A Batch Constructing Method of Weighted Concept Lattice Based on Deviance Analysis

Sulan Zhang1,2, Ping Guo1
1 School of Computer
Beijing Institute of Technology
Beijing 100081, China
zhulan@126.com, pguo@ieee.org

Jifu Zhang, Xinxin Wang2
2 School of Computer
Taiyuan Uni. of Science and Technology
Shanxi 030024, China
jifuzh@sina.com, xinxwang@126.com

Abstract

Concept lattice is an effective formal tool for data expression and knowledge acquisition. Weighted concept lattice (WCL) is a structure of concept lattice which depicts the importance of intents. A batch constructing method of weighted concept lattice based on deviance analysis is presented in this paper. The importance deviance value among the multi-attribute intent is computed by the standard deviation. According to the importance threshold and importance deviance threshold of the intent which are given by the user, strongly frequent weighted nodes and their edges are generated from bottom to top, so that the time and storage complexity of constructing weighted concept lattice is reduced, and the practicability and pertinence of concept lattice are improved. The experiment results validate the correctness and validity of the algorithm with the celestial spectrum data as the formal context.

1. Introduction

Concept lattice, proposed by R. Wille in 1982 [1], has become an effective formal tool for data expression and knowledge acquisition and successfully applied in the fields of data mining, literature retrieve, software engineering, natural language processing, and so on [2-4,6,7]. In the field of concept lattice, construction methods and algorithms of concept lattice are one of the main research problems. There are mainly two types of construction methods: batch and incremental algorithm. The two construction methods have their own advantages and disadvantages, and there exists distinct difference in the construction efficiency with the amount of dataset in the formal context. To incremental algorithms, concept lattice is dynamically built with the increasing of the object, and its construction efficiency is better than most batch algorithms under the condition of small number of data. But, its construction efficiency is lower when to deal with large quantity of data because of the comparisons between the generated nodes and the object which is newly added. The construction efficiency of batch algorithm is better than incremental algorithms while deal with large quantity of data [5].

Generally, it is always assumed that intents of concept lattice are the same important. Because of the completeness of the concept lattice’s structure, its complexity of construction is exponent. Furthermore, in data mining, a user is not satisfied with all the attributes in the data set and he is always extracting knowledge according to his preference for some attribute characteristics. Accordingly, the importance of attribute is different; namely the importance of intent is different. Therefore, it is significance to introduce weight value into the intent of concept lattice according to the interest degree of the user for extracting useful knowledge.

In [8], the weight value is introduced into the intent of concept lattice, and a new structure of concept lattice: weighted concept lattice, and its incremental construction method are presented. For the weighted concept lattice, the weight value of intent which is made up of single attribute is given by the experts according to his experience. The weight value of intent which is composed of multi-attribute is computed by the arithmetic mean. Weighted concept lattice expands the structure of general concept lattice and has practicability. Whereas, there exists some deficiencies as follows: the incremental construction efficiency of weighted concept lattice is low when to deal with large quantity of data; the importance of multi-attribute intent which is computed by the arithmetic mean, reflects the average degree of attributes’ importance, but the importance deviance among the multi-attribute is ignored. Thus, these are unfavourable to quickly extract some deviant knowledge which users are interested in from the large quantity of data.
Considering these deficiencies in [8], a batch constructing method of weighted concept lattice based on deviance analysis is presented in this paper. The weight value of single attribute intent is decided by the expert according to his experience or objectively computed by the information entropy. From the point of view of reflecting the average importance and importance deviation of the data, the weight value of multi-attribute intent is computed with arithmetic mean, and the importance deviation among the multi-attribute intent is computed with standard deviation. Strongly frequent weighted concept nodes are generated from bottom to top in the hierarchy order before generating the nodes at high layer. Those nodes at the current layer which are unsatisfied with the greatest expansibility, are expanded with objects according to the importance threshold of intent and the importance deviance threshold given by the user, so that avoid the building of redundant nodes. After the nodes at high layer have been generated, strongly frequent weighted nodes at the current layer are retained. Finally, the experiment results validate the correctness and validity of the algorithm by taking the celestial spectrum data as the formal contexts.

