Journal of Network Intelligence
Taiwan Ubiquitous Information
 ISSN 2414-8105 (Online)
Volume **6**, Number **1**, February **2021**

Vertical Union Algorithm of Concept Lattice Based on Grade

[1]Haixia Li*, [2]Linlin Tang and [3]Jyh-haw Yeh

[1]Department of General Education, Anhui Xinhua University, Hefei 230088, China
[2]Harbin Institute of Technology, Shenzhen, Shenzhen 518055, China
[3]Department of Computer Science, Boise State University, Idaho, USA
Learain1@126.com*

Received September 2020; revised November 2020

Abstract. *The union of concept lattice is an important research direction in the formal concept analysis, where one vertical union method of concept lattice is given. To improve the efficiency of union, A vertical union algorithm is introduced in the paper. It only needs to compare the intent intersection of two nodes in two sublattices with the intent of the just generated node or the intents of some nodes in some grades from top to bottom. And there are not only fewer comparisons, but also no redundant nodes, which will reduce the complexity of the algorithm. At the same time, Hasse diagram of concept lattice is generated.*
Keywords: Concept Lattice, Node, Vertical Uion, Intent, Grade

1. **Introduction.** Formal concept analysis(FCA) put forward by Wille in 1982[1], according to the partial order relation, formal context can generate a concept lattice. Nowadays, concept lattice has been widely used in various research fields, such as data mining[2-5], information retrieval[6], medical research[7-9], mining engineering and so on[10,11].

As a prerequisite for the application of concept lattice, the construction of concept lattice is a very important subject, which is divided into two categories: batch algorithm and incremental algorithm. With the development of network technology and the objective demand of distributed data storage and processing, union algorithm of multiple concept lattices also arises. In this paper, one vertical union algorithm of concept lattices is discussed. During the union, all the nodes of the subconcept lattices are arranged in ascending order of intent, the concept lattice can be generated from the top to bottom. There is not only fewer comparisons, but also no redundant nodes, which will reduce complexity of the algorithm and increase efficiency. At the same time, Hasse diagram of concept lattice is generated.

2. **Preliminaries. Definition 1**[12] One formal context $K = (O, D, R)$ is a triple, O is an object set, D is an attribute set, $R \subseteq O \times D$ is binary relation between O and D. For $A \subseteq O$, $B \subseteq D$,the mapping is defined as

$$A' = \{m \in D | \forall g \in A, (g, m) \in R\} \tag{1}$$

$$B' = \{g \in O | \forall m \in D, (g, m) \in R\} \tag{2}$$

If $A = B'$, $B = A'$, (A, B) is named as one node, and A is the extent of the node (denoted as extent(C)) and B the intent(denoted as intent(C)), respectively.

Definition 2[12] If and are nodes $C_1 = (A_1, B_1)$ and $C_2 = (A_2, B_2)$ are nodes and $A_1 \supseteq A_2$ ($\Leftrightarrow B_1 \subseteq B_2$), C_2 is called the child node of C_1, C_1 is the parent node of C_2, which is denoted as $C_1 \geq C_2$. If there is no node C_3 which satisfies $C_1 \geq C_3 \geq C_2$. C_2 is called the direct child node of C_1 and C_1 the direct parent node of C_2. In this order, the set of all nodes is called the concept lattice of K and denoted as $L(O, D, R)$. The greatest node is (O, O'), the smallest node is (D', D).

Definition 3[12] $K_1 \pm K_2 = (O_1 \cup O_2, D, R_1 \cup R_2)$ is said to be vertical union between the formal contexts K_1 and K_2, if both $K_1 = (O_1, D, R_1)$ and $K_2 = (O_2, D, R_2)$ are the formal contexts with the same attributes set.

3. The Vertical Union Algorithm.

3.1. Definition and Theorem.

Vertical union is to construct concept lattice from formal context corresponding, then insert nodes of a sublattice in another sublattice to generate a new lattice. Then insert the nodes of another sublattice in this new lattice, and so on to generate the lattice.

