Research Article

Deep First Formal Concept Search

Tao Zhang,¹ Hui Li,¹ Wenxue Hong,² Xiamei Yuan,³ and Xinyu Wei¹

¹ School of Information Science and Engineering, Yanshan University, Qinhuangdao 066004, China

² School of Electrical Engineering, Yanshan University, Qinhuangdao 066004, China

³ College of Foreign Languages, Yanshan University, Qinhuangdao 066004, China

Correspondence should be addressed to Tao Zhang; zhtao@ysu.edu.cn

Received 11 June 2014; Accepted 21 July 2014; Published 31 August 2014

Academic Editor: Yunqiang Yin

Copyright © 2014 Tao Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The calculation of formal concepts is a very important part in the theory of formal concept analysis (FCA); however, within the framework of FCA, computing all formal concepts is the main challenge because of its exponential complexity and difficulty in visualizing the calculating process. With the basic idea of Depth First Search, this paper presents a visualization algorithm by the attribute topology of formal context. Limited by the constraints and calculation rules, all concepts are achieved by the visualization global formal concepts searching, based on the topology degenerated with the fixed start and end points, without repetition and omission. This method makes the calculation of formal concepts precise and easy to operate and reflects the integrity of the algorithm, which enables it to be suitable for visualization analysis.

1. Introduction

FCA, a branch of lattice theory, presented by Ganter and Wille [1] in 1982, is a discipline that concerns the mathematical concepts and conceptual hierarchies. As a powerful tool for data analysis and knowledge processing, FCA has been widely applied in data mining [2, 3], web search [4, 5], software engineering [6, 7], ontology analysis [8, 9], and so forth and still has a great potential value in applications.

Computing all formal concepts and the concept lattice is the most basic issue that has been investigated by many domestic and foreign researchers from different angles. Various forms of algorithms in concepts computing and concept lattice generation can be generally divided into three categories: batch processing algorithm [10–14], progressive algorithm [15, 16], and parallel algorithms [17, 18]. The basic idea of batch processing algorithm is to form the edges according to the presubsequent relationship among all concepts generated and then constructs the concept lattice. Progressive algorithm intends to initialize the concept lattice to empty firstly and then updates the concepts according to the different results of the intersection operation between the new concept which is to join to the lattice and all the concepts of the current concept lattice. Parallel algorithm

intends to split the formal context into some subcontexts, based on which to construct sublattices, and then makes the appropriate merge operations. Besides, progressive algorithm [19, 20] and parallel algorithms [21, 22] work very effectively in the calculation of concepts.

However, the structure of concept lattice is relatively complex and the relationship among attributes of formal context cannot be visually represented. Attribute topology [23–25] presented by Zhang et al. is a new method to represent the formal context. Different with the traditional representation of formal context, attribute topology is represented by the weighted graph, with the attributes as the vertices and inclusion relation between attributes as the weights, which performs the coupling relationship and the coupling strength between attributes intuitively. Attribute topology, providing a new method to represent formal context, not only performs the association and the strength of association simply and intuitively but also has a one-to-one relationship with formal context [23].

Based on the new representation, Zhang et al. present a calculation algorithm of formal concepts by attribute topology [24]. The method, firstly, intends to construct the subtopology successively according to the number of

connections with the top-attributes, as the core of the subtopology, respectively, with the order from more to less. Then, all extensions in every subtopology are achieved after the arrangement and operation of all the available object sets according to the association and correlation intensity, and all concepts in every subtopology are generated with the corresponding intensions, that is, all concepts in the formal context. This method calculating concept by the attribute topology not only provides a new idea for the calculation of formal concepts but also enables the calculation simple and easy to operate. However, the method is not suitable for visualization analysis due to the subtopology that is achieved by splitting the whole topology. Secondly, it is not available for the concepts' calculation in the formal context of large-scale data due to the poor visibility.

Global formal concepts searching of attribute topology is proposed in this paper with the basic idea of Depth First Search. This method, firstly, adds two vertices, global start and end points, and edges to the original attribute topology, degenerating it into the topology with the fixed start and end points and then, limited by the constraints and calculation rules, explores and backtracks the vertices, sorted formerly, repetitively, completing the traversal of paths. All formal concepts are achieved by the path traversing between the global start and end points. This method constructs the topology into a complete whole, avoiding the decomposition process of the whole topology, and reflects the integrity of the algorithm. Calculation process is demonstrated intuitively in the process of traversing paths, reflecting the good visibility.

2. Basic Notions

2.1. The Basis of Formal Context. Formal context, which acts as the research object and data representation, is an important basic aspect in FCA. Here are a few notions about formal context [1].

Definition 1. A formal context can be expressed as $K := (G, M, I)$ which is composed of two sets G and M and the relation I between them. Where G is a set of objects, M is a set of attributes, and I is an object-attribute relation.

Definition 2. Let $K := (G, M, I)$ be a formal context, $A \subseteq G, B \subseteq G$; let

$$\begin{aligned} f(A) &= \{m \in M \mid \forall g \in A, (g, m) \in I\}, \\ g(B) &= \{g_0 \in G \mid \forall m \in B, (g_0, m) \in I\}. \end{aligned} \quad (1)$$

If $f(A) = B$ and $g(B) = A$, then the pair (A, B) is a formal concept. The sets A and B are the extension and intension of (A, B) , respectively. The set of all concepts in $K := (G, M, I)$ is denoted by $\beta(G, M, I)$ or $\beta(K)$.

Definition 3. The concept, with the whole attributes as the intension and the corresponding object set as the extension (called global concept of whole attribute) or the whole objects as the extension and the corresponding attribute set as the intension (called global concept of whole object), is called global concept. There exist only two global concepts in $K := (G, M, I)$, that is, $(G, f(G))$ and $(g(M), M)$.

Concept lattice, the core data structure of FCA, is a complete lattice denoted by all concepts and the generalization-specialization relationships between them. Hasse diagram, equipped with the partial order of concept lattice simply and effectively, is the best way and common method to represent the concept lattice, which can express the relationships between all concepts intuitively and integrally.

Every vertex in Hasse diagram is a concept and all vertices respect the whole concepts in formal context. The entire Hasse diagram is constructed by all the concepts together with the order inclusion relation between two formal concepts, and each layer is arranged in descending order of the extension (ascending order of the intension).

Formal contexts mentioned in this paper are all simplified contexts [23–25].

2.2. The Basis of Attribute Topology. From the perspective of graph theory, the attribute topological representation is the weighted graph which concerns the relationship among attributes. So it can use the storage method of graph, that is, adjacency matrix sequence. In [23, 24], attribute topology is represented by adjacency matrix from the perspective of inclusion relation of attributes.

Suppose $T = (V, \text{Edge})$ is the attribute topological representation for formal context $K := (G, M, I)$. $V = M$ is the set of vertices in attribute topology. Edge is the set of weights on edges in attribute topology.