2. Weighted concept lattice

In [8], focusing on the different importance of attributes, the weight value \( w(0 \leq w \leq 1) \) is introduced into the intent of concept lattice to mark the different importance of intents and a new structure of concept lattice: weighted concept lattice, is presented.

**Definition 1** Let \( K_w=(G, M, I, W) \) be a formal context, where \( G \) be a set of objects, an attribute set \( M=\{m_1, m_2, \cdots, m_n\} \), and a set of weight value \( W=\{w_1, w_2, \cdots, w_n\} \). \( w_i \in W \) marks the important degree of single attribute \( m_i \) in \( M \) respectively, and \( 0 \leq w_i \leq 1 \). \( I \) is a binary relation between \( G \) and \( M \) and \( I \subseteq G \times M \). \( h_w=(A, B, w) \) is a triple in \( K_w \). \( A \subseteq G \), \( B \subseteq M \), \( w=\text{weight}(B) \), \( w \) is the weight value of attribute set \( B \), \( 0 \leq w \leq 1 \), and the following two mapping relations are both satisfied:

1) \( f(A)=\{m \in M \mid \forall n \in A, m|n\} \)
2) \( g(B)=\{n \in G \mid \forall m \in B, m|n\} \).

If \( f(A)=B \), \( g(B)=A \), i.e., the triple \((A, B, w)\) satisfies the greatest expansibility, then \( h_w=(A, B, w) \) is called weighted concept of \( K_w \). \( A \) is called extent of weighted concept of \( h_w \), \( B \) is called intent of weighted concept \( h_w \), and \( w \) is called weight value of intent \( B \).

**Definition 2** Let \( h_{w1}=(A_1, B_1, w_1) \), \( h_{w2}=(A_2, B_2, w_2) \) and \( h_{w3}=(A_3, B_3, w_3) \) be the weighted concepts of a formal context \( K_w \). If \( A_1 \subseteq A_2 \subseteq B_1 \subseteq B_2 \), then \( h_{w1} \) is called the descendant node of \( h_{w2} \), \( h_{w2} \) is the ancestor node of \( h_{w1} \), denoted by \( h_{w1} \leq h_{w2} \), and \( \leq \) is called the hierarchy order of concepts. In particular, if not exist \( h_n \), make \( h_{w1} \leq h_{w2} \leq h_{w3} \), then \( h_{w1} \) is called the child node of \( h_{w2} \), and we denote \( \text{child}(h_{w1}) \) as a set of all the child nodes of \( h_{w1} \). \( h_{w2} \) is called the parent node of \( h_{w1} \), and we denote \( \text{parent}(h_{w2}) \) as a set of all the parent nodes set of \( h_{w2} \). All the weighted concepts of \( K_w \) and all the hierarchy orders among the concepts construct the weighted concept lattice, and denoted by \( < L_w(G, M, I, W) > \).

3. Acquisition of the intent weight value and the importance deviance

Definition 3 let \( h_w=(A, B, w) \) be a weighted concept of a formal context \( K_w \) and \( B=m_1 \cap m_2 \cap \cdots \cap m_n \), if \( n=1 \), then \( B \) is called single attribute intent, or else \( B \) is called multi-attribute intent.

For a data set, in the structure of weighted concept lattice [8], the intent weight value of single attribute is decided by the expert according to his experience. Under the condition of lacking experience knowledge, the importance degree of single attribute intent can be objectively weighed by the information entropy. Information entropy, an index to measure the uncertainty of the information source and proposed by Shannon in 1948 [9], has gained successful application. Because each object in the formal context is independent, information of the same attribute provided by them can be added and the information of some attribute is the total of information provided by the objects, namely adopting information entropy to denote the uncertain degree of information source. According to the definition of entropy defined by Shannon, we take information entropy as the weight value of single attribute intent and denote the importance degree of single attribute intent with lacking of experience knowledge.

