$ below node concept $(U, \{\})$ are rolled up into a new concept $(156, ac)$, and the node concept $(U, \{\})$ is replaced by (U, c), which are the parent concept (upper bound node) of the new concept $(156, ac)$ and the concept $(34, b)$ that does not contain lower attributes.

Step 4. The parallel lower subconcepts $(3, b(c, c_2))$, $(4, b(c, c_3))$ below node concept $(34, b)$ are rolled up into a new concept $(34, bc)$. There are no other concepts below $(34, b)$ that do not contain lower attributes (c, c_2), (c, c_3), and then $(34, b)$ is updated into $(34, bc)$.

Step 5. The lower attributes (c, c_1), (c, c_2), (c, c_3) in the minimum node concept (empty concept) $(\{\}, ab(c, c_1)(c, c_2)(c, c_3))$ are integrated, and then we obtain the minimum concept $(\{\}, abc)$.

Step 6. There is only one layered attribute, and the algorithm ends.

After $C(\overline{K})$ is rolled up, the concept lattice is exactly the same as shown in Figure 3. This also shows that the rolling-up and drilling-down algorithms for layered concept lattice have the reducibility.

5. Experimentation

The previous example has shown that the roll-up building algorithm and the drill-down building algorithm proposed in this paper are effective. To further show the utility of Algorithms 1 and 2, we implement two algorithms by using MTLAB program language. The computing environment is a PC (Pentium Win7 × 64, Intel (R) 3.4 GHz, RAM 4 GB). The validity verification about the roll-up building algorithm is mainly to investigate the improvement on time consumption by comparing building the concept lattice $C(K)$ from the lower concept lattice $C(\overline{K})$ with building the concept lattice $C(K)$ directly from the formal context K.

In the experiment, the number of objects is set to 15000, and the total number of attributes is set to 60 (including lower attributes). The number of lower attributes of each layered attribute is set to 5, the number of layered attributes is 2, 4, 6, 8, and 10, respectively, and the total number of lower attributes correspondingly is 10, 20, 30, 40, and 50, respectively, so the number of upper attributes in the lower form context $\overline{K}$ is 52, 44, 36, 28, and 20, respectively. The test data with medium strength filling ratio $(|I|/(|G||M|))$ 20% are randomly generated. If a layered attribute has no filling value, then its corresponding lower attribute values are all 0. If a layered attribute has filling value, then select randomly a lower attribute to be assigned a value of 1. The improved Bordat algorithm proposed in [12] was adopted for building context lattice directly. The comparison on time consumption building directly the concept lattice from the dataset with building the concept lattice by using roll-up building algorithm is shown in Figure 5, and the time saved by using the roll-up building algorithm to build the concept lattice is shown in Figure 6.

It can be seen from Figures 5 and 6 that using the roll-up building algorithm to build concept lattice $C(K)$ on the basis of $C(\overline{K})$ takes less time than building directly the concept lattice $C(K)$ from the formal context K. When the number of layered attributes is relatively small, the effect of time saving by using roll-up building algorithm is very obvious, and as the number of layered attributes increases, the time saving decreases.

The validity verification about the drill-down building algorithm is mainly to investigate the improvement on time consumption by comparing building the lower concept lattice $C(\overline{K})$ from the upper concept lattice $C(K)$ with building the concept lattice $C(\overline{K})$ directly from the formal context $\overline{K}$.

In the experiment, the number of objects is set to 15000, and the total number of attributes is set to 60 (including lower attributes). The number of nonlayered attributes is set to 10, the number of layered attributes is set to 2, 4, 6, 8, and 10, respectively, and the number of lower attributes of each layered attribute is set to 5, so the total number of attributes in the lower formal context $\overline{K}$ is 20, 30, 40, 50, and 60, respectively. The experimental environment and dataset are the same as the roll-up building algorithm experiment. The comparison on time consumption building directly the concept lattice from the dataset with building the concept lattice by using drill-down building algorithm is shown in Figure 7, and the time saved by using the drill-down building algorithm to build the concept lattice is shown in Figure 8.