In this paper, adjacency matrix, induced in [23, 24], is streamlined, and the adjacency matrix modified is

$$\begin{aligned} \text{Edge}(m_i, m_j) &= \begin{cases} \Phi, & g(m_i) \cap g(m_j) \\ g(m_i) \cap g(m_j), & \text{others.} \end{cases} \quad (2) \end{aligned}$$

Here are a few notions about attribute topology [22–25].

Definition 4. Global attribute is the attribute that possess all objects. In the formal context $K := (G, M, I)$, if $m \in M$ and $g(m) = G$, then attribute m is a global attribute. Correspondingly, global object is the object that possesses all attributes.

Definition 5. Empty attribute is the attribute that does not possess any object in formal context. In $K := (G, M, I)$, if $m \in M$ and $g(m) = \Phi$, then attribute m is an empty attribute. Correspondingly, empty object is the object that does not possess any attribute.

According to the lattice theory, the global objects and global attributes only emerge on the top and at the bottom of the concept lattice, and they will not influence the structure of the concept lattice, so the global objects and global attributes can be reduced in concept lattice [24]. Empty objects and empty attributes which have no association with any other attributes and objects are independent existence that have no influence on the process of calculating concepts.

Definition 6. In attribute topology, if any $\text{Edge}(m_i, m_j) \neq \Phi$ satisfying $\text{Edge}(m_j, m_i) \neq \Phi$, then m_j attribute is called top-attribute [21].

Edges connected to the top-attributes in attribute topology are bidirectional or unidirectional pointing to the outside.

Definition 7. In $K := (G, M, I)$, $m_i \in M$, let N be a nonempty set, for $\forall m_i \in N \subset M - \{m_i\}$ satisfying $g(m_i) \subset g(m_j)$; then m_i attribute is the child-attribute of m_j attribute and m_j attribute is the father-attribute of m_i attribute [25].

Theorem 8. In $K := (G, M, I)$, if $A \subset M$ and for $\forall a_i \in A$, satisfying $g(m) \subset g(a_i)$, that is, m attribute is the child-attribute of some attributes; m attribute may possess several father-attributes.

$F(m)$ is induced by all father-attributes of m child-attribute.

Theorem 9. A top-attribute is definitely not the child-attribute.

3. Global Formal Concepts Searching of Attribute Topology

Global formal concepts searching of attribute topology are induced in the paper in order to construct the relationship between concepts while calculating all the concepts. Limited by the constraints and calculation rules, a series of paths are formed through exploring and backtracking the vertices of attributes sorted formerly while computing all formal concepts, in which removes the relevant edges of topology simultaneously, based on the topology degenerated with the fixed start and end points.

3.1. Degeneration of Attribute Topology. According to [25], all attributes in formal context are divided into the top-attributes and child-attributes. Meanwhile, [24] puts the top-attributes as the core to decompose the topology into some subtopologies.

This paper, firstly, establishes the global attribute Ψ and the empty attribute E from the global point of view. Then, without changing the basic structure of the topology, the topology model, possessing the global start and end points, is constructed. This model, denoted by topological degeneration model, represents the degeneration of topology.

Topological degeneration model mentioned above is described as follows: vertices Ψ and E ($g(\Psi) = G, g(E) = \Phi$) are added to the original topology, with Ψ as the global start point and E as the global end point. Suppose that sets $A, B \subset M$, for $\forall m_i \in A$, generate the unidirectional edges $\langle \Psi, m_i \rangle$ and let $\text{Edge}(\Psi, m_i) = g(m_i)$; for $\forall m_j \in B$, let $\langle m_j, E \rangle = \text{End}$, where End is the terminal symbol and is represented by unidirectional edge when drawing in order to describe the model uniformly.

The establishment of the topological degeneration model can be divided into the following two cases.

TABLE 1: A formal context that contains child-attributes.

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>
0	1	0	1	1	1	1	1
1	0	0	0	0	0	1	1
2	1	1	1	0	1	1	0
3	1	1	1	0	0	1	1
4	0	1	0	1	0	1	1
5	1	1	1	1	0	0	1
6	0	1	1	1	1	0	1
7	1	0	0	0	0	1	1
8	1	1	0	1	0	1	1

FIGURE 1: The attribute topology of the formal context from Table 1.

(1) *No Existence of Child-Attributes.* Then let $A = M$ and $B = M$; that is, for $\forall m \in M$, let $\text{Edge}(\Psi, m) = g(m)$, $\langle m, E \rangle = \text{End}$.

(2) *Existence of Child-Attributes.* Let A be the set of top-attributes and B the set of child-attributes; that is, $A \cup B = M$ and $A \cap B = \Phi$.

The topological degeneration model is established by the following example: Table 1 shows a formal context that contains child-attributes. The attribute topology of the formal context listed in Table 1 is presented in Figure 1.

In the formal context (see Table 1), $\{a, b, c, g, f\}$ is the set of top-attributes and $\{d, e\}$ is the set of child-attributes. The degeneration of the topology (see Figure 1) is shown as follows: vertices Ψ and E are added to the original topology and let $A = \{a, b, c, g, f\}$, $B = \{d, e\}$, then construct the unidirectional edges from Ψ to every element in set A and unidirectional edges from every element in set B to E . The topology degenerated is presented in Figure 2.

The degeneration of the topology (see Figure 2) is presented by the newly added vertices Ψ , E , and the dashed lines with an arrow, according to the comparison of Figures 1 and 2. A complete whole of topology is achieved after the degeneration, by constructing the unidirectional edges from global start point to related vertices and unidirectional edges from related vertices to global end point.

The degeneration of the topology (see Figure 2) that contains child-attributes is actually achieved by constructing

FIGURE 2: The topological degeneration model from the topology shown in Figure 1.

the unidirectional edges from global start point to all top-attributes and unidirectional edges from all child-attributes to the global end point, while, in terms of the topology without child-attributes, its degeneration is realized by constructing the unidirectional edges from global start point to all attributes and unidirectional edges from all attributes to the global end point.

Above analysis shows that the added two vertices and edges have no influence on the original association and correlation intensity between attributes. That is, the structure of the original topology has no changes, still containing the association and correlation intensity between all attributes and objects. The association and correlation intensity required in calculating the concepts are not damaged, and subsequent calculation of concepts based on the degenerated topology has not been affected.

From the perspective of graph theory, a complete diagram with fixed start and end points, suitable for the following path searching, is achieved by the degeneration.

3.2. Algorithm Description. The traversal on path, the case of exploring and backtracking vertices, is fundamentally carried from the global start point Ψ and ends with traversing all paths between Ψ and E . The certificates for exploring and backtracking vertices are provided by the constraints and calculation rules.

3.2.1. Preliminaries

(1) *Sorted Vertices.* Suppose that M is the set of all attributes in attribute topology, the set of all top-attributes $A \subseteq M$, and the set of all child-attributes $B \subseteq M$.

