A Novel Concept Lattice Merging Algorithm Based on Collision Detection

Caifeng Zou[1], Jiafu Wan[2(✉)], and Hu Cai[3]

[1] College of Information Engineering, Guangdong Mechanical & Electrical College, Guangzhou, China
caifengzou@gmail.com
[2] School of Mechanical and Automotive Engineering, South China University of Technology, Guangzhou, China
jiafuwan_76@163.com
[3] College of Electrical Engineering and Automation, Jiangxi University of Science and Technology, Ganzhou, China
396210149@qq.com

Abstract. Concept lattice has been widely used in machine learning, pattern recognition, expert systems, computer networks, data analysis, decision analysis, data mining and other fields. The algorithms of constructing concept lattices are introduced. This work proposes a novel concept lattice merging algorithm based on collision detection, which can remove the redundant information in distributed construction of concept lattice. Further research to distributed concept lattice construction algorithm is needed.

Keywords: Concept lattice · Distributed construction · Merging algorithm · Collision detection

1 Introduction

The concept is the basic unit of human cognition and an important research object of artificial intelligence disciplines. German mathematician Wille proposed Formal Concept Analysis(FCA) in 1982 [1]. He systematically studied the hierarchies of concepts, properties of lattice algebra, and the isomorphic nature of concept lattice and formal context, which established foundations for the field of Formal Concept Analysis (FCA).

FCA is a powerful tool for data analysis and rule extraction from the formal context. FCA expresses concepts, attributes, and relationships of the ontology with formal context. According to the context, concept lattice is constructed to show the structure of the ontology clearly, and describe the generalization and specialization of the concept. Concept Lattice, also known as Galois Lattice, is the core data structure of FCA. In concept lattice, each node is a formal concept. Formal concept consists of extension part and intension part [2]. Extension of the concept is considered as the set of all objects belonging to this concept, and intension is considered as the set of the common characteristics or attributes of all these objects [3]. Concept lattice essentially describes the affiliation between objects and features, and shows the relationships

V.C.M. Leung et al. (Eds.): TridentCom 2014, LNICST 137, pp. 489–495, 2014.
DOI: 10.1007/978-3-319-13326-3_48

of generalization and specialization between the concepts. The corresponding Hasse diagram contributes to data visualization.

The basic concepts of FCA include formal context, concept lattice, Hasse diagram, senior concept and parental concept, sympatric formal context and sympatric concept lattice, independent context and independent concept lattice, and so on [4].

(1) Formal context

Formal context is defined as a triple K (U, A, I), where U is a set of objects, A is a set of attributes, and I is a binary relation between object U and attribute A, ie. $I \subseteq U \times A$. If there is $(x, a) \in I$, then xIa shows that object x has attribute a. The form of two-dimensional data table is also a type of formal context. The tuple represents object or instance, and the column represents attribute.

When there is
$$X^* = \{a \mid a \in A, \forall x \in X, xIa\}, X \subseteq U$$
$$B^* = \{x \mid x \in U, \forall a \in B, xIa\}, B \subseteq A$$

if $\exists X^* = B$ and $B^* = X$, then (X, B) is called a formal concept or simply a concept. X is defined as the extension of concept (X, B). B is defined as the intension of concept (X, B). Extension of the concept indicates the set of all objects belonging to this concept, and intension of the concept indicates the set of the common attributes of all these objects. For example, $C((1,5),\{b,c,e\})$ indicates that concept C covers two objects 1 and 5. The common attribute of these two objects is $\{b,c,e\}$.

(2) Concept lattice

Concept lattice is used to indicate the relationship between attributes and objects. There is a kind of partially ordered relationship between the nodes of concept lattice. Given $C_1(X_1, B_1), C_2(X_2, B_2)$, then $C_1 < C_2 \Leftrightarrow B_1 < B_2$.