There are two problems to be solved when inserting a node into a concept lattice: new nodes generation and edges update. For this, the following definitions and theorems should be firstly established.

Definition 4 For one node $C = (A, B)$, the node C_1 is called renewed node if C_1 satisfies $B_1 \subseteq B$.

Definition 5 For one node $C = (A, B)$, the node C_1 in one concept lattice is called generator node if C_1 satisfies the following condition:(1) there is no any node C_2 in the concept lattice, intent$(C_2)?B \cap B_1$. (2) intent $(C_3) \cap B \neq B \cap B_1$, for any parent node C_3 of C_1.

Theorem 1 If the node $C_1 = (A_1, B_1)$ in the concept lattice $L(K_1)$ is generator node of the node $C = (A, B)$ in the concept lattice $L(K_2)$, then the new produced node is $(A \cup A_1, B \cap B_1)$.

Proof There is no node in the original concept lattice, which intent is $B \cap B_1$, therefore, one new node will be generated in the new concept lattice. Let $B_2 = B \cap B_1$, then $f(g(B_2))$.

We proceed to prove $g(B_2 = A \cup A_1)$. If $A \cup A_1 \subset g(B_2)$, then $A_1 \cup A_x = g(B_2)$, $A \cup A'_x = g(B_2)$, $A_1 \cup A_x \in O_1$, $A_2 \cup A_{x'} \in O_2$. Simultaneously,$f(g(B_2)) = B_2$, we would conclude that $f(A_1 \cup A_x) = f(g(B_2)) = B_2$. That is, $(A_1 \cup A_x, B_2)$ is one node of $L(K_1)$ and one parent node of the node (A, B), in contradiction with the definition 5, hence $A_x = \phi$; similarly $A'_x = \phi$.

Hence $g(B_2) = A_1$ in the concept lattice $L(K_1)$, and $g(B_2) = A_1$ in the concept lattice $L(K_2)$, the intent (B_2) is $A \cup A_1$ in the new concept lattice $L(K_1 \pm K_2)$. This prove the theorem.

From the definition 4 and theorem 1, it is obvious that the theorem 2 holds:

Theorem 2 If node C_1 in the concept lattice $L(K_1)$ and node C_2 in the concept lattice $L(K_2)$ can produce one new node C, then node $C = (extent(C_1) \cup extent(C_2)$, intent $((C_1) \cap intent(C_2))$.

Theorem 3 The nodes in concept lattices $L(K_1)$ and $L(K_2)$ are arranged in ascending order of the intents, if the same node in $L(K_1)$ is merged with two different nodes in $L(K_2)$ and two nodes are generated followed, the latter must be the child node of the previous one, and which must be the direct child node of the node generated just before.

Proof Let the two nodes $C_1 = (A_1, B_1)$ and $C_2 = (A_2, B_2)$ are generated by the node D_1 and E_1, D_1 and E_2 respectively, where D_1 is the node of the concept lattices $L(K_1)$, E_1 and E_2 are the nodes in $L(K_2)$, so $B_1 = intent(D_1) \cap intent(E_1)$, $B_2 = intent(D_1) \cap intent(E_2)$.

Because C_2 is generated after C_1, and the nodes are arranged in ascending order of the intents, then $B_1 \subset B_2$. By the definition 2, C_2 is the child node of C_1.

Assuming that C_2 is newly generated after the next node C_1, it has been proved that C_2 is the child node of C_1, if there exists $C_3 = (A_3, B_3)$ generated by the node D_1 and one node E_3, which E_2 is the nodes in $L(K_2)$, $B_3 = intent(D_1) \cap intent(E_3)$ and $B_1 \square B_3 \square B_2$ can be concluded. Therefore, the number of intent(E_3) must be less than intent(E_2) and more than intent(E_1). Otherwise, there exists one father node E of E_2 and E_3 in $L(K_2)$, which and D_1 can generate the node C_3. It follows that the node C_2 is the direct child node of C_1.