For $\forall m_i \in C \subseteq M$, let $\text{num}(m_i) = \#\{n \mid \text{Edge}(n, m_i) \neq \Phi \text{ or } \text{Edge}(m_i, n) \neq \Phi, n \in M - m_i\}$. $\#\{\cdot\}$ represents the number of elements in the collection.

Definition 10. Let C be a nonempty set and $C \subseteq M$. A mapping $T : C \rightarrow C$ satisfies the following:

- (1) $C \mapsto T(C) \triangleq C^T = \{c_1, c_2, \dots, c_i \mid c_k \in C, k \in [1, i]\}$;
- (2) $\text{num}(c_1) \leq \text{num}(c_2) \leq \dots \leq \text{num}(c_i)$.

According to Definition 10, A^T , an ordered set, is the result of reordering all the elements in set A . B^T is the result of reordering elements in B by the same ruler.

Let $M_{(A,B)}^T = \{\Psi, A^T, B^T, E\}$; that is, $M_{(A,B)}^T = \{\Psi, m_1, m_2, \dots, m_i, n_1, n_2, \dots, n_j, E\}$.

According to the above analysis, $M_{(A,B)}^T$ is an ordered set after reordering the set of all attributes added to the global start and end points. The subsequent exploring and backtracking of vertices are based on $M_{(A,B)}^T$, in which the global start point is followed by a series of ordered top-attributes followed by ordered child-attributes and the global end point in the last.

In the formal context shown in Table 1, $M = \{a, b, c, d, e, f, g\}$, the set of top-attributes $A = \{a, b, c, g, f\}$, and set of child-attributes $B = \{d, e\}$. According to the mapping T , $A^T = \{a, b, c, g, f\}$, $B^T = \{d, e\}$, and $M_{(A,B)}^T = \{\Psi, a, b, c, g, f, d, e, E\}$.

(2) *Improved Description of Child-Attribute.* In order to calculate concepts, some related properties of child-attribute are introduced in this section.

Definition 11. P^a (empty or nonempty) is a set that follows the following:

- (1) a is a child-attribute;
- (2) $\#P^a \geq 1$ or $\#P^a = 0$;
- (3) $P^a = \bigcup_{i \in T} P_i^a$,

where $\forall P_i^a \in P^a$ holds the following conditions:

- (1) P_i^a is a nonset of attributes;
- (2) $\#P_i^a \geq 2$;
- (3) for $\forall m \in P_i^a$, $g(am) = O_i^a \subset g(a)$ where O_i^a is a nonempty set of objects corresponding to P_i^a ;
- (4) for $\forall n \in M - P_i^a$, then $g(an) \neq O_i^a$;
- (5) if $\#P^a > 1$, for $\forall P_j^a \in P^a (j \neq i)$, satisfying $P_j^a \neq P_i^a$ and $O_j^a \neq O_i^a$.

Theorem 12. If $\#P^a \neq 0$, for $\forall P_i^a \in P^a$, $M_T = \{m_i \mid m_i \in a \cup \{P_i^a\}\}$ must be a complete polygon.

Proof. For $\forall P_i^a \in P^a$, $\forall m \in P_i^a$, then

$$g(a) \cap g(m) = O_i^a \neq \Phi. \quad (3)$$

For $\forall m_i, m_j \in P_i^a$, then

$$g(a) \cap g(m_i) = O_i^a, \quad g(a) \cap g(m_j) = O_i^a. \quad (4)$$

Also

$$\begin{aligned} g(a) \cap g(m_i) \cap g(m_j) \\ = (g(a) \cap g(m_i)) \cap (g(a) \cap g(m_j)) \end{aligned} \quad (5)$$

combined with Formulae (4) and (5):

$$g(a) \cap g(m_i) \cap g(m_j) = O_i^a \neq \Phi. \quad (6)$$

Also

$$g(a) \cap g(m_i) \cap g(m_j) \subseteq g(m_i) \cap g(m_j). \quad (7)$$

Simultaneous Formulae (6) and (7) will get

$$g(m_i) \cap g(m_j) \neq \Phi \quad (8)$$

combined with Formulae (4) and (8):

$$g(m_i) \cap g(m_j) \neq \Phi \quad (i \in T, j \in T). \quad (9)$$

So $M_T = \{m_i \mid m_i \in a \cup \{P_i^a\}\}$ is a complete polygon. $\square$

Theorem 13. If $\#P^a \neq 0$, for $\forall P_i^a, P_j^a \in P^a$, satisfying $P_i^a \cap P_j^a = \Phi$.

Proof. According to Definition 11, for $\forall P_1^a, P_2^a \in P^a$, there exists O_1^a, O_2^a , respectively, and $O_1^a \neq O_2^a$:

$$\text{for } \forall m_i \in P_1^a, g(m_i, a) = O_1^a;$$

$$\text{similarly, for } \forall m_j \in P_2^a, g(m_j, a) = O_2^a;$$

$$\text{then for } \forall m_i \in P_1^a:$$

$$g(m_i, a) = O_1^a \neq O_2^a; \quad (10)$$

that is,

$$m_i \notin P_2^a; \quad (11)$$

$$\text{similarly, for } \forall m_j \in P_2^a, m_j \notin P_1^a;$$

$$\text{that is, } P_i^a \cap P_j^a = \Phi. \quad \square$$

Theorem 14. For $\forall P_i^a \in P^a$, satisfying $P_i^a \cap F(a) = \Phi$.

Proof. Suppose $P_i^a \cap F(a) \neq \Phi$; that is, there exists element m satisfying $m \in P_i^a$ and $m \in F(a)$:

$$\text{since } m \in F(a),$$

so

$$g(am) = g(a); \quad (12)$$

according to Definition 11,

$$g(am) = O_i^a \subset g(a); \quad (13)$$

obviously, Formulae (12) and (13) are mutually contradictory, so the assumption is not valid.

$$\text{So } P_i^a \cap F(a) = \Phi. \quad \square$$

In the formal context presented in Table 1, $F(d) = \{g\}$, $F(e) = \{c\}$ are the father-attributes of child-attributes d and e , respectively.

As Definition 11 shows $P^d = \Phi$, while e is provided with $P^e \neq \Phi$: $P_1^e = \{a, f\}$, $O_1^e = \{0, 2\}$; $P_2^e = \{g, d\}$, $O_2^e = \{0, 6\}$; and $P^e = \{\{a, f\}, \{g, d\}\}$.

3.2.2. Exploration of Vertex

(1) *Representation of the Path.* Suppose the set of all attributes in attribute topology is $X = \{x_1, x_2, \dots, x_n\}$; that is, $\#X = m$.