This partially ordered relationship means that $C_1(X_1, B_1)$ is a senior concept of $C_2(X_2, B_2)$, or $C_1(X_1, B_1)$ is a generalization of $C_2(X_2, B_2)$.

For formal context (U, A, I), there is a unique partially ordered set in relationship I. This partially ordered set produces a lattice structure. Lattice L generated from the context (U, A, I), is called the concept lattice. The concept lattice of a formal context is unique.

2 Main Construction Algorithms of Concept Lattice

Algorithm for constructing concept lattice is the basis for the concept lattice research. Concept lattice construction is a concept clustering process. The completeness of the concept lattice means the concept lattice generated from the same data is unique. The current concept lattice construction algorithms can be divided into three categories: batch processing algorithm, incremental algorithm and distributed algorithm. The first two algorithms are stand-alone construction algorithms, in which the incremental algorithm is considered to be more promising. With the rapid growth of data-scale, distributed algorithm for constructing concept lattice has also become an important research content.

2.1 Batch Processing Construction Algorithm

Batch algorithm generates all nodes at a time, then generates edges according to the relationship of direct predecessor and direct successor of nodes, and establishes the whole concept lattice ultimately. There are many batch processing algorithms of constructing the concept lattice, in which only a few algorithms can generate Hasse diagram.

The main idea of Bordat algorithm [5] is top-down construction of the lattice starting from supremum. Firstly the topmost node is established, then all the child nodes of the topmost node are generated, and the child nodes are added to the lattice and connected to the parent node. Then the processes are executed iteratively for each child node. The defect of Bordat algorithm is: the number of repeated emergence of each child node (concept) is equal to the number of its parent nodes in the final concept lattice. This method is not suited to the concept lattice construction of large-scale formal context. Bordat algorithm can generate the concept lattice and Hasse diagram.

Ganter algorithm [6] uses feature vectors to enumerate the attribute sets of the lattice. The length of each vector is the cardinality of the attribute set. If the value of an attribute appears in the vector, then the corresponding bit is set to 1, otherwise it is set to 0. This algorithm does not generate Hasse diagram.

Chein algorithm [7] is a bottom-up lattice construction algorithm. The algorithm starts from the first layer l_1, which consists of the set of all of the pairs $(\{x\}, f(\{x\}))$ of X. Then it uses an iterative approach to construct the concept lattice from down to up layer by layer. It merges two pairs in layer l_k to create a new pair in layer l_{k+1}. The merging process is to find the intersection of all pairs in layer l_k , and test whether the intersection has appeared before. If the intersection has appeared in the upper layer, then the intersection in the upper layer is not complete pair, and should be marked for remove at the end of this layer. Chein algorithm does not generate Hasse diagram.

2.2 Incremental Construction Algorithm

The idea of the incrementally constructing concept lattice is: firstly the concept lattice is initialized to be the whole concept and the empty concept, and then the concept lattice is incrementally constructed using different operations according to the difference of the intersection of inserted object's attributes and the intension of the concept lattice nodes. When the context changes, such as adding an instance, the incremental construction method can maximize the use of existing concept lattice to avoid constructing the lattice from the beginning each time.

Godin algorithm [8] is a typical incremental concept lattice construction algorithm. This algorithm starts from a single object, and the new object is added into the lattice one by one, with only the necessary structural updates each time. This method can produce not only complete pair of the concept lattice, but also Hasse diagram.

Kuznetsov [9] pointed out that Godin algorithm is more suitable for sparse formal context. When the formal context becomes dense, the performance of Godin algorithm declines sharply.

Z. Xie [10] proposed an incremental algorithm for constructing concept lattice which organized the concept lattice nodes by tree structure. Y. Jiang [11] proposed an incremental concept lattice construction algorithm based on the list structure, which use list structure to organize the lattice nodes, and use the index table to achieve a quick update on concept lattice. H. Mao [12] presented a concept lattice incremental construction algorithm based on the binary tree structure according to the features of a certain kind of concept lattice.