Definition 6 The nodes in concept lattices $L(K_1)$ and $L(K_2)$ are arranged in ascending order of the intents, when the node &i in $L(K_1)$ is merged with the nodes in $L(K_2)$. If there is a new node generated, the new node is called i grade node and marked as C_{ij} , where i and j denote the *i-th* node in $L(K_1)$ and *j-th* node generated in the i grade, respectively.

For example, the node &1 in $L(K_1)$ is merged with the nodes in $L(K_2)$, if new nodes are generated, which will be marked as $C_{1,1}$, $C_{1,2}$ and so on.

Theorem 4 The nodes in concept lattices $L(K_1)$ and $L(K_2)$ are arranged in ascending order of the intents, one node in $L(K_1)$ is vertical merged with any node in $L(K_2)$, if there are new nodes generated in the same grade, then the node generated later is the child node of the node generated before, and which must be the direct child node of the node generated just before.

Proof Obviously, if there are new nodes generated in the same grade, the new nodes must be generated by the same node in one concept lattice and the different nodes in another concept lattice, from **Therom 3**, the proof is immediate.

Therefore, when the same node in $L(K_1)$ is merged with different nodes in $L(K_2)$, if there are new nodes generated, they are must be in the same grade, and there is no need to compare the directpaternity relationship between new nodes, by the generation sequence, just the edge is connected, which can be improve the efficiency of lattice construction.

3.2. **The Principle of the Algorithm.** By Definition 6 and Theorem 3, if the nodes in concept lattices $L(K_1)$ and $L(K_2)$ are arranged in ascending order of the intents, in the same grade in the concept lattice vertical merged, the child node is generated after the father node. And the node generated is the direct child node of the node generated just before. Therefore, for the the same node in $L(K_1)$ is merged with two different nodes in $L(K_2)$, if the intent intersection of the two nodes is contained in the intent of the new node generated just, there is no node generated. If the intersection contains the intent of the new node generated just, there is a new node generated, which must be the direct child node of the previous one. That is to say, in the same grade, it is only need to compare the inclusion relation between the intent intersection with the intent of the node generated just before we can judge one new node is generated or not. On the other hand, in different grades, there may be parent relationship between two nodes, hence, it is necessary to judge relation between node generated with the nodes in any other grade. The algorithm is described as follows:

Firstly, the nodes in concept lattices $L(K_1)$ and $L(K_2)$ are arranged in ascending order of the intents, and the new node generated is labeled with grades. Secondly, each node in concept lattice $L(K_1)$ is merged into the concept lattice $L(K_2)$ in sequence, only the intent intersection of the two nodes in the two concept lattices is judged. If there is a new node generated, its intent and extent are the intersecion of the intents and the union of the extents of the two nodes merged. When judging the set B with the intent of the node $C_{i,j} = (A_j, B_j)$ generated just before.

The algorithm is described as follow:

The nodes in concept lattices $L(K_1)$ and $L(K_2)$ are arranged in ascending order of connotation. It is assumed that there are m and n nodes in $L(K_1)$ and $L(K_2)$, respectively. The nodes in $L(K_1)$ are inserted into $L(K_2)$ in order, and the newly generated nodes are labeled by level. If the nodes in the concept lattice $L(K_1)$ are (A^1, B^1), and the nodes in the concept lattice $L(K_2)$ are (A^2, B^2), let $B = B^1 \cap B^2$, assuming that the node generated by the latest merging in the concept lattice $L(K)$ formed by merging is $C_{i,j} = (A_j, B_j)$ (i-level node), the following four cases are handled:

(1) If $B \subseteq B_i$, the new node must not be generated. And the next merging operation (i.e., merging the next node in (A^1, B^1) and $L(K_2)$) will be performed.