For an object \( g_i \) and \( g_i \in G(1 \leq i \leq n) \), \( P(m/g_i) \) is denoted as the probability of \( g_i \) possessing the attribute \( m \), \( H(m) \) is the average information of \( G \) providing the attribute \( m \), and is computed by formula (1) to denote the importance of attribute \( m \).

\[
H(m) = - \sum_{i=1}^{n} p(m/g_i) \log_2 \left( \frac{p(m/g_i)}{p(g_i)} \right) . \quad (1)
\]

Definition 4 let \( h_w=(A, B, w) \) be a weighted concept of a formal context \( K_w \) and an attribute set of \( M=\{m_1, m_2, \cdots, m_n\} \), if \( B=m_i (i=1, \ldots, n) \), then weight(B)=H(m_i)=w_i, \( w_i \) is the weight value of single attribute intent \( B \) under the condition of lacking
experience of the expert.

Let \( W \) be a weight value set of single attribute, if \( h_w=(A, B, w) \), and \( B=m_1 \cap m_2 \cap \cdots \cap m_n, m_1, m_2, \cdots, m_n \in M, \text{weight}(m_i)=w_i (i \in 1 \cdots n) \), then the weight value of multi-attribute intent \( B \) is computed by formula (2).

\[
\text{weight} (B) = \frac{w_a + w_2 + \cdots + w_n}{n} = \frac{\sum w_i}{n} (2)
\]

Actually, each node of concept lattice is a set of the largest item and it is benefit for the knowledge extraction. For each node of the weighted concept lattice, we can extract some important knowledge which the users are interested in if the intent of concept lattice is assigned the weight value. In [8], the weight value of multi-attribute intent is computed by the arithmetic mean and reflects the average importance degree of multi-attribute, but the importance deviation among the attributes is ignored, which are unfavourable to quickly and sensitively extract deviant knowledge according to the user’s preference. For one example, Selection of qualified personnel in university, we often evaluate a student from the comprehensive quality; for another example, when a doctor diagnose for the patient, he always investigate the impacts of symptoms with little deviance to guide the correct diagnosis, and accordingly obtain the better cure effects for the patient from the synthesis of treatment. Therefore, by making use of the importance deviation of multi-attribute intent to guide the acquisition of the multi-attribute intent weight according to the importance deviation threshold decided by the user, can effectively reflect interesting knowledge which the users are interested in.

Let \( W \) be a intent weight value set of single attribute, if \( h_w=(A, B, w) \) and \( B=m_1 \cap m_2 \cap \cdots \cap m_n, m_1, m_2, \cdots, m_n \in M, \text{weight}(m_i)=w_i (i \in 1 \cdots n) \), then the importance deviation of multi-attribute intent \( B \) is computed with the formula (3).

\[
\text{dev} (B) = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (w_i - \frac{w_a}{n})^2} (3)
\]

Apparently, dev(B)=0(n=1).

4. Batch construction of weighted concept lattice

Definition 5 Let \( \alpha (0 \leq \alpha \leq 1) \) be the importance threshold of intent which is given by the user according to his interest degree with the attribute. For a weighted concept \( h_w=(A, B, w) \) of \( <L_n(G, M, I, W), \leq> \), and \( w=\text{weight}(B) \), if \( w \geq \alpha \), then \( h_w \) is called frequent weighted concept (node). If any one weighted concept \( h_w \) of \( <L_n(G, M, I, W), \leq> \) is a frequent weighted concept, then \( <L_n(G, M, I, W), \leq> \) is called frequent weighted concept lattice.

Definition 6 Let \( \beta (0 \leq \beta \leq 1) \) be the importance deviance threshold of multi-attribute intent which is given by the user. For a frequent weighted concept \( h_w=(A, B, w) \) of \( <L_n(G, M, I, W), \leq> \), if \( \text{dev}(B) \leq \beta \), then \( h_w \) is called the strongly frequent weighted concept(node). If any one weighted concept \( h_w \) of weighted concept lattice is a strongly frequent weighted concept, then weighted concept lattice is called strongly weighted concept lattice.