2.3 Distributed Construction Algorithm

With the development of distributed systems and database technology, distributed computing, parallel computing and cloud computing has become the mainstream technologies [13, 14]. In practical applications, mass data distributed storage technology has been used very widely. For large databases, batch processing and incremental construction method of concept lattice still need to cost a lot of time. It has become a hot topic to get the global concept lattice from distributed database and establish the whole structure of the concept lattice.

P. Valtchev [15] proposed a divide and conquer method to construct concept lattice. It forms distributed multiple sub-contexts through the split of the formal context, then constructs the corresponding sub-lattices, and combines the sub-lattices to obtain a complete concept lattice. Y. Li [16], L. Zhang [17], and W. Wang [18] also studied the distributed algorithm for constructing concept lattice.

3 Concept Lattice Merging and Distributed Construction

With the development of cloud computing and big data processing technology, distributed storage and parallel data processing has become an inevitable trend. For concept lattice construction of big data, it is necessary to break the formal context up into several sub-sets, and then construct and merge them.

When a formal context splits into multiple sub-contexts, the corresponding concept lattice is called the sub-concept lattice. Concept lattice corresponding to formal context can be obtained by merging the sub-context concept lattices. This construction method of concept lattice uses the divide and conquer strategy, namely the distributed construction model of concept lattice. Concept lattice distributed construction is based on the form context merging. For example, when the company establishes workers file (formal context), each employee will fill in some fixed contents (properties) and hand to the department manager, then the department manager organizes the sector information (sub formal context) and turn them over to the company personnel department.

For the formal context merging, Wille proposed overlay and juxtaposition [2]. Overlay is vertical merging of the formal contexts, which possess the same attribute items and the different object domains. Juxtaposition is horizontal merging of the formal contexts, which have the different attribute items and the same objects domains. In distributed construction of concept lattice, the formal context is split to construct the corresponding sub-lattices, then the generated sub-lattices are merged to obtain the complete concept lattice. Overlay and juxtaposition of formal context is based on the consistency of extension or intension, and the consistent formal contexts need some processing before merging. Maedche proposed the similarity method in 2002 [19].

Distributed construction of concept lattice requires a lot of comparisons in the merging of sub-lattices. Some useless comparisons are redundant information, and can not affect the structure of concept lattice. The redundant comparisons will reduce the performance of the algorithm. The larger scale data will result in the more lattice nodes, and the more redundant information. Elimination of redundant information can significantly improve the distributed construction efficiency of concept lattice.

4 Concept Lattice Merging Algorithm Based on Collision Detection

Because of completeness of concept lattice, the distributed concept lattice construction algorithm often need to search for a large number of irrelevant concepts, which will increase the number of comparisons in the construction process, and reduce construction efficiency, but will not affect the structure of concept lattice. These irrelevant concepts are called redundant information. The formal context of distributed concept lattice is usually constructed by the massive high-dimensional data, which will generate a lot of redundant information. It is necessary to improve the distributed construction algorithm of concept lattice to remove redundant information and improve construction efficiency. This paper presents the method of removing redundant information in distributed merging of concept lattice. The collision detection technology is used to eliminate redundant information generated in the sub-lattice merging, and reduce the duplicate comparison of concept intension, in order to improve the construction efficiency of concept lattice.

The basic unit of the collision is concept C1, C2, ..., Cn. If there is an association between the two concepts, such as a common property, then a collision will occur. For example, if two concepts Ci = (Ui, Ai) and Ci' = (Ui', Ai ') comprise at least one common attribute, then the collision between Ci and Ci' will occur.

In the merging of concept lattice, the detection sub-process is executed step by step, and the time interval between adjacent detection steps is a constant which is set as t. Meanwhile, the collision weight from concept Ci to Ci' is denoted by w(Ci, Ci', t) in step t, and the merging probability of concept Ci and Ci' is denoted by P(Ci, t) and P (Ci ', t). In step t, the change rate of P(Ci, t) is the accumulation result of collision between concept Ci and other related concepts as shown in Figure 1.