(2) If $B \supset B_i$, new nodes $C_{i,j+1} = (A^1 \cup A^2, B)$ must be generated, and $C_{i,j+1}$ and $C_{i,j}$ must be connected. We should compare the connotation of set B and i-1 level nodes from bottom to top. There are two situations: 1) When the intersection of set B and the lowest node of this level is empty set, it turns to level i-2. 2) When the intersection of the connotation of set B and the lowest node of the level is not empty set, the connotation of B and each node is compared from bottom to top from the lowest node of the level. In this case, there are two situations: First, when a node connotation is included in set B, the edge is added between the new node $C_{i,j+1}$ and the node. Then we can turn to the next level i-2 to continue the above process. Second, when intersection of a node connotation and set B is an empty set, we should turn to level i-2 to continue the above process. Until $i = 2$, the next merge operation is performed.

(3) If $B \cap B_i = \phi$, (a new node may be generated, and if the generated new node must be an i+1 level node, or it may not generate a new node), then it will turn to the i-1 level node. If set B contains the connotation of the lowest node of the level, it will not generate a new node and carry out the next merging operation. If the connotative intersection of the lowest node of the set B and i-1 level is also empty set, it will turn to level i-2 until i=2. A new node $C_{i+1,1} = (A^1 \cup A^2, B)$ is formed and connected with nodes $C_{1,1}$, and then the next merging operation is performed.

(4) If $B \cap B_i \neq \phi$, and set B and connotation B_j are not mutually inclusive, (new nodes may be generated, and if new nodes generated must be i+1 nodes, or no new nodes are generated), then there will be three situations: 1) If set B contains into the connotation of the lowest node of the level, then no new nodes will be generated and the next merging operation will be performed. 2) If the intersection of the connotation of set B and the lowest node of level i-1 is empty set, it turns to the node of level i-2. 3) If the intersection of the connotation of set B and the lowest node of i-1 level is not empty set, the connotation of set B and each node is compared from bottom to top from the bottom node of the level. When a node connotation is included in set B, a new node $C_{i+1,1} = (A^1 \cup A^2, B)$ is generated, and the new node C_{i+1} is added, Connect the edge between them. Then go to the next i-2 level, continue the above process until i=2, and then perform the next merge operation.

(5) When all nodes in $L(K_1)$ are finished, the whole process.

3.3. **Analysis of the Algorithm.** Since completeness of concept lattice, time complexity of building concept lattice is a very important aspect of formal concept analysis. Generally speaking, the construction complexity of concept lattices increases exponentially. For this problem, grade of concept lattice is defined, nodes of the two subconcept lattices are arranged in ascending order of the intents. During the vertical union of any two nodes, set B (let the intents intersecion of the two nodes be B) is compared with the intents of nodes generated just before, and with some nodes generated in some grades from bottom to top. Judging whether one new node is generated, it is no need to compare

the set B with all nodes generated before, in the same grade, only compared with node generated before. In different grades, first compared with the intent of the bottom node in one grade to determine whether further comparisons are needed. If not, turn to next grade. And if necessary, compared from bottom to top, until the intent of one node is contained in the set B (there must be a new node generated), edge between the two nodes is connected. Then turn to the next grade, until all the grades defined before are judged and then the next union operation begins.

When judging a new node generated or not, the direct paternity relationship between new nodes can be also judged, that is to say, the edge of two nodes can be renewed. Hence, we can not only obtain all new generated nodes and the Hasse diagram of the concept lattice merged, but also there is no redundant node.

4. **Experiments.** Consider the formal context $K_1 = (O_1, D, R)$ and $K_2 = (O_2, D, R)$ shown in Table 1 and Table 2, where $O_1 = \{1, 2, 3\}$, $O_2 = \{4, 5\}$, $D = \{a, b, c, d, e\}$, their Hasse diagrams are Figure 1 and Figure 2.