Definition 15. $(\Lambda^n X, \angle, \theta)$ is triple which satisfies the following properties:

$$(1) \Lambda^n X = \angle(\Lambda^n X) \cdot \theta^{(\Lambda^n X)};$$

$$(2) \angle(\Lambda^n X) = g(X);$$

$$(3) \theta^{(\Lambda^n X)} \triangleq \langle x_1, x_2, \dots, x_n \rangle,$$

where $\Lambda^n X$ satisfies the following:

$$(1) \Lambda^n X = \{x_1 \Lambda x_2, \dots, \Lambda x_n \mid \forall x_i \in X, i \in [1, n]\},$$

$$(2) \Lambda^1 X = \{x_1 \mid \forall x_1 \in X\},$$

$$(3) n \leq m,$$

$$(4) \{x_1 \Lambda x_2\} \neq \{x_2 \Lambda x_1\}, \forall x_1, x_2 \in X,$$

$$(5) \{\{x_1 \Lambda x_2\} \Lambda x_3\} = \{x_1 \Lambda \{x_2 \Lambda x_3\}\}, \forall x_1, x_2, x_3 \in X,$$

$$(6) \{x_1 \Lambda x_2 \Lambda x_3\} = \{x_1 \Lambda x_2\} \Lambda \{x_2 \Lambda x_3\}, \forall x_1, x_2, x_3 \in X.$$

As shown in Definition 15, $P = \Lambda^n X$ is uniquely determined by its magnitude and direction if $\#\{\angle(\Lambda^n X)\} \neq 0$. Its magnitude is expressed by $\angle P = \angle(\Lambda^n X)$ where $\angle(\Lambda^n X) = \bigcap_{i=1}^n g(x_i)$ and its direction is expressed by $\theta^P = \theta^{\Lambda^n X}$.

According to the above analysis, the path can be represented by $P = \Lambda^n X$: the attributes successively passed by current path, and $\langle x_1, x_2, x_3, \dots, x_n \rangle$, are recorded by θ^P . There exists unidirectional edge between each two adjacent vertices; that is, edges passed by the current path are $\langle x_1, x_2 \rangle, \langle x_2, x_3 \rangle, \dots, \langle x_{n-1}, x_n \rangle$ successively. Let $\angle P$ be the weights on the edge $\langle x_{n-1}, x_n \rangle$.

Suppose adding a new vertex x_{n+1} to the existing path; then $P = P \Lambda \{x_{n+1}\}$. Update on the path: a new vertex x_{n+1} and a new edge $\langle x_n, x_{n+1} \rangle$ are generated; that is, the attributes successively passed by the path are $\langle x_1, x_2, x_3, \dots, x_n, x_{n+1} \rangle$ and the weights on the edge $\langle x_n, x_{n+1} \rangle$ are $\angle P = \angle(\Lambda^{n+1} X)$.

Theorem 16. $X_n = \{x_1, x_2, x_3, \dots, x_n\}$ must be a complete polygon in attribute topology if $\#\{\angle(\Lambda^n X)\} \neq 0$.

Proof. Consider the following:

$$\#\left\{\angle\left(\Lambda^n X\right)\right\} \neq 0. \quad (14)$$

Then

$$\begin{aligned} \angle \left(\bigwedge^n X \right) &\neq \Phi, \\ \angle \left(\bigwedge^n X \right) &= \bigcap_{i=1}^n g(x_i) \neq \Phi \end{aligned} \quad (15)$$

for $\forall x_k, x_j \in X$

$$\bigcap_{i=1}^n g(x_i) \subseteq g(x_k) \cap g(x_j) \quad (16)$$

combined with Formulae (15) and (16), for $\forall x_k, x_j \in X$

$$g(x_k) \cap g(x_j) \neq \Phi. \quad (17)$$

So $X_n = \{x_1, x_2, x_3, \dots, x_n\}$ must be a complete polygon. $\square$

Lemma 17. $\{x_1, x_2, x_3, \dots, x_n\}$ not be the intension certainly if $\#\{\angle(\bigwedge^n X) \mid n < m\} = 0$.

Proof. Consider the following:

$$\#\left\{\angle \left(\bigwedge^n X \right) \mid n < m\right\} = 0. \quad (18)$$

Then

$$\angle \left(\bigwedge^n X \right) = \Phi \quad (19)$$

so

$$\bigcap_{i=1}^n g(x_i) = g\left(\bigcup_{i=1}^n x_i\right) = \Phi. \quad (20)$$

To prove that $\{x_1, x_2, x_3, \dots, x_n\}$ is the intension of a concept, satisfying Formulae (1) is needed; that is,

$$f\left(g\left(\bigcup_{i=1}^n x_i\right)\right) = \left\{\bigcup_{i=1}^n x_i\right\} \quad (21)$$

since

$$f\left(g\left(\bigcup_{i=1}^n x_i\right)\right) = f(\Phi) \quad (22)$$

in this formal context,

$$f(\Phi) = \{x_1, x_2, x_3, \dots, x_m\} \quad (23)$$

combined with Formulae (22) and (23):

$$f\left(g\left(\bigcup_{i=1}^n x_i\right)\right) = \{x_1, x_2, x_3, \dots, x_m\}. \quad (24)$$

Obviously, Formulae (21) and (24) are mutually contradictory; that is, $\{x_1, x_2, x_3, \dots, x_n\}$ is not the intension. $\square$

Lemma 18. $X' = \{x_1, x_2, x_3, \dots, x_{n-1}\}$ must be the intension of a concept if $\#\{\angle(\bigwedge^{n-1} X)\} \neq 0$ and $\#_{\forall x_s \in X - \{x_1, x_2, \dots, x_{n-1}\}} \{\angle(\bigwedge^{n-1} X) \wedge x_s\} = 0$.

Proof. Consider the following:

$$\#\left\{\bigwedge_{i=1}^{n-1} x_i \mid x_i \in X\right\} \neq 0 \quad (25)$$

and then

$$\bigcap_{i=1}^n g(x_i) = g(X') \neq \Phi \quad (26)$$

since $\#_{\forall x_s \in X - \{x_1, x_2, \dots, x_{n-1}\}} \{\angle(\bigwedge^{n-1} X) \wedge x_s\} = 0$

so for $\forall x_s \in X - X'$,

$$g(X') \cap g(x_s) = \Phi. \quad (27)$$

According to Formulae (1), to prove the intension of a formal concept, it is only needed to satisfy $f(g(X')) = X'$. From the relationship of formal context,

$$X' \subseteq f(g(X')) \quad (28)$$

if $X' \subset f(g(X'))$; then there must be

$$f(g(X')) = X' \cup N, \quad (29)$$

where $N \subseteq X - X'$ and $N \neq \Phi$, and then

$$g(f(g(X'))) = g(X' \cup N) = g(X') \cap g(N). \quad (30)$$

Also

$$g(f(g(X'))) = g(X'). \quad (31)$$

Simultaneous Formulae (30) and (31) will get

$$g(X') = g(X') \cap g(N). \quad (32)$$

Combined with Formulae (27) and (32):

$$g(X') = \Phi \quad (33)$$

apparently the result is wrong.

