A Fast Algorithm for Attribute Reduction of Formal Concept Analysis

Ensheng Zhang
Anshan Normal University
Anshan, 114000
Liaoning, China
Zhangensheng@126.com

ABSTRACT: *Formal concept analysis is an effective tools for knowledge discovery, information retrieval, machine learning, software engineering, etc.., The attribute reduction of concept lattices can reduce the complexity of concept lattices. Concept lattice is the central notion of a formal concept analysis, a new area of research which is based on a set-theoretical model of concepts and conceptual hierarchies. This model yields not only a new approach to data analysis, but also methods for formal representation of conceptual knowledge. Several algorithms were proposed for the attribute reduction of concept lattices, such as the discernibility attribute matrix method, requiring all the formal concepts in concept lattices to be solved, which is also a difficult job. This paper proposes a fast algorithm to work out reduction attribute directly on object set, which reduces the complexity and the calculation of concept lattice structuring.*

Keywords: Formal Concept, Concept Lattices, Attribute Reduction

Received: 8 September 2016, Revised 9 October 2016, Accepted 15 October 2016

1. Introduction

Formal concept analysis, which has been developed during the last ten years and shall be explained in this paper, is supposed to achieve the aims as they are formulated in the German standards on concepts and conceptual systems; these standards are seen as a general aid in the sciences, economy and administration for a better understanding and use of "conceptual tools." The standards are based on the philosophical understanding of a concept as a unit of thoughts consisting of two parts: the extension and the intension (comprehension); the extension covers all objects (or entities) belonging to the concept while the intension comprises all attributes (or properties) valid for all those objects [1-3]. A set-theoretic model for these relationships is the root of formal concept analysis. This model yields not only a new approach to data analysis, but also methods for formal representation of conceptual knowledge. The research on concept lattice is mostly focused on the theory and methods of solution, establishment, decomposition and composition. For a deeper understanding of the structure of concept lattices, researchers begin to study on reduction theory and method [4-5].

Reduction of concept lattices provides a means to research concept lattice by defining reductiveness and reductible object.

Reduction of a concept lattice also makes tacit knowledge easily to be discovered and expressed, representing a new method to establish a concept lattice, enriching the theory of the concept lattice, which is significant to both research and application of the theory [6-7]. The methodology of attribute reduction is to find a minimum attribute subset that can exactly define the concept and structure of the original formal context. The concept lattice of formal context of reduced subset is isomorphic with that of the original, hence the reduced concept lattice can be used to work out the original lattice, which actually reduces the complexity of a concept lattice of original formal context. Presently discerned attribute matrix is adopted to solve the reduced attribute set [8], as the solution to discerned matrix must be based on all the concepts, its time complexity is pretty great. This paper, by looking into reduced collective attributes characters, provides a reduced set solution based directly on the original formal context [9-10]. This proposed algorithm is highly efficient, with time complexity 0(mn), where m represents the number of attributes, *n* represents the number of objects.

2. Definitions and Properties of Concept Lattice and Its Attribute Reduction

Concept analysis permits grouping objects that have common attributes. The starting point of concept analysis is a context (O, A, R), consisting of a set of objects O, a set of attributes A, and a binary relation R between objects and attributes, stating which attributes are possessed by each object [11-12]. A concept is a maximal collection of objects that possess common attributes, i.e., it is a grouping of all the objects that share a common set of attributes. More formally, a concept is a pair of sets (X, Y), X is said to be the extent of the concept and Y is said to be the intent (See figure 1).

Figure 1. Concept analysis permits

In the remainder of the paper, concept analysis will be instantiated as follows: Objects are program variables, while attributes are program statements. The binary relation between objects and attributes contains a pair (variable, statement) if statement belongs to the decomposition slice of a variable. The resulting concepts contain sets of variables (extent) which share some computation (statements in the intent). Both sets are maximal: If a variable has all the statements with the intent of a concept in its decomposition slice, then it is assured that such a variable is in the extent of that concept [13-14]. Conversely, if a statement belongs to all the decomposition slices of the variables in a concept extent, it must necessarily be included in the concept intent (See figure 2).

All definitions of concept in formal concept analysis are based on a formal context.

Definition 1 We define (U,A,I) as a formal context, of which U={x1,...,xn} is the object set, every xi ($i \leq n$) in this set is called an object; A= {a1,...,am } is the attribute set, every aj ($j \leq m$) in this set is called an attribute; I is the binary relation between U and A, $I \subseteq U \times A$. If $(x,a) \in I$ is true, then we define a as an attribute of x, expressed as xIa.

