The Decomposition Algorithm of Concept Lattice Based on Hierarchy

Hai-Xia Li[1] Lin-Lin Tang[2]

[1] Department of Public Teaching, Anhui Xinhua University
Hefei 230088, China
`learain1@126.com`

[2] Harbin Institute of Technology Shenzhen Graduate School
Shenzhen 518055, China
`hittang@126.com`

Received 13 July 2014; Revised 10 November 2014; Accepted 1 December 2014

Abstract. The decomposition of concept lattice is a very important part in the formal concept analysis. In this paper, the hierarchy of concept lattice was introduced and some conclusions were given, then the decomposition algorithm of concept lattice was proposed which used the hierarchy of concept lattice, which not only reduces the comparison, but also no abundant node is created. At the same time, the Hasse diagram is produced.

Keywords: Concept Lattice, Node, Sub-concept Lattice, Decomposition, Layer, Extent, Path

1 Introduction

Since Professor Wille put forward formal concept analysis[1] in 1982, as the data structure, concept lattice can show the hierarchy relation between concepts, which has been widely used in data mining, software engineering, information retrieval and clustering analysis[2], [3], [4].

In real life, the data is very huge and complex, the processing is very inconvenient. Therefore, it is need to decompose the concept lattice built on the data background into some independent sub-concept lattices and then processed, which can bring great convenience for data analysis and rule extraction. At the present, there hasn't been much research on this aspect. In literature [5], there will be a lot of redundancy during the decomposition, which will greatly increase the complexity of the problem itself; in Literature [6], the decomposition algorithm proposed contains only an object or attribute, when the objects and attributes of interest are more, the decomposition of efficiency may be greatly reduced; in the paper [7], during the nodes generated in sub-concept lattice, the comparison is more and can be reduced further.

In this paper, the hierarchy of concept lattice was introduced and some conclusions was given, then the decomposition algorithm of concept lattice was proposed which used the hierarchy of concept lattice. The algorithm arranges the nodes of concept lattice in ascending order of the extent, if the intersection of the extent appears for the first time, the node in sub-concept lattice is generated; in judging the first appearing or not, only comparing the extents of the nodes which are generated in the same layer before, the comparison with all the nodes generated before is unnecessary, which not only reduces the comparison, but also no abundant node is created. At the same time, the Hasse diagram is produced.

2 Preliminaries

Definition 1 [5] 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 the binary relation between O and D. For $A \subseteq O$, $B \subseteq D$, the mapping is defined as

$$A' = \{m \in D \mid \forall g \in A, (g,m) \in R\};$$

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

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

Definition 2 [5] If $C_1=(A_1,B_1)$ and $C_2=(A_2,B_2)$ is two nodes and $A_1\supseteq A_2$ ($\Leftrightarrow B_1\subseteq B_2$), C_2 is called the child node of C_1, is the father node of C_2, which is denoted as $C_1\geq C_2$. If there is no node C_3 which satisfies $C_1\geq C_3\geq C_2$, then C_2 is called the direct child node of C_1 and C_1 is the direct father 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).

Theorem 1 [5] For the concept lattices $L(K_1)$ and $L(K_2)$, where $K=(O,D,R)$, $K_1=(O_1,D,R)$, $K_2=(O_2,D,R)$, $O_1\cup O_2$, $O_1\cap O_2=\phi$, then $L(K)$ can be mapped to concept lattices $L(K_1)$ and $L(K_2)$ through the function ϕ

$$\phi((A,B))=((A\cap O_1,(A\cap O_1)'),(A\cap O_2,(A\cap O_2)')) \tag{1}$$

Theorem 2 [7] For $\phi(C)=(C_1,C_2)$, the node $C\in L(K)$, $C_i\in L(K_i)$ and $i=1,2$. If the extent of the nodes generated by all child nodes of the node C is not equal to the extent (C_i) ($i=1$ or $i=2$), then intent (C_i)=intent(C).

Theorem 3 [7] All the nodes of the concept lattice are arranged in ascending order of the extent base, one concept lattice is decomposed into sub-concept lattices with the same attributes sets. If the extent of the node C_1 generated from the decomposition of the node C already exists, then the node C_1 can be ignored in the process of the decomposition of C.

Theorem 4 [7] One concept lattice is decomposed into sub-concept lattices with the same attributes sets if its nodes are arranged in ascending order of the extent base, then the child nodes of one node of sub-lattices are absolutely generated before its father nodes.

3 The Algorithm Based on Hierarchy

3.1 The Definition and Theorem

Definition 3 In concept lattice, the ordered node is defined as follows:
for the node $C=(A,B,\tau)$, where A= extent(C), B=intent (C), τ is the path between the node C 錯誤! 找不到參照來源。 and the smallest node.