So

$$\begin{aligned} X' &\not\subseteq f(g(X')), \\ f(g(X')) &= X'. \end{aligned} \quad (34)$$

So X' is the intension of a formal concept. $\square$

(2) *Process of Depth Exploration of Vertex.* Depth exploration of vertices in topology is essentially the process of exploring the adjacent points through the arcs between vertices. Suppose the set of all attributes of formal context is X ($\#X = n - 2$), and the sets A and B are the sets of top-attributes and child-attributes, respectively. According to Section 3.2.1(1),

FIGURE 3: Flowchart of the process of exploring attribute m and the constrains related.

the set of vertices sorted is $X_{(A,B)}^T = \{x_1, x_2, \dots, x_n\}$, based on which to explore vertices.

This algorithm begins with the first element, that is, start point Ψ , and then explores the subsequent elements successively. For the sorted set $X_{(A,B)}^T$, $x_2 = \text{Next}(\Psi)$, $x_3 = \text{Next}(x_2)$.

Suppose $P = \Lambda^{k-1} X_{(A,B)}^T$; $k \leq n$ is the current path and ordered set $I = \{x_1, x_2, \dots, x_{k-1}\}$ is the set of attributes which passed by the path successively. Suppose that attribute m is

the current attribute explored, and the process of exploring m and the constrains of exploring are plotted in Figure 3.

As seen in Figure 3, when exploring the attribute m , constrains are shown as follows:

- (1) $\forall x_i \in I - \{\Psi\}$, $\text{Edge}(m, x_i) \neq \emptyset$, or $\text{Edge}(x_i, m) \neq \emptyset$;
- (2) m is a child-attribute;
- (3) $F(m) \subseteq I$;
- (4) $P^m = \emptyset$;

FIGURE 4: The updated path from the path $P = \{\Psi\}$.

- (5) for $\forall P_i^m \in P^m$, satisfying $P_i^m \subseteq I$, or $I \cap P_i^m = \Phi$;
- (6) $g(I) \cap g(m) \neq \Phi$.

Figure 3 displays that the current path is updated, that is, $P = P \Delta \{m\}$ and $I = I \cup \{m\}$, when it meets any one of the following conditions, that is, the constrains of exploring:

- (a) $(1) \cap (2) \cap (6) = \text{true}$;
- (b) $(1) \cap (2) \cap (3) \cap (4) \cap (6) = \text{true}$;
- (c) $(1) \cap (2) \cap (3) \cap (4) \cap (5) \cap (6) = \text{true}$.

It is needed to traverse the attribute next to attribute $m(v = m, m = \text{Next}(v))$, when it meets the condition $\bar{a} \cap \bar{b} \cap \bar{c} = \text{true}$, that is, not satisfying the constrains of exploring.

Then we analyze the exploring process through an example. For the formal context shown in Table 1, $X_{(A,B)}^T = \{\Psi, a, b, c, g, f, d, e, E\}$.

This algorithm begins with the start point, that is, $P = \{\Psi\}$, $I = \{\Psi\}$; current attribute explored $m = a$. Attribute a satisfies the following conditions: a is not the child-attribute; $g(I) \cap g(a) \neq \Phi$; that is, attribute a meets the above condition (a). The current path is updated: $P = \{\Psi \Delta a\}$, $I = \{\Psi, a\}$; the weights on the newly added edge $\langle \Psi, a \rangle$ are $w \langle \Psi, a \rangle = \angle P = g(\Psi) \cap g(a) = g(a) = \{0, 2, 3, 5, 7, 8\}$; and the direction of the path updated is $\theta^P = \langle \Psi, a \rangle$, that is, from Ψ to a .

The process above can be demonstrated by Figure 4.

Figure 4 displays that the path $= \{\Psi \Delta a\}$: θ^P is presented as the arrow shown; $\angle P$, as the weights, is marked on the newly generated edge.

Additionally, suppose $P = \{\Psi \Delta a \Delta b \Delta c\}$; $I = \{\Psi, a, b, c\}$ is the current path, and current attribute explored $m = e$. Attribute e satisfies the following conditions:

- (1) $\forall x_i \in I - \{\Psi\}$, $\text{Edge}(x_i, e) \neq \Phi$;
- (2) e is a child-attribute;
- (3) $F(e) \subseteq I(F(e) = \{c\})$;
- (4) $P^e = \Phi(P_1^e = \{a, f\}, P_2^e = \{d, g\})$;
- (5) $P_1^e \not\subseteq I, I \cap P_2^e = \Phi$.

That is, e satisfies the above condition $\bar{a} \cap \bar{b} \cap \bar{c} = \text{true}$; then it explores the attribute E next to attribute $e(\text{Next}(e) = E)$: when the current attribute explored m is updated from e to E , the current path does not update.

(3) *Updated Data in Process of Depth Exploration of Vertex.* From the previous section analysis, the path is updated when the explored attribute m meets the constrains of exploring and adds to the current path. There are a series of data to be updated, mainly in the following two aspects.

(1) *Updated Set of Concepts C.* Suppose the set of all attributes of formal context is X ($\#X = n - 2$), $X_{(A,B)}^T = \{x_1, x_2, \dots, x_n\}$,

FIGURE 5: The path updated from the path listed in Figure 4.

$\#X_{(A,B)}^T = n$. Let the set of attribute category be $\{0, 1\}$ and the category of attribute be $\text{Mark}(x_i)$, that is, $\text{Mark}(x_i) \in \{0, 1\}$. For $\forall x_i \in X_{(A,B)}^T$, $\text{Mark}(x_i)$ is initialized to 0. The set of concepts C is initialized to Φ .

Suppose the current path is $P = \Delta^k X_{(A,B)}^T, k \leq n, I = \{x_1, x_2, \dots, x_k\}$, and the current set of concepts is $C = \{c_1, c_2, \dots, c_l\}$, and for $\forall c_j \in C, c_j = (A_j, B_j)$, that is, $C = \{(A_1, B_1), (A_2, B_2), \dots, (A_l, B_l)\}$.

Suppose the current attribute explored m satisfies the constrains of exploring shown in previous section and the path is updated, that is, $P' = P \Delta \{m\}$, $\angle P' = \angle P \cap g(m)$, $I' = I \cup \{m\}$.

Simultaneously, the set C is updated.

If $\angle P' = \angle P$, let the newly generated two-tuples $(\angle P', I') \rightarrow (A_s, B_s) : A_s = \angle P, s \leq l$. That is, the set C is updated to $C = \{(A_1, B_1), (A_2, B_2), \dots, (A_{l-1}, B_{l-1}), (\angle P', I')\}$, $s = l$ or $C = \{(A_1, B_1), (A_2, B_2), \dots, (A_{s-1}, B_{s-1}), (\angle P', I'), \dots, (A_s, B_s)\}$, $s < l$; $\text{Mark}(x_k) = 1$.