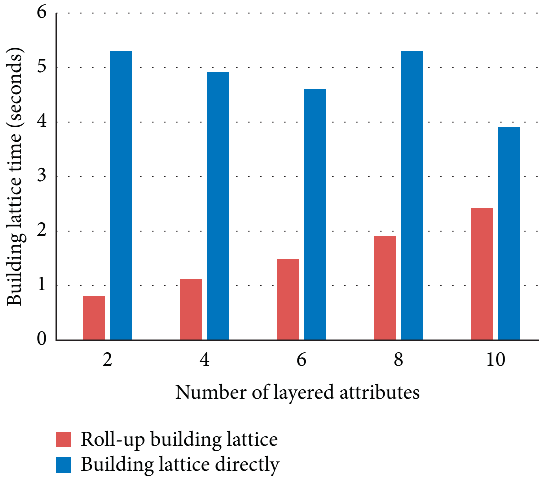

FIGURE 5: Time comparison between building concept lattice by using roll-up algorithm and building concept lattice directly.

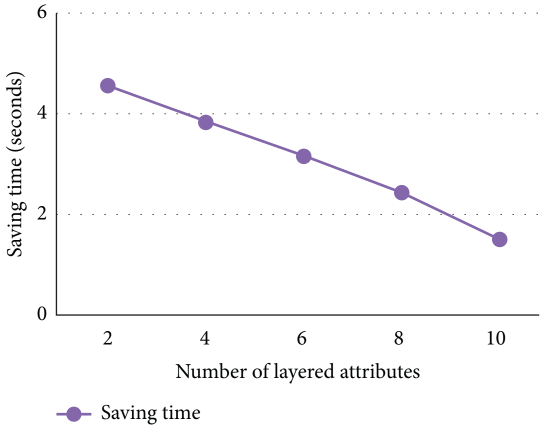

FIGURE 6: Saving time of roll-up building lattice.

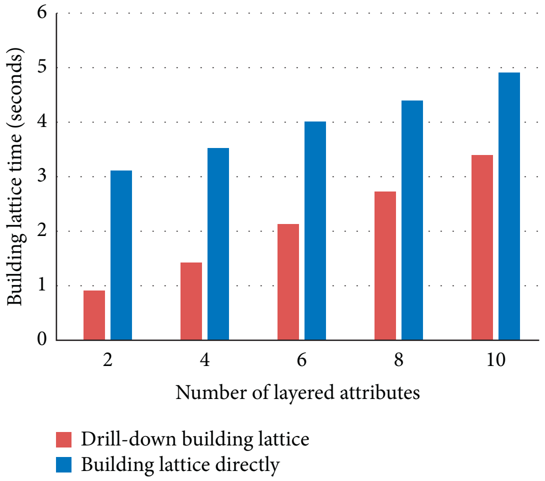

FIGURE 7: Time comparison between building concept lattice by using drill-down algorithm and building concept lattice directly.

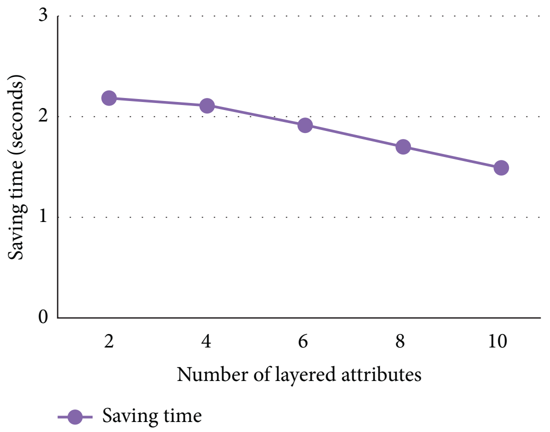

Figure 8: Saving time of drill-down building lattice.