Theorem 1 Let \( h_w=(A, B, w) \) be any one node of general weighted concept lattice \( <L_n(G, M, I, W), \leq> \) and \( w=\text{weight}(B). \) If \( w \geq \alpha \) and \( B \) is composed of single attribute, then \( <L_n(G, M, I, W), \leq> \) is a strongly weighted concept lattice.

Theorem 2 Let \( h_w=(A, B, w) \) be any one node of general weighted concept lattice \( <L_n(G, M, I, W), \leq> \) and \( w=\text{weight}(B). \) If \( w \geq \alpha \) and \( B \) is composed of multi-attribute; moreover, the weight value of all the attributes is equal, then \( <L_n(G, M, I, W), \leq> \) is a strongly weighted concept lattice.

As can be seen from the above theorems, strongly weighted concept lattice is the expansion of general weighted concept lattice. Strongly weighted concept lattice reduces the number of nodes; hence, the time and storage complexity of construction is lower, and the practicability and pertinence are also better than general concept lattice.

4.1. Methods and algorithms of construction

The main idea of strongly weighted concept lattice batching construction is building nodes with expansion of objects gradually from bottom to top in the hierarchy order: 1) at first, nodes are generated from the bottom layer. Those nodes at the \( k \) layer, which are unsatisfied with the greatest expansibility, are expanded with the objects before generating the nodes at the \( k+1 \) layer. Then, each two nodes at the \( k \) layer satisfying the greatest expansibility are combined to create the strongly frequent node at the \( k+1 \) layer; 2) after those nodes at the \( k+1 \) layer having been created, the nodes at the \( k \) layer are scanned, and those strongly frequent nodes are reserved; 3) all the edges between the nodes are built according to the relationship of the
node’s intent. To ensure the completeness of strongly weighted concept lattice, we define \( weight(\Phi)=1, \) \( dev(\Phi)=0, \) \( weight(B)=1 \) and \( dev(B)=0 \) \((B=M, g(B)=\Phi)\).

Based on the above relative theorems and ideas, a batch constructing algorithm of strongly weighted concept lattice is given as the follows:

Algorithm: BCSWCL (A Batch Constructing Algorithm of strongly Weighted Concept Lattice)

Input: formal context \( K \), intent importance threshold \( \alpha \), importance deviance threshold \( \beta \).

Output: strongly weighted concept lattice \( L_{sw} \)

1. \( k=1 \);
2. pick up data from database and initialize the nodes at the second layer;
3. While(.T.) Do create a temporary node set \( L_{temp} \);
4. \( k=k+1 \); expand nodes with objects;
5. For each node \( C \) at the \( k+1 \) layer Do
6. compute \( weight(C) \) and \( dev(C) \);
7. If \( (C \) is a strongly frequent node) Then \( L_{temp}=L_{temp}\cup C \);
8. End if
9. End For;
10. //according to \( \alpha \) and \( \beta \), judge whether \( C \) is a strongly frequent node
11. For each node \( C \) at the \( k \) layer Do
12. If \( (C \) is a strongly frequent node) Then \( L_{sw}=L_{sw}\cup C \);
13. End if
14. End For;
15. If (the total number of nodes is less than two)
16. Then create the nodes at the top layer;
17. Exit while;
18. End if
19. End while;
20. Build all the parent-child relation between the nodes according to the relations of the node intent;
21. End BCSWCL

4.2. An example

Table 1 is a formal context \( K_w=(G, M, I, W) \), \( G=\{1, 2, 3, 4, 5, 6\} \) and \( M=\{s, d, g, f, h\} \). The intent weight value set of single attribute \( W \) is computed by formula (1), then normalized, and get \( W=(0.11, 0.19, 0.21, 0.25, 0.24) \). The intent weight value of multi-attribute and importance deviance value are computed by formula (2) and (3). Suppose \( \alpha=0.15 \), and \( \beta=0.09 \), general weighted concept lattice and strongly weighted concept constructed from this formal context are as shown in Figure. 1 and Figure. 2.