If $\angle P' \neq \angle P$, $C = C \cup \{(\angle P', I')\}$, that is, $C = \{(A_1, B_1), (A_2, B_2), \dots, (A_l, B_l), (\angle P', I')\}$.

(2) *Updated Attribute Topology.* If $\angle P' = \text{Edge}(x_k, m)$, then let $\text{Edge}(x_k, m) = \text{Edge}(m, x_k) = \Phi$; that is, the edge (unidirectional or bidirectional), between x_k and m , is removed from the attribute topology.

If $\angle P' \neq \text{Edge}(x_k, m)$, the attribute remains unchanged.

On the contrary, updates on the two aspects described above will not be induced if the current attribute explored m does not meet the constrains of exploring shown in previous section.

Then we analyze the date updated through an example.

For the formal context shown in Table 1, suppose the current path is $P = \{\Psi \Delta a\}$ and $I = \{\Psi, a\}$, $C = \{(\{0, 1, 2, 3, 4, 5, 6, 7, 8\}, \{\Psi\}), (\{0, 2, 3, 5, 7, 8\}, \{\Psi, a\})\}$, which is presented in Figure 4. Suppose the current attribute explores $m = b$. Attribute b meets the constrains of exploring shown in previous section and path is updated: $P' = \{\Psi \Delta a \Delta b\}$, $\angle P' = g(\Psi a) \cap g(b) = \{0, 2, 3, 5, 7, 8\} \cap \{2, 3, 4, 5, 6, 8\} = \{2, 3, 5, 8\}$, and $I' = \{\Psi, a, b\}$. The updated path is demonstrated by Figure 5, and for convenience of description, the newly generated edge is presented by the route without an arrow.

The updated data is shown as follows.

Updated Set C. $\angle P' = \{2, 3, 5, 8\} \neq \angle P = \{0, 2, 3, 5, 7, 8\}$ and then adds the newly generated two-tuples $(\{2, 3, 5, 8\}, \{\Psi, a, b\})$ to the set C , that is, $C = \{(\{0, 1, 2, 3, 4, 5, 6, 7, 8\}, \{\Psi\}), (\{0, 2, 3, 5, 7, 8\}, \{\Psi, a\}), (\{2, 3, 5, 8\}, \{\Psi, a, b\})\}$. The update on the set of concepts C , along with the update on path, is shown in Table 2.

TABLE 2: The update on set C , along with the update on the path shown in Figure 5.

P	C
$\{\Psi\Lambda a\}$	$(\{0, 1, 2, 3, 4, 5, 6, 7, 8\}, \{\Psi\})$ $(\{0, 2, 3, 5, 7, 8\}, \{\Psi, a\})$
$\{\Psi\Lambda a\Lambda b\}$	$(\{0, 1, 2, 3, 4, 5, 6, 7, 8\}, \{\Psi\})$ $(\{0, 2, 3, 5, 7, 8\}, \{\Psi, a\})$ $(\{2, 3, 5, 8\}, \{\Psi, a, b\})$

$\Psi \xrightarrow{\{023578\}} a \xrightarrow{\{0235\}} c \xrightarrow{\{035\}} g \xrightarrow{\{03\}} f \xrightarrow{\{0\}} d$

FIGURE 6: The current path $P = \{\Psi\Lambda a\Lambda c\Lambda g\Lambda f\Lambda d\}$.

$\Psi \xrightarrow{\{023578\}} a \xrightarrow{\{0235\}} c \xrightarrow{\{035\}} g \xrightarrow{\{03\}} f \xrightarrow{\{0\}} d \xrightarrow{\{0\}} e$

FIGURE 7: The path updated from the path listed in Figure 6.

Updated Attribute Topology. Suppose $\angle P' = \{2, 3, 5, 8\}$. Figure 2 presents that $\text{Edge}(a, b) = \{2, 3, 5, 8\}$ which satisfies the condition $\angle P' = \text{Edge}(x_k, m)$, so let $\text{Edge}(a, b) = \text{Edge}(b, a) = \Phi$; that is, the bidirectional edge, between a and b , is removed from the attribute topology.

Additionally, Suppose $P = \{\Psi\Lambda a\Lambda c\Lambda g\Lambda f\Lambda d\}$, listed in Figure 6, is the current path and $C = \{(\{0, 1, 2, 3, 4, 5, 6, 7, 8\}, \{\Psi\}), \dots, (\{0\}, \{\Psi, a, c, g, f, d\})\}$, $I = \{\Psi, a, c, g, f, d\}$.

Suppose the current attribute explored $m = e$. Attribute e satisfies the constrains of exploring and then the path is updated: $P = \{\Psi\Lambda a\Lambda c\Lambda g\Lambda f\Lambda d\Lambda e\}$, $\angle P' = \angle P \cap g(b) = \{0, 3\} \cap \{0, 4, 5, 6, 8\} = \{0\}$. The updated path is presented in Figure 7.

The updated data is shown as the following.

Updated Set C. Suppose $\angle P' = \{0\} = \angle P = \{0\}$. Then the element $(\{0\}, \{\Psi, a, c, g, f, d, e\})$ in set C is replaced with the newly generated two-tuples $(\{0\}, \{\Psi, a, c, g, f, d, e\})$; that is, the number of elements in set C remains unchanged: $\text{Mark}(d) = 1$.

Table 3 lists each update on set C , along with each update on path, from path $P = \{\Psi\}$ to the current path $P = \{\Psi\Lambda a\Lambda c\Lambda g\Lambda f\Lambda d\Lambda e\}$.

Updated Attribute Topology. Suppose $\angle P' = \{0\}$. Figure 2 presents that $\text{Edge}(d, e) = \{0, 6\}$ which satisfies the condition $\angle P' \neq \text{Edge}(x_k, m)$, so the attribute topology remains unchanged.

3.2.3. Backtracking of Vertex. Suppose the set of all attributes of formal context is X ($\#X = n - 2$), $X_{(A,B)}^T = \{x_1, x_2, \dots, x_n\}$, $\#X_{(A,B)}^T = n$ and the current path is $P = \Lambda^k X_{(A,B)}^T$, $k \leq n$, $I = \{x_1, x_2, \dots, x_k\}$.

FIGURE 8: Flowchart of the process of backtracking vertices.

TABLE 3: The update on set C , along with the update on the path shown in Figure 7.