As Figure 1 shown, the collision effect of the concept Ci' to Ci is defined as the product of collision weight of the concept Ci' to Ci and the merging probability of the concept Ci', i.e., w(Ci', Ci, t)*p*(Ci', t). For concept Ci, the collision effect of the concept

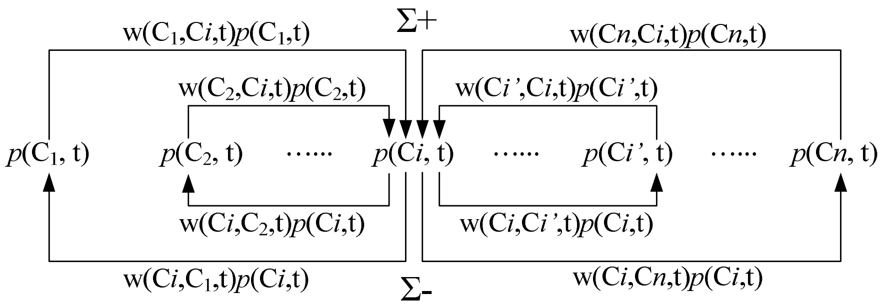

Fig. 1. Collision between concept Ci and other association concepts

Ci' to Ci is positive, which can enhance the merging probability of concept Ci, while the collision effect of the concept Ci to Ci' is negative, which can reduce the merging probability of concept Ci. The difference of positive effect and negative effect is the integrated effect. For the concept Ci, the integrated collision effect between concept Ci' and Ci is w(Ci' , Ci, t) *p*(Ci' , t)- w(Ci , Ci', t) *p*(Ci , t). The change rate of *p*(Ci, t) is the accumulation result of collision effect between concept Ci and other related concepts. If the collision directions between concepts are not considered, then the collision reaction equation of concept Ci is defined as follows:

$$\frac{\partial p(c_i,t)}{\partial t}=\sum_{i'=1}^{n}[w(c_{i'},c_i,t)p(c_{i'},t)-w(c_i,c_{i'},t)p(c_i,t)] \tag{1}$$

$\partial p(c_i,t)/\partial t$ is the change rate of $p(c_i,t)$. According to Euler equation, the merging collision reaction equation of concept lattice can be expressed as follows:

$$p(c_i,t+1)=p(c_i,t)+h\sum_{i'=1}^{n}[w(c_{i'},c_i,t)p(c_{i'},t)-w(c_i,c_{i'},t)p(c_i,t)] \tag{2}$$

According to equation (2), $p(c_i,t+1)$ can be calculated by iteration pattern:

$$p(c_i,t+1)=p(c_i,t)+h\cdot p_t^{'}(c_i,t) \tag{3}$$

$p_t^{'}(c_i,t)=\partial p(c_i,t)/\partial t$, wherein, h is the iteration step length and is set to be 1.

The target of collision reaction is expanding the difference between the merging probabilities of the concepts. When the change of the merging probability of the concept is small, the collision reaction ends and the final result of the collision reaction is obtained. Based on the collision reaction result, the concept lattice can be effectively merged.

5 Conclusions

Concept lattice is gaining more and more attention of the researchers because of its unique advantages. It has been widely used in machine learning, pattern recognition, expert systems, computer networks, data analysis, decision analysis, data mining and other fields. However, it is still a young and rapidly developing field. There are many problems on concept lattice needed to study deeply, such as distributed concept lattice construction algorithm, concept lattice merging algorithm, and so on. This work proposes a novel concept lattice merging algorithm based on collision detection, which can remove the redundant information in distributed construction of concept lattice. Further research to distributed concept lattice construction algorithm is needed.

Acknowledgment. The authors would like to thank the National Natural Science Foundation of China (No. 61262013), the High-level Talent Project for Universities, Guangdong Province, China (No. 431, YueCaiJiao 2011), and the Open Fund of Guangdong Province Key Laboratory of Precision Equipment and Manufacturing Technology (PEMT1303) for their support in this research.