Definition 4 In concept lattice, the node C 錯誤! 找不到參照來源。is called the node in the $s+1$ 錯誤! 找不到參照來源。 layer, if the length of the greatest path between C 錯誤! 找不到參照來源。 and the smallest node is s 錯誤! 找不到參照來源。.

Theorem 5 There is no parent-child relation between the nodes in the same layer.

Proof. For any two nodes C_1 and C_2, which are in the same layer, assume that there is parent-child relationship between C_1 and C_2, then there exists edge between the two, by the Definition 4, the length of the greatest chain C_1 and C_2 to the smallest node is not equal. That is to say, C_1 and C_2 is not at the same layer, so the nodes in the same layer is incomparable.

Theorem 6 Except the smallest node, any node is the direct father node of one node in the next layer.

Proof. For any node C_1 which is in the s layer, assume that C_1 is not a parent node of any node in the $s-1$ layer. By theorem 4, child nodes are produced before their parent nodes, it is deduced that C_1 is the direct parent node in the $s-2$ layer or the below; on the other hand, by definition 4, we can conclude that C_1 is at most the node in the $s-1$ layer. This is contradiction, so C_1 is the direct parent node of one node in the $s-1$ layer.

Corollary 1 If there is inclusion between the extents of two nodes, the two nodes are in different layers.

3.2 The Principle of the Algorithm

According to the above theorems, firstly, the nodes in a concept lattice are arranged in ascending order of the extents, if the extent cross appears for the first time, the new node in sub-lattice is generated and its extent is the extent cross and its intent is equal to the intent of the node decomposed; secondly, the child nodes are produced before their parent nodes, and any node is the direct father node of one node in the next layer; finally the nodes in the same layer are incompatible. So, arrange the nodes in a concept lattice in ascending order of the extents, only just compare the extent cross with the extent of the nodes at the same layer(supposed in the s layer), if the extent cross first appears and is not included by the extents of the nodes generated before, the new node in sub-lattice is produced and is at the s layer; on the other hand, if the extent cross first appears and is included by the extent of a node (in the s layer) generated before, the new node in sub-lattice is also produced and is at the $s+1$ layer. Denoted the new node generated of the sub-lattice as C, during the comparison, find out all the direct child nodes at the upper layer, add the number after any path of all the direct child nodes, as the path of the node C. Thus, layer after layer, we can get all the nodes of the sub-lattice, connect edges by the path of the greatest node, the Hasse diagrams of the sub-lattice can be simultaneously obtained. In the decomposing process, only compare the extents of the nodes generate before at the same layer and the upper layer, the former comparison can judge the node C is at a new layer or not, the latter comparison can ensure the parent-child relationship with the nodes at the upper layer. That is to say, the comparison with the extent of all nodes of the sub-lattice generated before is unnecessary, which can greatly improve the decomposing efficiency.

3.3 The Algorithm Based on Hierarchy

Input: concept lattice $L(K)$, the object sets O_1 and O_2

Output: sub-concept lattices $L(K_1)$ and $L(K_2)$ and the respective Hasse diagrams.

```
Begin
Int k;
Int , t
Int m[][], n[][];
String H[][], L[][];
String A[], B[];
Int P[];
Input k; L[1][1]=null; H[1][1]=null; m[1][1]=0; n[1][1]=0;
      s=1; t=1; P[1]=1; H[1][1]= A[1]; L[1][1]=B[1];
Int kk; kk=t;
    For (i=2; i is less than or equal to k; i++)
        T=0
        For (j=1; j<kk+1; j++)
            If A[i] is contained in H[s][j]
              T=0
              H[s+1][1]=A[i]
```

```
                    L[s+1][1]=B[i]
                    m[s][j]=s+1; m[s][j]++;
                    n[s][j]=s+1; n[s][j]++;
                    P[s+1]=1; T=1
                    For (int h=j+1; h<kk+1; h++)
                        If A[i] is contained in H[s][j]
                            m[s][h]=s+1; m[s][h]++;
                            n[s][h]=1; m[s][h]++;
                        Endif
                    Endfor
                Endfif
            Endfor
        Endfor
        If  j=t and T=0
            t++; P[s]=t;
            H[s][t]=A[i];
            L[[s][t]=B[i]
            If  s>1
                Int  h=P[s-1]
                For (int l=1; l<h+1; l++)
                        If  H[s-1][1] is contained in A[i]
                            m[s-1][1]=s; m[s-1][1]++;
                            n[s-1][1]=1; n[s-1][1]++;
                        Endif
                Endfor
            Endif
            Else
                s++; kk=P[s]
            Endif
      Endfor
    Endfor
End
```

4 Example

Consider the formal context $K=(O,D,R)$ shown in Table 1, where $O=\{1,2,3,4,5\}$, $D=\{a,b,c,d,e\}$, its Hasse diagram is shown in Figure 1, the sub=formal contexts decomposed are $K_1=(O_1,D,R)$ and $K_2=(O_2,D,R)$, where $O_1=\{1,2,3\}$ and $O_2=\{4,5\}$ Arrange the nodes of $L(K)$ in ascending order of the extents which are decomposed as below.