Definition 2 Assume (U,A,I) is a formal context, if there is a binary set (X,B) meets X*= B, and B*=X, then we define (X,B) a formal concept, or “concept” for short. X is called the extension of the concept, and B is called the intension of the concept.

Figure 2. Typical concept lattice

Theorem 1 There is at least, but not limited to, one reduction for any formal context (U,A,I)'

Theorem 2 For formal context (U,A,I), if D $\subset$ A, D $\neq \phi$, E=A-D; then "D is a consistent set" $\Leftrightarrow$ $\forall$e $\in$ E, $\exists$C$\subseteq$D, C $\neq \phi$ such that C*=e*.

Theorem 3 For formal context (U,A,I), if $\forall$a $\in$ A, expressed as Ga={g|g $\in$ A,g*$\supset$a*}. Then the following conclusions are true: "a" is a core attribute $\Leftrightarrow$ (a**-{a})*$\neq$a* . "a" is an absolutely dispensable attribute $\Leftrightarrow$ (a**-{a})*=a*, and Ga*=a* . "a" is a relatively required attribute $\Leftrightarrow$ (a**-{a})*=a*, and Ga*$\neq$a* (See figure 3).

	S_1	S_2	S_3	S_4
c	√			
d		√		
f	√	√	√	
g	√	√		√

$S_1 = \{3, 9, 10\}$
$S_2 = \{4, 11, 12\}$
$S_3 = \{1, 5, 6, 13\}$
$S_4 = \{2, 7, 8, 14\}$

$c_1 = (\mathrm{V}, \emptyset)$
$c_2 = (\{c, f, g\}, S_1), Int[c_2] = S(c)$
$c_3 = (\{d, f, g\}, S_2), Int[c_3] = S(d)$
$c_4 = (\{f, g\}, S_1 \cup S_2)$
$c_5 = (\{f\}, S_1 \cup S_2 \cup S_3), Int[c_5] = S(f)$
$c_6 = (\{g\}, S_1 \cup S_2 \cup S_4), Int[c_6] = S(g)$
$c_7 = (\emptyset, \mathrm{P})$

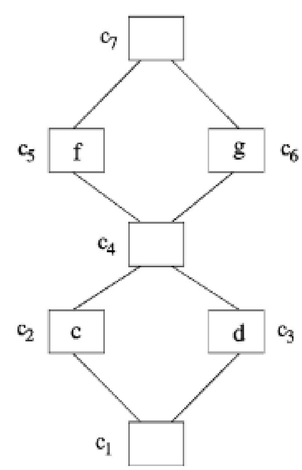

Figure 3. The example for concept lattice. Context and concepts are shown on the left

3. Structure of the Reduced Set

Definition 3 For B$\subset$A, x $\in$ U, if x contains every attributes in attribute set B, we say x contains attribute B, expressed as xIB.

Definition 4 For formal context (U,A,I), let $a,b \in U$, if $a^*=b^*$, then we call a, b equivalent to each other.

Theorem 4 For formal context (U,A,I) ,if $a,b \in A$, and $a^*=b^*$, then the following conclusions are true: For $x \in U$, $xIa \Rightarrow xIb$, $xI\{a,b\}.(a^{**}-\{a\})^*= (a^{**}-\{b\})^* = (b^{**}-\{b\})^* =(b^{**}-\{a\})^*$.

Theorem 5 For formal context (U,A,I), if exists $a,b \in A$, makes $a^*=b^*$, then the following conclusions are true: If a is relatively required attribute, b is also relatively required.If a is absolutely dispensable, the same is b.

Theorem 6 For formal context (U,A,I), if $a,b \in A$, a is a relatively required attribute and $a^*=b^*$, then a,b cannot be contained in the same reduced set.

Theorem 7 For formal context (U,A,I),D is the equivalent set of its relatively required attributes and $D \neq \phi$. If B is a reduced set of A, $\forall [a] \in D, B \cap [a] \neq \phi$.

Theorem 8 For formal context (U,A,I), $D=\{[a1], [a2],\ldots, [at]\}$ is its equivalent set of relatively required attributes and $D \neq \phi$. If K is the core set of A,$\forall ai' \in [ai]$, $i=1,2,\ldots,t$, we have $K \cup \{a1',a2',\ldots,at'\}$ is a reduced set of A and all the other reduced set can be expressed in the same form (See figure 4).