P	C
$\{\Psi\}$	$(\{0, 1, 2, 3, 4, 5, 6, 7, 8\}, \{\Psi\})$
$\{\Psi\Lambda a\}$	$(\{0, 1, 2, 3, 4, 5, 6, 7, 8\}, \{\Psi\})$ $(\{0, 2, 3, 5, 7, 8\}, \{\Psi, a\})$
$\vdots$	$\vdots$
	$(\{0, 1, 2, 3, 4, 5, 6, 7, 8\}, \{\Psi\})$
$\{\Psi\Lambda a\Lambda c\Lambda g\Lambda f\Lambda d\}$	$\vdots$ $(\{0, 3\}, \{\Psi, a, c, g, f\})$ $(\{0\}, \{\Psi, a, c, g, f, d\})$
	$(\{0, 1, 2, 3, 4, 5, 6, 7, 8\}, \{\Psi\})$
$\{\Psi\Lambda a\Lambda c\Lambda g\Lambda f\Lambda d\Lambda e\}$	$\vdots$ $(\{0, 3\}, \{\Psi, a, c, g, f\})$ $(\{0\}, \{\Psi, a, c, g, f, d, e\})$

Backtracking the vertex occurs if the current attribute explored m satisfies any of the following two conditions, that is, constrains of backtracking:

- (1) $m = E$ & $P \neq \{\Psi\}$;
- (2) $\text{Mark}(m) = 1$.

Suppose the current path is $P = \Lambda^k X_{(A,B)}^T$, $k \leq n$, and $I = \{x_1, x_2, \dots, x_k\}$. When the current attribute explored m satisfies the constrains of backtracking, backtracking the vertex: $P' = \Lambda^{k-1} X_{(A,B)}^T$, that is, $P' = \{x_1 \Lambda x_2 \Lambda \dots \Lambda x_{k-1}\}$, $\angle P' = \bigcap_{i=1}^{k-1} g(x_i)$, $I' = \{x_1, x_2, \dots, x_{k-1}\}$, $\text{Mark}(x_k) = 0$, and set C remains unchanged.

$$\Psi \xrightarrow{\{023578\}} a \xrightarrow{\{0235\}} c \xrightarrow{\{035\}} g \xrightarrow{\{03\}} f$$

FIGURE 9: The path updated from the path listed in Figure 7.

Combined with the description above, the process of backtracking vertices is listed in Figure 8. $\text{Last}(\cdot)$ represents the last element of the order set.

The process of backtracking vertices is finished when it meets the condition $\overline{(1)} \cap \overline{(2)} = \text{true}$, according to Figure 8.

Then, the process of backtracking vertices is illustrated by the example.

Suppose the current path is shown in Figure 7. According to Section 3.2.2, $\text{Mark}(\bigcup_{i=1}^7 x_i) = \{0, 0, 0, 0, 0, 1, 0\}$, $\text{Mark}(d) = 1$.

Suppose the current attribute explored $m = E$, which satisfies the constraints of backtracking, then it backtracks the vertex, that is, $P = \{\Psi \wedge a \wedge c \wedge g \wedge f \wedge d\}$, $I = \{\Psi, a, c, g, f, d\}$, $\text{Mark}(e) = 0$.

Then a judge on $\text{Mark}(\text{Last}(I))$ is achieved, presented in Figure 9. Due to the fact that $\text{Mark}(\text{Last}(I)) = \text{Mark}(d) = 1$ satisfies the constraints of backtracking, backtracking of vertex is achieved again, that is, $P = \{\Psi \wedge a \wedge c \wedge g \wedge f\}$, $I = \{\Psi, a, c, g, f\}$, $\text{Mark}(d) = 0$ and $v = d$.

Repeating the judge on $\text{Mark}(\text{Last}(I))$, and $\text{Mark}(\text{Last}(I)) = \text{Mark}(f) = 0$, which leads to the completion of the backtracking vertices. Then the current attribute explored m is updated to $\text{Next}(v) = \text{Next}(d) = e$ and the set C remains unchanged. The updated path is shown in Figure 9.

3.2.4. Discussion. Flowchart of the complete algorithm is listed in Figure 10.

As seen in Figure 10, the process of exploring m is achieved when the backtracking process is finished. In the process of depth exploration, the process of backtracking m is achieved if it satisfies the constraints of backtracking. That is, all paths between the two fixed points which meet related constraints are achieved in the process of repeating, exploring, and backtracking vertices. The update on the set of concepts C and attribute topology are induced in the process of traversing the path.

Figure 10 presents that the algorithm ends when it meets the condition $(m = E) \cap (P = \{\Psi\}) = \text{true}$.

The update on set C and attribute topology, after the end of the algorithm, are demonstrated, respectively, by the example presented above:

- (1) updated topology is shown in Figure 11;
- (2) updated set C .

Each two-tuples $(\{\Psi\} \cup A, B)$ is updated to (A, B) due to the fact that the start point Ψ is not of practical significance. Then every element in set C is a concept; that is, $\beta(G, M, I) = C \cup \{(\Phi, abcdefg)\}$ in formal context $K := (G, M, I)$.

The traversal process of all paths between the start and end points is illustrated by the tree structure shown in Figure 12.

Figure 12 displays that each path begins with Ψ and ends with a leaf node. Each path is generated from top to bottom and all paths are formed from left to right.

Any node n in Figure 12 can be seen as a concept, with the set of all attributes passed by the current path in which current node n located, from Ψ to node n , as the intension and the weights on the edge, whose lower end is connected to the current node n , as the extension. The node $e(d)$ presents that attribute d is also passed by the current path where node n located.

As seen in Figure 13, the concepts represented by nodes in Figure 12 are indicated on the corresponding nodes and generated a concept tree. Figure 13 displays all concepts except the global concept $(\Phi, abcdefg)$.

Figure 14 shows the Hasse diagram of the formal context plotted in Table 1 and each node represents a formal concept.

The concepts successively represented by $c(0)$ – $c(47)$ are $(012345678, \emptyset)$, $(01345678, g)$, $(0123478, f)$, $(013478, gf)$, $(04568, gd)$, $(048, gfd)$, $(02356, c)$, $(0356, cg)$, $(026, ce)$, $(056, cgd)$, $(06, cgde)$, $(234568, b)$, $(34568, bg)$, $(2348, bf)$, $(348, bgf)$, $(4568, bgd)$, $(48, bdfg)$, $(2356, bc)$, $(356, bcbg)$, $(6, bce)$, $(56, bcgd)$, $(6, bcgde)$, $(023578, a)$, $(03578, ag)$, $(02378, af)$, $(0378, agf)$, $(058, agd)$, $(08, agfd)$, $(0235, ac)$, $(035, acg)$, $(023, acf)$, $(03, acgf)$, $(02, acfe)$, $(05, acgd)$, $(0, acgfde)$, $(2358, ab)$, $(258, abg)$, $(238, abf)$, $(38, abgf)$, $(58, abgd)$, $(8, abgfd)$, $(235, abc)$, $(35, abcg)$, $(23, abcf)$, $(3, abcgf)$, $(2, abcgfe)$, $(5, abcgd)$, and $(\emptyset, abcdefg)$, that is, all concepts in the formal context.

Under the constraints and calculation rules, the process of computing all formal concepts is achieved by traversing vertices successively. The path is formed by the traversed vertices and all concepts are induced in the process of traversing all paths. As shown in Figures 12 and 13, the process of traversing all paths is represented by the concept tree, demonstrating the calculation of all the concepts intuitively and vividly.