TABLE 1. Formal Context $K_1 = (O_1, D, R)$

	a	b	d	e
1	1	0	1	0
2	1	1	1	0
3	0	1	0	0

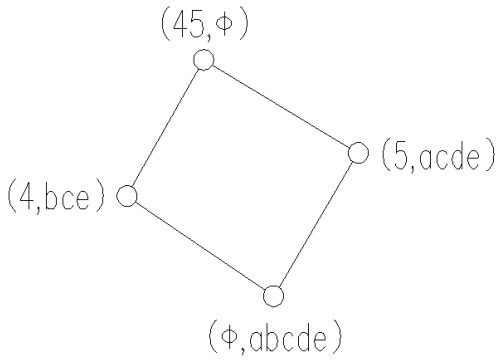

FIGURE 1. Hasse Diagram of $L(K_1)$

Arrange the nodes of $L(K_1)$ and $L(K_2)$ in ascending order of the intents, which numbers are also shown in the Figures. Now according to the above description of the algorithm, insert the nodes of $L(K_2)$ into the concept lattice $L(K_1)$ in turn, the process is described in following Table 3.

In the union process, all nodes in the new concept lattice can be generated, at the same time, the Hasse diagram of the new concept lattice is obtained, which is shown in Figure 3. The colors represent the different grades of the new nodes. There is no need to compare one new generated node with all generted nodes before, we can judge whether it is a new node and its all direct parent node, which can greatly improve the construcion efficiency.

For vertical union of concept lattices, the number of the splitting of formal backgrounds will affect the speed of the algorithm. It's not that the thinner the splitting, the better, or the less the splitting, the better. So what criteria can be followed for splitting formal background will be a future study.

TABLE 2. Formal Context $K_2 = (O_2, D, R)$

	a	b	c	d	e
4	0	1	1	0	1
5	1	0	1	1	1

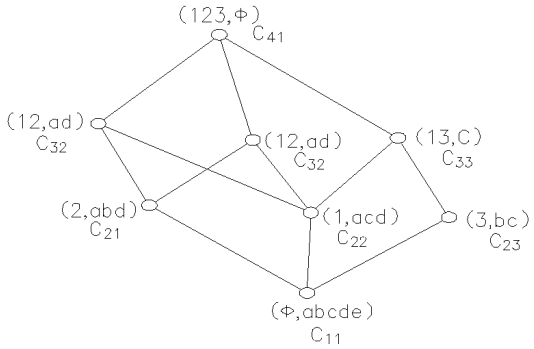

FIGURE 2. Hasse Diagram of $L(K_2)$

TABLE 3. Union Process

Vertical union	$B_i \cap B_j$	Node generated	No.	grade
&1 and #1	$\{\phi\}$	(12345, ϕ)	$C_{1,1}$	1
&1 and #2	–	–	–	–
&1 and #3	$\{c\}$	(1345, c)	$C_{1,2}$	1
&1 and #4、#5、#6、 #7	–	–	–	–
&1 and #8	$\{ce\}$	(45, ce)	$C_{1,3}$	1
&2 and #1	–	–	–	–
&2and #2	$\{b\}$	(234, b)	$C_{2,1}$	2
&2 and #3、#4	–	–	–	–
&2 and #5	$\{bc\}$	(34, bc)	$C_{2,2}$	2
&2 and #6、#7	–	–	–	–
&2 and #8	$\{bce\}$	(4, bce)	$C_{2,3}$	2
&3 and #1、#2、#3	–	–	–	–
&3 and #4	$\{ad\}$	(125, ad)	$C_{3,1}$	3
&3 and #5	–	–	–	–
&3 and #6	$\{acd\}$	(15, acd)	$C_{3,2}$	3
&3 and #7	–	–	–	–
&3 and #8	$\{acde\}$	(5, $acde$)	$C_{3,3}$	3
&4 and #1、#2、#3、#4、#5、#6	–	–	–	–
&4 and #7	$\{abd\}$	(2, abd)	$C_{4,1}$	4
&4 and #8	$\{a-e\}$	(ϕ, $a-e$)	$C_{4,2}$	4

5. **Conclusion.** A vertical union algorithm is introduced in the paper. During the construction, it only needs to compare the intent intersection of two nodes in two sublattices with the intent of the just generated node or the intents of some nodes in some grades from the top to the bottom. A new node generated can be judged, and Hasse diagram of concept lattice united can be got. There are less comparison and judging steps and no redundant nodes, which can improve the efficiency.