References

1. Wille, R.: Restructuring lattice theory: an approach based on hierarchies of concept, ordered sets. In: Rival, I. (ed.), pp. 445–470 (1982)
2. Ganter, B., Wille, R.: Formal concept analysis: mathematical foundations. Springer, Berlin (1999)
3. Yang, Q., Zhao, M.: Progress in concept lattice research. Computer Engineering and Design **29**(20), 5293–5296 (2008)
4. Cai, Y., Cercone, N., Han, J.: An attribute-oriented approach for learning classification rules from relational databases. In: Proceedings of Sixth International Conference on Data Engineering, pp. 281–288 (1990)
5. Bordat, J.P.: Practical Calculation of Lattice Galois correspondence. Mathematiques et Sciences Humaines **96**, 31–47 (1986)
6. Ganter, B.: Two Basic Algorithms in Concept Analysis. In: Kwuida, L., Sertkaya, B. (eds.) ICFCA 2010. LNCS, vol. 5986, pp. 312–340. Springer, Heidelberg (2010)
7. Chein, M.: Algorithme de recherche des sous-matrices premieres d'une matrice. Bull. Math. Soc. Sci. Math. Roumanie, R.S. 13, 1–25 (1969)
8. Godin, R., Missaoui, R., Alaoui, H.: Incremental concept formation algorithms based on Galois (concept) lattices. Computational Intelligence **11**(2), 246–267 (1995)
9. Kuznetsov, S.O., Obiedkov, S.A.: Comparing performance of algorithms for generating concept lattices. Journal of Experimental & Theoretical Artificial Intelligence **14**(2–3), 189–216 (2002)
10. Xie, Z., Liu, Z.: A Fast Incremental Algorithm for Building Concept Lattice. Chinese Journal of Computers **25**(5), 490–496 (2002)
11. Jiang, Y., Zhang, J., Zhang, S.: Incremental construction of concept lattice based on linked list structure. Computer Engineering and Applications **43**(11), 178–180 (2007)
12. Mao, H., Xhang, Z.: Algorithm of generating concept lattice based on binary tree. Computer Engineering and Applications **45**(33), 35–37 (2009)
13. Wan, J., Zou, C., Ullah, S., Lai, C.F., Zhou, M., Wang, X.: Cloud-enabled wireless body area networks for pervasive healthcare. IEEE Network **27**(5), 56–61 (2013)
14. Zou, C., Deng, H., Qiu, Q.: Design and Implementation of Hybrid Cloud Computing Architecture Based on Cloud Bus. In: 2013 IEEE Ninth International Conference on Mobile Ad-hoc and Sensor Networks (MSN), Dalian, China, pp. 289–293 (2013)
15. Valtchev, P., Missaoui, R., Lebrun, P.: A partition-based approach towards constructing Galois (concept) lattices. Discrete Mathematics **256**(3), 801–829 (2002)
16. Li, Y., Liu, Z., Chen, L., Xu, X., Cheng, W.: Horizontal Union Algorithm of Multiple Concept Lattices. ACTA Electronica Sinica **32**(11), 1849–1854 (2004)
17. Zhang, L., Shen, X.-J., Han, D.-J., An, G.-W.: Vertical union algorithm of concept lattices based on synonymous concept. Computer Engineering and Applications **43**(2), 95–98 (2007)
18. Wang, W., Wu, Y.: Research on a Divide-and-conquer Algorithm for Constructing Concept Lattice. International Journal of Advancements in Computing Technology **4**(11), 96–105 (2012)
19. Maedche, A., Zacharias, V.: Clustering Ontology-Based Metadata in the Semantic Web. In: Elomaa, T., Mannila, H., Toivonen, H. (eds.) PKDD 2002. LNCS (LNAI), vol. 2431, p. 348. Springer, Heidelberg (2002)