It can be seen from Figures 7 and 8 that using the drill-down building algorithm to build concept lattice $C(\overline{K})$ on the basis of $C(K)$ takes less time than building directly the concept lattice $C(\overline{K})$ from the formal context $\overline{K}$. When the number of layered attributes is relatively small, the effect of time saving by using drill-down building algorithm is very obvious, and as the number of layered attributes increases, the time saving decreases.

6. Conclusion

This paper applies the idea of granular computing to build concept lattice in a formal context with complex structure attribute data. When some attributes in a formal context are made up of some subattributes, a layered concept lattice model is established, and the relation between the upper concept lattice based on the original formal context and the lower concept lattice based on the lower formal context is discussed. A roll-up building algorithm that builds the upper concept lattice from the lower concept lattice and a drill-down building algorithm that builds the lower concept lattice from the upper concept lattice are proposed; the time complexity of two algorithms is analyzed. The application of the algorithm and the reducibility of the roll-up building algorithm and the drill-down building algorithm are illustrated by some practical examples. Practical examples and experiments show that the layered concept lattice model can be used to model complex structural data, and the roll-up building algorithm and the drill-down building algorithm proposed in this paper are effective. In future work, we will discuss in detail the reducibility between the roll-up building algorithm and the drill-down building algorithm, the roll-up building algorithm and the drill-down building algorithm for the layered concept lattice in a not-normal layered context. This paper does not discuss the relationship between layered attributes from a quantitative point of view. It also may be another possible research direction to use the analytic hierarchy process to discuss the layered concepts in layered formal context from a quantitative point of view.

Data Availability

The data used to support the findings of this study are randomly available according to the method described in the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

This study was supported in part by the Key Science Research Project of Hunan Province Education Department (no. 19A463), the Natural Science Foundation of Hunan Province (no. 2018JJ2371 and no. 2020JJ2251), and the Science and Technology Development Project of Chenzhou (no. zdyf201807).

References

[1] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum, “Human-level concept learning through probabilistic program induction,” *Science*, vol. 350, no. 6266, pp. 1332–1338, 2015.

[2] S. W. Hou, H. J. Wen, S. X. Feng, H. Wang, and L. Zhibin, “Application of layered coding genetic algorithm in optimization of unequal area production facilities layout,” *Computational Intelligence and Neuro-Science*, vol. 2019, Article ID 3650923, 17 pages, 2019.

[3] Y. Y. Yao, “Granular computing: basic issues and possible Solutions,” in *Proceedings the 5th Joint Conference on Information Sciences*, pp. 186–189, Atlantic City, NJ, USA, March 2000.

[4] W. X. Zhang, Y. Liang, and W. Z. Wu, *Rough Sets and Concept Lattice*, Xian Jiao Tong University Press, Xian, China, 2006.

[5] X. Wu, J. Zhang, and R. Lu, “Attribute logic formula description of granule and its application to build concept lattice,” *IEEE Access*, vol. 8, pp. 12592–12606, 2020.

[6] L. Zadeh, “Fuzzy sets and information granularity,” *Advances in Fuzzy Set Theory and Applications*, pp. 3–18, World Scientific, Singapore, 1979.

[7] T. Y. Lin, *Granular Computing on Binary Relations II: Rough Set Representations and Belief Functions, Rough Sets in Knowledge Discovery*, pp. 121–140, Physica-Verlag, Berlin, Germany, 1998.

[8] W. Pedrycz, *Granular Computing: An Emerging Paradigm*, Physica-Verlag, Berlin, Germany, 2001.

[9] B. Zhang and L. Zhang, *Theory and Application of Problem Solving*, Tsinghua University Press, Beijing, China, 2005.

[10] B. Ganter and R. Wille, *Formal Concept Analysis Mathematical Foundations*, Springer, Berlin, Germany, 1999.