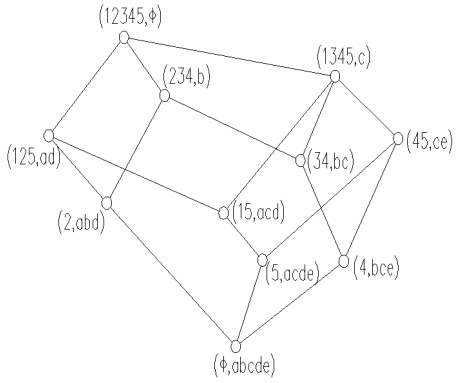

FIGURE 3. Hasse Diagram of $L(K)$

Acknowledgment. This work was supported by the Key Natural Science Research Projects of Anhui Education Department (No. KJ2018A0598, KJ2019A0876) and the Research Project of Anhui Xinhua University (No.2018xxk13, 2016jxtdx03). It was also supported by Shenzhen Science and Technology Plan Fundamental Research Funding JCYJ20180306171938767 and Shenzhen Foundational Research Funding JCYJ2018050718 3527919.

REFERENCES

[1] Wille R. Restructuring lattice theory: an approach based on hierarchies of concepts[M]. *Dordrecht: Reidel Publishing Company*, 83:445-470, 1982.

[2] Tzung-Pei Hong, Jimmy Ming-Tai Wu, Yan-Kang Li, and Chun-Hao Chen. Generalizing Concept-Drift Patterns for Fuzzy Association Rules[J]. *Journal of Network Intelligence*, 3(2): 126-137, May 2018.

[3] Xu WH, Li WT. Granular computing approach to two-way learning based on formal concept analysis in fuzzy datasets [J]. *IEEE Transactions on Cybernet ics*, 46(2): 366-379, 2016.

[4] Mehdi Kaytoue, Sergei O Kuznetsov, Amedeo Napoli, etal, Mining Gene Expression Data with Pattern Structures in Formal Concept Analysis[J]. *Information Sciences*, 1819(10): 1989-2001, 2011.

[5] Jin-hai Li, Chang-lin Mei, Yue-jin Lv. Knowledge reduction in real decision formal contexts[J]. *Information Sciences*, 189(15):191-207, 2012.

[6] Chung-Ming Kuo, Nai-Chung Yang, Shen-Cha Tseng, Meng-Tso Chen, A Novel Texture Descriptor for Texture Image Retrieving[J]. *Journal of Network Intelligence*, 3(4): 278-290, November 2018.

[7] Prem Kumar Singh, C.Aswani Kumar, Jinhai Li. Knowledge representation using interval-valued fuzzy formal concept lattice[J]. *Soft Computing*, 20(4):1485-1502, 2016.

[8] Prem Kumar Singh. Medical diagnoses using three-way fuzzy concept lattice and their Euclidean distance[J]. *Computational and Applied Mathematics*, 37(3):3283-3306, 2018.

[9] Bao Yulai. Research on Construction of Knowledge Base and Knowledge Discovery of Traditional Mongolian Medicine Based on Domain Ontology[D]. *Changchun: Jilin University*, 2018.(in Chinese)

[10] Liu Yang. Prediction of stability of roadway surrounding rock based on concept lattice and probabilistic neural network[D]. *Wuhan: Wuhan University of Science and Technology*, 2018.(in Chinese)

[11] Aswani Kumar C. Fuzzy clustering based formal concept analysis for association rules mining[J]. *Applied Artifical Intellgence*, 26(3):274-301, 2012.