Table 1. The formal context $K=(O,D,R)$

	a	*b*	*c*	*d*	*e*
1	1	0	1	1	0
2	1	1	0	1	0
3	0	1	1	0	0
4	0	1	1	0	1
5	1	0	1	1	1

For the node $(\phi, abcde)$, $\phi \cap O_1 = \phi$, the extent ϕ first appears in the sub-lattice $L(K_1)$, so $(\phi, abcde)$ is one new node and in the first layer, let $s=1$, denote $C_{11} = (\phi, abcde)$ and the ordered node $(\phi, abcde, C_{11})$;

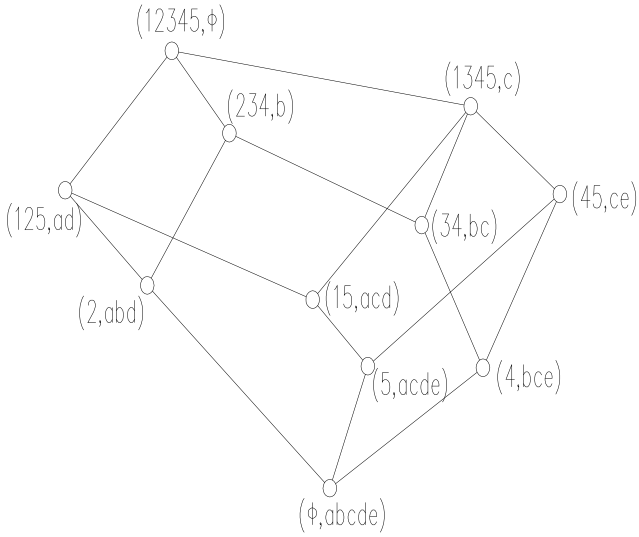

Fig. 1. Hasse diagram of the formal context 1

For the node $(5, acde)$, $\{5\} \cap O_1 = \phi$, compare with ext(C_{11}), so no new node is generated;

For the node $(4, bce)$, there is no new node generated, too;

For the node $(2, abd)$, $\{2\} \cap O_1 = \{2\} \supset$ ext(C_{11}), let $s=2$; in the first layer find out its direct child node C_{11}, therefore, denote $C_{21} = (2, abd)$ and the ordered node $(2, abd, C_{11}C_{21})$;

For the node $(15, acd)$, $\{15\} \cap O_1 = \{1\}$, compare $\{1\}$ and ext(C_{21}) first, they do not contain each other, keep $s=2$; then compare with ext(C_{11}), $\{15\} \cap O_1 = \{1\} \supset$ ext(C_{11}), i.e. C_{11} is the direct child node, so $(1, acd)$ is another new node, denote $C_{22} = (1, acd)$ and the ordered node $(1, acd, C_{11}C_{22})$;

For the node $(45, ce)$, generate no new node;

For the node $(34, bc)$, $\{34\} \cap O_1 = \{3\}$, compare $\{3\}$ with ext(C_{21}) and ext(C_{22}) first, they do not contain each other, keep $s=2$; then compare with ext(C_{11}), $\{3\} \supset$ ext(), i.e. C_{11} is the direct child node, so generate a new node $C_{23} = (3, bc)$ and the corresponding order node is $(3, bc, C_{11}C_{23})$;

For the node $(234, b)$, $\{234\} \cap O_1 = \{23\}$, compare $\{23\}$ with ext(C_{21}),ext(C_{22})and ext(C_{23}) first, $\{23\} \supset$ ext(C_{23}) and $\{23\} \supset$ ext(C_{21}), so let $s=3$,and meanwhile C_{21} and C_{23} are the direct child nodes, generate a new node $C_{31} = \{23, b\}$ and the corresponding order node is $\{23, b, C_{11}C_{21}C_{31}, C_{11}C_{23}C_{31}\}$;

For the node $(125, ad)$, $\{125\} \cap O_1 = \{12\}$, compare $\{12\}$ with ext(C_{31}) first, they do not contain each other, keep $s=3$; then compare with ext(C_{21}), ext(C_{22}) and ext(C_{23}), find out the direct child nodes are C_{21} and C_{22}, so generate a new node $C_{32} = (12, ad)$ and the corresponding order node is $(12, ad, C_{11}C_{21}C_{32}, C_{11}C_{22}C_{32})$;

For the node $(1345, c)$, generate a new node $C_{33} = (13, c)$ and the corresponding order node is $(13, c, C_{11}C_{22}C_{33}, C_{11}C_{23}C_{33})$;

For the node $(12345, \phi)$, let $s=4$, generate a new node $C_{41} = (123, \phi)$ and the corresponding order node is

$(123,\phi,C_{11}C_{21}C_{31}C_{41},C_{11}C_{23}C_{31}C_{41},C_{11}C_{21}C_{32}C_{41},C_{11}C_{22}C_{32}C_{41},C_{11}C_{22}C_{33}C_{41},C_{11}C_{23}C_{33}C_{41})$, which is the greatest node of $L(K_1)$.