[11] W. H. Xu, J. H. Li, L. Wei et al., *Theory and Application of Formal Concept Analysis*, Science Press, Beijing, China, 2016.

[12] Q. Y. Chen, “Improvement on Bordat algorithm for building concept lattice,” *Computer Engineering and Application*, vol. 46, no. 35, pp. 33–35, 2010.

[13] Y.-M. Chai, Z. Zhang, and L.-M. Wang, “An algorithm for mining global closed frequent itemsets based on distributed frequent concept direct product,” *Chinese Journal of Computers*, vol. 35, no. 5, pp. 990–1001, 2012.

[14] S. Y. Fan, L. M. Wang, Q. Jiang et al., “Research of distributed integration algorithm on concept lattice,” *Computer Science*, vol. 43, no. 1, pp. 223–278, 2016.

[15] Z. Zhang, *Research on Web Base Extraction Based on Formal Concept Analysis*, Wuhan University, Wuhan, China, 2011.

[16] H. L. Zhi, “Knowledge representation on incomplete formal context,” *Computer Science*, vol. 42, no. 1, pp. 276–278, 2015.

[17] Q. Li and L. Guo, “Formal query systems on contexts and a representation of algebraic lattices,” *Information Sciences*, vol. 239, no. 4, pp. 72–84, 2013.

[18] J. Xie, M. Yang, J. Li, and Z. Zheng, “Rule acquisition and optimal scale selection in multi-scale formal decision contexts and their applications to smart city,” *Future Generation Computer Systems*, vol. 83, pp. 564–581, 2018.

[19] F. Herrera, E. Herrera-Viedma, and L. Martı́nez, “A fusion approach for managing multi-granularity linguistic term sets in decision making,” *Fuzzy Sets and Systems*, vol. 114, no. 1, pp. 43–58, 2000.

[20] K. E. Wolff, “Conceptual interpretation of fuzzy theory,” *The 6th European Congress on Intelligent Techniques and Soft Computing*, vol. 1, pp. 555–562, EUFIT, Aachen, Germany, 1998.

[21] A. Burusco and R. Fuentes, “The study of L-fuzzy concept lattices,” *Mathware & Soft Computing*, vol. 1, no. 3, pp. 209–218, 1994.

[22] Y. Qiang, Z. T. Liu, W. Lin et al., “Research on fuzzy concept lattice in knowledge discovery and a building algorithm,” *Acta Electronica Sinica*, vol. 33, no. 2, pp. 350–353, 2005.

[23] J. Ali and E. Samir, “Galois connection, formal concepts and Galois lattice in real relation application in a real classifier,” *Journal of Systems & Software*, vol. 60, no. 2, pp. 149–163, 2002.

[24] K.-S. Qu, Y. H. Zhai, J. Y. Liang et al., “Representation and extension of rough set theory based on formal concept analysis,” *Journal of Software*, vol. 18, no. 9, pp. 2174–2182, 2007.

[25] W. Liu, W. J. Xu, J. F. Fu et al., “An extended description logic for event ontology,” *Advances in Grid and Pervasive Computing*, Springer, Hualien, Taiwan, pp. 471–481, 2010.

[26] Y. Z. Wang, X. L. Jin, and X. Q. Cheng, “Network big data: present and furture,” *Chinese Journal of Computers*, vol. 36, no. 6, pp. 1125–1138, 2013.

[27] H. Y. Wang and Y. F. Guo, “Application of big data technology in artificial intelligence,” *Digital Technology and Application*, vol. 12, pp. 109-110, 2015.

[28] H. L. Zhi, “Extended model of formal concept analysis oriented for heterogeneous data analysis,” *Acta Electronica Sinica*, vol. 41, no. 12, pp. 2451–2455, 2013.

[29] H. Zhi and J. Li, “Granule description based on formal concept analysis,” *Knowledge-Based Systems*, vol. 104, pp. 62–73, 2016.