As seen in Figure 13, all concepts are achieved completely and the calculation process is presented intuitively, according to the global formal concepts searching of attribute topology. Comparing Figures 13 and 14, as the part of Hasse diagram, the concept tree not only demonstrates the hierarchical structures among concepts clearly but is also much simpler than the Hasse diagram.

4. Conclusion

Global formal concepts searching of attribute topology is proposed in this paper based on the concepts of calculating with subtopologies. With the basic idea of Depth First Search, this algorithm, beginning with the global start point, employs the constraints and calculation rules to explore and backtrack the attributes of the degenerated topology repeatedly until traversal of all paths is achieved. The set of concepts is

FIGURE 10: Flowchart of the complete algorithm.

FIGURE 11: The updated attribute topology after the end of the algorithm.

updated throughout the process and all concepts are obtained ultimately. This method avoids the decomposition process of the whole topology, which reflects the integrity of the algorithm. Visualization features of the calculation process

are enhanced by the concept tree. This method makes the whole process more logical and feasible, easy to implement, and suitable for large-scale data sets. Attribute topology provides a new approach of representation of the formal

FIGURE 12: The path traversal tree of the attribute topology presented in Figure 2.

FIGURE 13: The concept tree of the attribute topology presented in Figure 2.

FIGURE 14: Hasse diagram of the formal context presented in Table 1.

context. The further step is to refine and optimize the method and put it into application.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China (nos. 61273019 and 81373767) and National Social Science Foundation of China (no. 12BYY121). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

References

- [1] B. Ganter and R. Wille, *Formal Concept Analysis: Mathematical Foundations*, Springer, Berlin, Germany, 1999.
- [2] M. Kaytoue, S. O. Kuznetsov, A. Napoli, and S. Duplessis, "Mining gene expression data with pattern structures in formal concept analysis," *Information Sciences*, vol. 181, no. 10, pp. 1989–2001, 2011.
- [3] T. Abudawood, "Improving predictions of multiple binary models in ILP," *The Scientific World Journal*, vol. 2014, Article ID 739062, 10 pages, 2014.
- [4] Y. Du and Y. Hai, "Semantic ranking of web pages based on formal concept analysis," *Journal of Systems and Software*, vol. 86, no. 1, pp. 187–197, 2013.
- [5] A. Formica, "Semantic Web search based on rough sets and fuzzy formal concept analysis," *Knowledge-Based Systems*, vol. 26, pp. 40–47, 2012.
- [6] B. Li, X. Sun, and H. Leung, "Combining concept lattice with call graph for impact analysis," *Advances in Engineering Software*, vol. 53, pp. 1–13, 2012.
- [7] B. Li, X. Sun, and J. Keung, "FCA-CIA: an approach of using FCA to support cross-level change impact analysis for object oriented Java programs," *Information and Software Technology*, vol. 55, no. 8, pp. 1437–1449, 2013.
- [8] L. Seremeti and K. Ioannis, "Yoneda philosophy in engineering," *International Journal of Engineering Mathematics*, vol. 2013, Article ID 758729, 11 pages, 2013.
- [9] X. Kang, D. Li, and S. Wang, "Research on domain ontology in different granulations based on concept lattice," *Knowledge-Based Systems*, vol. 27, pp. 152–161, 2012.
- [10] J. Outrata and V. Vychodil, "Fast algorithm for computing fixpoints of Galois connections induced by object-attribute relational data," *Information Sciences*, vol. 185, pp. 114–127, 2012.
- [11] F. Škopljanač-Maćina and B. Blašković, "Formal concept analysis—overview and applications," *Procedia Engineering: Proceedings of the 24th DAAAM International Symposium on Intelligent Manufacturing and Automation(ISIMA '13)*, vol. 69, pp. 1258–1267, 2014.
- [12] J. Pang, X. Zhang, and W. Xu, "Attribute reduction in intuitionistic fuzzy concept lattices," *Abstract and Applied Analysis*, vol. 2013, Article ID 271398, 12 pages, 2013.
- [13] C. Zhang, J. Ruan, H. Huang, L. Li, and B. Yang, "An algorithm on generating lattice based on layered concept lattice," *Telkomnika—Indonesian Journal of Electrical Engineering*, vol. 11, no. 8, pp. 4477–4483, 2013.
- [14] C. Zhang, J. Ruan, H. Huang, L. Li, and B. Yang, "Frequent itemsets mining based on concept lattice and sliding window," *Telkomnika—Indonesian Journal of Electrical Engineering*, vol. 11, no. 8, pp. 4780–4787, 2013.
- [15] C. Zhang, J. Ruan, H. Xia, H. Huang, and B. Yang, "A frequent pattern mining algorithm based on concept lattice," *Sensors & Transducers*, vol. 156, no. 9, pp. 351–359, 2013.
- [16] H. Hidenobu, T. Takanari, H. Fumiaki, H. Nur, S. Kazuhito, and N. Hajime, "An algorithm for recomputing concepts in microarray data analysis by biological lattice," *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol. 17, no. 5, pp. 761–771, 2013.
- [17] Y. Yang, "Parallel construction of variable precision concept lattice in fuzzy formal context," *AASRI Procedia: 2013 AASRI Conference on Parallel and Distributed Computing and Systems*, vol. 5, pp. 214–219, 2013.
- [18] L. Yang, Y. Wang, and Y. Xu, "A combination algorithm of multiple lattice-valued concept lattices," *International Journal of Computational Intelligence Systems*, vol. 6, no. 5, pp. 881–892, 2013.
- [19] L. Zhang, H. Zhang, L. Yin, and D. Han, "Theory and algorithms of attribute decrement for concept lattice," *Computer Research and Development*, vol. 50, no. 2, pp. 248–259, 2013.
- [20] L. Zhang, H. Zhang, X. Shen, and L. Yin, "An incremental algorithm for removing object from concept lattice," *Journal of Computational Information Systems*, vol. 9, no. 9, pp. 3363–3372, 2013.
- [21] X. Gong, "A novel incorporate algorithm of concept lattice," *Advanced Materials Research*, vol. 756, pp. 2803–2807, 2013.
- [22] J. Muangprathub, V. Boonjing, and P. Pattaraintakorn, "A new case-based classification using incremental concept lattice knowledge," *Data and Knowledge Engineering*, vol. 83, pp. 39–53, 2013.
- [23] T. Zhang and H. Ren, "Attribute topology of formal context," *Journal of Chinese Computer Systems*, vol. 35, no. 3, pp. 590–593, 2014.
- [24] T. Zhang, H. Ren, X. Wang, Y. Zhang, and W. Hong, "A calculation of formal concept by attribute topology," *ICIC Express Letters B: Applications*, vol. 4, no. 3, pp. 793–800, 2013.
- [25] T. Zhang, H. Ren, W. Hong, and H. Li, "The visual calculation of formal concept based on the attribute topology," *Acta Electronica Sinica*, vol. 42, no. 5, pp. 918–924, 2014.