According to the path of the greatest node, we can get the Hasse diagrams of the sub-lattice $L(K_1)$, which is shown in Fig.2. The nodes of sub-lattice $L(K_2)$ can be obtained similarly, the greatest order node is $(45,ce,C'_{11}C'_{21}C'_{31},C'_{11}C'_{22}C'_{31})$.

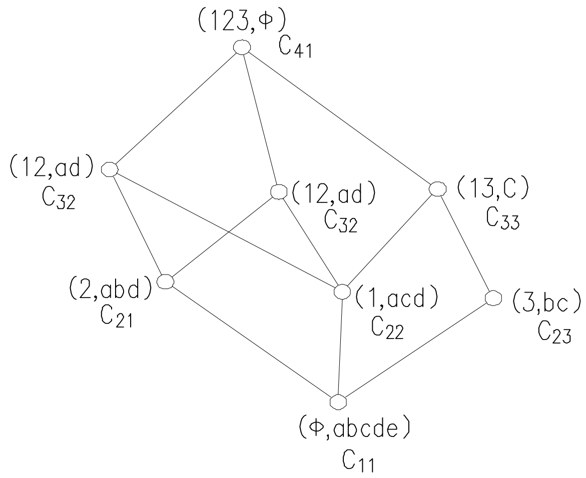

Fig. 2. Hasse diagram of sub-concept lattice $L(K_1)$

The comparison of the concept nodes is the most important factor determining the efficiency in the decomposition. Consider one concept lattice $L(K)$ which is to be decomposed into two sub-lattices $L(K_1)$ and $L(K_2)$ with the same attributes set. In $L(K_1)$, suppose that $L(K_1)$ consists of s layers, the nodes of each layer are N_1, N_2, ..., N_s, respectively. Based on the hierarchy, compare to the decomposition method in the paper [8], the comparison times about can reduce

$$N_3N_1 + N_4(N_1 + N_2) + \cdots + N_s(N_1 + N_2 + \cdots + N_{s-2})$$

If $N_1 + N_2 + \cdots + N_s = n$, the times of comparison can probably be reduced $L(K_1)$ times. Therefore, based on the hierarchy, the larger the sub-lattice, the more obvious the efficiency.

5 Conclusion

This paper discusses the decomposition of concept lattice, based on hierarchy, one new method is proposed, because the comparison with the extents of all nodes of the sub-lattice generated before is unnecessary, the constructing efficiency of sub-concept lattices can improve.

6 Acknowledgement

The authors are grateful to the National Natural Science Foundation of China, grant 61202456, Anhui Province Natural Science Project, grant KJ2013B107, and the Project of Xinhua University, grant 2014zr011, 2013xgg05, 2012jpkcx03.

References

[1] R. Wille, "Restructuring Lattice Theory: an Approach Based on Hierarchies of Concepts," in *Ordered Sets*, pp.445-470,1982

[2] J. Li, C. Mei, Y. Lv, "Knowledge Reduction in Real Decision Formal Contexts," *Information Sciences*, Vol. 189, No. 15, pp.191-207, 2012

[3] M. Kaytoue, S. O Kuznetsov, A. Napoli, S. Duplessis, "Mining Gene Expression Data with Pattern Structures in Formal Concept Analysis," *Information Sciences*, Vol. 181, No. 10, pp. 1989-2001, 2011.

[4] K. Qu, Y. Zhai, J. Liang, M. Chen, "Study of Decision Implications Based on Formal Concept Analysis," *Journal of General Systems*, Vol. 36, No. 2, pp. 147-156, 2007.

[5] B. Ganter, R .Wille, *Formal Concept Analysis*, Springer Press, 1999.

[6] X. Bai, Y. Ma, X. Zhang, "Concept Lattice of Simplest Subdirect Concept and Algorithm (in Chinese)," *Journal of Anshan University of Science and Technology*, Vol. 27, No. 6, pp. 428-431, 2004.

[7] H. Li, J. Ding, D. Nie, L. Tang, "One Decomposition of Concept Lattice," in *Proceedings of the 8th International Conference on Genetic and Evolutionary Computing*, Vol. 329 of Advances in Intelligent Systems and Computing, pp. 325-333, 2014.