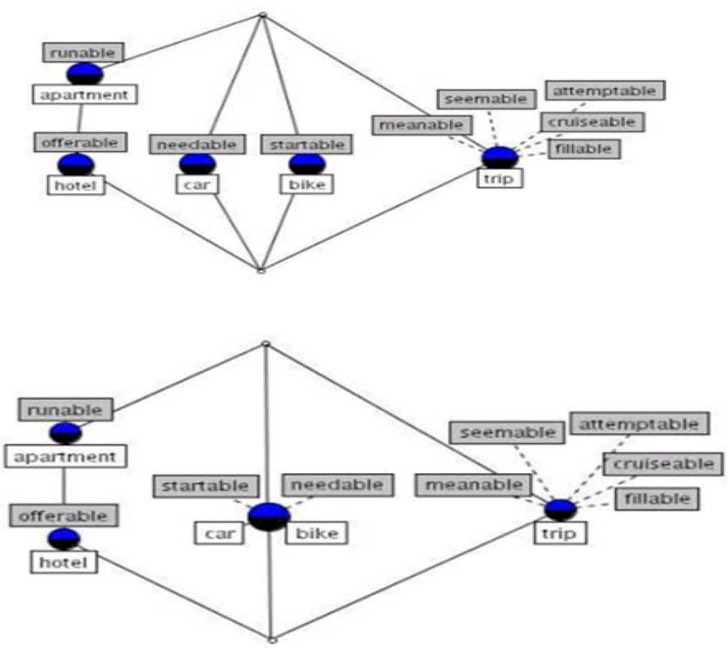

Figure 4. The comparison between structure of the reduced set before and after

4. Discrimination Theorem and Solution for Attribute Charaters

4.1 Discrimination Theorem for Attribute Characters

Now we present the discrimination theorem of attribute character.

Theorem 9 For formal context(U,A,I), $a \subseteq A$, if $a^{**}-\{a\} \neq \phi$, and $a^{**}-\{a\}=G_a \cup E_a$, among which $G_a=\{g|g \in A, g^* \supset a^*\}$, then E_a is the attribute set of all equivalents of a.

Theorem 10 (Discrimination for absolutely required attributes) For formal context (U,A,I), $a \subseteq A$, a is absolutely required attribute $\Leftrightarrow a^{**}=\{a\}$or $\exists u \subseteq U$, u is free of a, $uI(a^{**}-\{a\})$.

Theorem 11 (Discrimination for relatively required and absolutely dispensable) For formal context (U,A,I), $a \subseteq A$, $(a^{**}-\{a\})^*=a^*$.If

$G_a \neq \phi$, then a is relatively required. If $G_a \neq \phi$, and $\exists u \in U$, u does not contain a, and uIG_a, then a is relatively required, or else a is absolutely dispensable.

4.2 Solution to Attribute Characters in Object Set

Hereinafter presents the detailed method to solve core attribute, relatively required attribute and absolutely dispensable attribute, as well as the verification of the method [15-18]. The philosophy of this method is using n-tuple vector to represent the objects in the set of formal context (U,A,I). If A={a1,a2,…,am}, |U|=n, $\forall$ ui$\subseteq$U, ui=(ui1,ui2,…,uim), of which uij=1 $\Leftrightarrow$ uiIaj, or else uij=0. We use "attribute(a)" to represent the set of all attributes contained in u.

The following is algorithm description in C++ syntax.

Algorithm
Check(U,a_j,&M)

Function: categorize a_j by core attribute, relatively required attribute, or absolutely dispensable attribute, and use M to represent the equivalent attribute set to a_j.

Input: Object set U, attribute a_j, a_j equivalent attribute set.

Output: If a_j is a core attribute, then returns 0; or else a_j is relatively required, returns 1; or else a_j is absolutely dispensable returns 2.

```
{
e=(1,1,1,…,1)
 for(i=1; i<=n; i++)
{   fetch u_i from U;
    if(u_ij==1)e=e&u_i;
 }  //herein attribute(e)==a_j**.
 if(attribute(e)=={a_j}) return(0); //herein(a_j**={a_j}
 else
 {   assign the No.j component of e to zero;
//the nonzero components in e represent attribute set a_j**-{a_j}
   t=(0,0,…,0); flag=0;
   for(i=1; i<=n; i++)
    {   fetch u_i from U;
        if(u_ij==0)  //check the object without attribute a_j
       {   if(e == e&u_i)return(0); //herein u_iI (a_j**-{a_j})
           else if( t != (e&u_i)|t )
               {  t=(e&u_i)|t;
     if(t==t&u_i) flag=1; else  flag=0;
               }
             else  if(t==t&u_i) flag=1;
       }
   }
       M= attribute(e^t);   //M is the set of all attributes equivalent to aj
      if(flag==1) return(1);
       else  return(2)
}
}
```