<table>
<thead>
<tr>
<th>Table 1. A formal context</th>
</tr>
</thead>
<tbody>
<tr>
<td>s</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
</tbody>
</table>

![Figure 1. General weighted concept lattice](image1)

![Figure 2. Strongly weighted concept lattice](image2)

As can be seen from the example, \#8 and \#11 are not created in the strongly weighted concept lattice because their importance values and importance deviance values are not satisfied with the user’s preference. However, \#2 is retained to ensure the completeness of the lattice (the intent weight value of \#2 node is assigned -1, and we need not considering it when we extract knowledge which the users are interested in). Obviously, the strongly weighted
concept lattice reduces the number of nodes being built according to the user’s preference; moreover, intents of nodes are composed of the attributes which are important with deviance concerned with the user, and the structure of the weighted concept lattice is simple and practicability.

5. Experimental results and analysis

The experiment environment setup is Pentium(R) D-3.0G CPU, 512M memory, Windows XP operating system and ORACLE 9i DBMS. The BCSWCL algorithm is coded in Microsoft Visual Studio 2005 C++. 2000 pieces of data are selected from the celestial spectra data, which are provided by the National Observatory in Beijing, China. Experimental results are shown in table 2 and table 3. Table 2 shows comparison results at the different deviance thresholds $\beta$ with the same importance threshold $\alpha=0.01$. Table 3 shows comparison results at different important threshold $\alpha$ with the same importance deviance thresholds $\beta=0.002$.

### Table 2. Comparisons of different $\beta$ with the same $\alpha$

<table>
<thead>
<tr>
<th>Importance deviance threshold $\beta$</th>
<th>Time of building concept lattice(s)</th>
<th>Number of nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>219</td>
<td>27</td>
</tr>
<tr>
<td>0.002</td>
<td>223</td>
<td>1127</td>
</tr>
<tr>
<td>0.003</td>
<td>225</td>
<td>3126</td>
</tr>
<tr>
<td>0.004</td>
<td>227</td>
<td>6080</td>
</tr>
<tr>
<td>0.005</td>
<td>235</td>
<td>9668</td>
</tr>
<tr>
<td>1</td>
<td>243</td>
<td>12236</td>
</tr>
</tbody>
</table>

### Table 3. Comparisons of different $\alpha$ with the same $\beta$

<table>
<thead>
<tr>
<th>Importance threshold $\alpha$</th>
<th>Time of building concept lattice(s)</th>
<th>Number of nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>250</td>
<td>1651</td>
</tr>
<tr>
<td>0.005</td>
<td>246</td>
<td>1262</td>
</tr>
<tr>
<td>0.01</td>
<td>223</td>
<td>1127</td>
</tr>
<tr>
<td>0.013</td>
<td>220</td>
<td>913</td>
</tr>
<tr>
<td>0.015</td>
<td>124</td>
<td>377</td>
</tr>
</tbody>
</table>

As can be seen from table 2, with the same $\alpha$, the smaller $\beta$, there are the less nodes and time of building lattice. when $\beta$ is 1, the structure of the weighted concept lattice is a kind of frequent weighted concept lattice. We also can see from table 3, with the same $\beta$, the larger $\alpha$, there are the less nodes and time of construction. Therefore, strongly weighted concept lattice can effectively reduce the nodes of general concept lattice and improve efficiency of extracting important knowledge which the users are interested in.

6. Conclusions

In view of extracting some important deviant knowledge which the users are interested in from large quantity of data, a batch constructing method of weighted concept lattice based on deviance analysis is presented in the paper. The construction efficiency and practicality of the weighted concept lattice have been furtherly improved. The future work is knowledge extraction based on the weighted concept lattice.

Acknowledgment

The work described in this paper is supported by the grants from the National Natural Science Foundation of China (Project Nos. 60675011, 60773014). Prof. P. Guo is the author to whom the correspondence should be addressed.

References