5. Conclusions

In this paper, we firstly discuss the characters of equivalent attributes and conclude that an equivalent to relatively require attribute is also a relatively required attribute (RRA); therefore the RRAs of a formal context can be divided into different equivalent groups. Then we discuss the relations between RRAs and reduced set, and we conclude that two equivalent RRA cannot sit in the same reduced set, if a formal context has at least one RRA, then each of the intersections of attribute reduced set with every RRA equivalent set is not null, that is Theorem 8, which describes the component and structure of a reduced set. Finally, we present the theorem and an algorithm on attribute discrimination by object set, tell how to know the type of an attribute and work out the equivalent set of RRA and absolutely dispensable attribute so to work out the reduced set of all the attributes. For future study, we can research on the quick solution to solve reduced concept lattice by reduced sets, and acquire the original concept lattice from an isomorphic reduced concept lattice.

References

[1] Ganter, B., Stumme, G., Wille, R. (2005). Formal concept analysis : foundations and applications. Lecture Notes in Artificial Intelligence, no. 3626, Springer-Verlag. 21 (8) 926

[2] Valtchev, Petko., Missaoui, Rokia., Godin, Robert. (2004). Formal Concept Analysis for Knowledge Discovery and Data Mining. International Conference on Formal Concept Analysis, Springer, Berlin. 8 353-371

[3] Kuaznetsov, Sergei O.. (2004). Machine Learning and Formal Concept Analysis. International Conference on Formal Concept Analysis, Springer, Berlin. 6 288-311

[4] Martin, Ben (2004). Formal Concept Analysis and Semantic File Systems. International Conference on Formal Concept Analysis, Springer, Berlin. 3. 88-95.

[5] Priss, U. (2006). Formal concept analysis in information science. Arist. 40 (1) 521-543.

[6] Wolff, K E. (1993). A first course in formal concept analysis. *StatSoft*. 93 429-438

[7] Medina, J., Ojeda-Aciego, M., Ruiz-Calvino, J. (2009). Formal concept analysis via multi-adjoint concept lattices. *Fuzzy Sets and Systems* 160(2) 130-144

[8] Lai, H., Zhang, D. (2009). Concept lattices of fuzzy contexts: Formal concept analysis vs. rough set theory. *International Journal of Approximate Reasoning*. 50 (5) 695-707.

[9] Formica, A. (2006). Ontology-based concept similarity in formal concept analysis. *Information Sciences.* 176 (18) 2624-2641.

[10] Kuznetsov, S . O. (2007). On stability of a formal concept. *Annals of Mathematics and Artificial Intelligence*. 49 (1-4) 101-115.

[11] du Boucher-Ryan P, Bridge D. (2006). Collaborative recommending using formal concept analysis. *Knowledge-Based Systems*. 19 (5) 309-315.

[12] Formica, A. (2008). Concept similarity in formal concept analysis: An information content approach. *Knowledge-Based Systems* 21 (1) 80-87.

[13] Zhang W, Wei L, Qi J. (2005). Attribute reduction theory and approach to concept lattice. Science in China Series F: Information Sciences. 48 (6) 713-726.

[14] Wang X, Zhang W. (2008). Relations of attribute reduction between object and property oriented concept lattices. *Knowledge-Based Systems*. 21(5) 398-403.

[15] Medina J S. (2012). Relating attribute reduction in formal, object-oriented and property-oriented concept lattices. *Computers & Mathematics with Applications*. 64 (6) 1992-2002.

[16] Wei L, Qi J J. (2010). Relation between concept lattice reduction and rough set reduction. *Knowledge-Based Systems* 23 (8) 934-938.

[17] Jie Sun, Lihong, Li, Yan, Li, Baoxiang Liu. (2015). Set Pair Analysis of Lattice Order Decision-making Model and Application. *Journal of Information Technology Review*, 6 (4) 109-116.

[18] Boris Moltchanov, Oscar Rodriguez Rocha. (2015). A Concept Recommender Service on The Federated Cloud of the PPP European Project. *Journal of Intelligent Computin*g, 6 (1) 9-16.