A New Lattice Structure and Method for Extracting Association rules Based on Concept Lattice

JiYe Liang, JunHong Wang

Key Laboratory of Ministry of Education for Computation Intelligence and Chinese Information Processing, School of Computer and Information Technology, Shanxi University, Taiyuan 030006, China

Abstract

The problem of the relevance and the usefulness of extracting association rules are of primary importance because, in the majority of cases, the traditional association rule mining framework produce many redundant rules. In this paper, a new framework and method for extracting association rules based on concept lattice and the concept of closed itemsets is proposed. The number of non-redundant rules produced by the new approach is smaller than the rule set from the traditional approach. Experiments confirm the utility of the framework in terms of reduction in the number of rules, and in terms of time.

Key words:
Concept lattice, closed itemsets, frequent itemsets, association rules

1. Introduction

Association rule discovery, a successful and important mining task, aims at uncovering all frequent patterns among transactions composed of data attributes or items. Results are presented in the form of rules between different sets of items, along with metrics like the joint and conditional probabilities of the antecedent and consequent, to judge a rule's importance. Since the problem of mining association rules was originally introduced by R. Agrawal [1], it has been studied widely and deeply in [2-7].

Concept lattice was proposed by R. Wille [8] in 1982, which is a powerful tool for data mining and rule extracting. Concept lattice essentially represents the association between objects and attributes, and reflects the relationship of generalization and specialization among concepts, and Hasse graph realizes the visualization of data. It has been widely used in software engineering, knowledge engineering, data mining and information retrieval and so on [9-15]. Concept lattice is a natural tool for mining association rules. R. Godin [16] proposed an algorithm of mining implication rules, N. Pasquier and Y. Bastide [17, 18] discussed extracting rules used closed itemsets, and in [19] they addressed mining minimal non-redundant association rules using frequent closed itemsets, M. Zaki [20] presented an approach of generating non-redundant association rules based on the concept of closed frequent itemsets, the results shown that, by doing so, the total number of itemsets and rules can be reduced substantially, especially in dense data sets.

Based on above study, the paper address a new framework for extracting association rules based on concept lattice and the concept of closed itemsets. The paper is organized as follows. Section 2 reviews the basic concepts. In section 3, we present a new lattice structure and incremental building algorithm. Section 4 gives an algorithm of extracting implication rules and section 5 gives a method of extracting frequent itemsets and association rule based on the new framework. In section 6, we further explain the idea and its realizing process by an example. Finally, we summarize our work and conclude in section 7.

2. Basic Concepts

2.1 Concept Lattice

Definition 2.1[8] A formal context is a triple $K=(U,D,R)$, where U is a finite set of elements called objects, D is a finite set of elements called attributes and R is a binary relation between U and D. For arbitrary $x \in U$ and $y \in D$, $(x, y) \in R$ (also denoted by xRy) if the object x has the attribute y.

Now we define two mappings $f: P(U) \to P(D)$ and $g: P(D) \to P(U)$ as follows [8]:

$f(X)=\{y \in D \mid \forall x \in X, xRy\}$, *for* $X \in P(U)$,

$g(Y)=\{x \in U \mid \forall y \in Y, xRy\}$, *for* $Y \in P(D)$.

Where $P(U)$ and $P(D)$ are the power sets of U and D respectively. Then, the maps f and g form a Galois connection between power sets $P(U)$ and $P(D)$.

Definition 2.2[8] Let $K=(U,D,R)$ be a formal context, $X \in P(U)$ and $Y \in P(D)$. (X,Y) is called a concept, if $f(X)=Y$ and $g(Y)=X$ hold. X and Y are called the extent and intent of the concept (X,Y) respectively. By $L(K)$, we denote the set of all concepts in the formal context K.

Definition 2.3[8] Let $K=(U,D,R)$ be a formal context. For $H_1=(X_1,Y_1)$ and $H_2=(X_2,Y_2)$ in $L(K)$, $H_1 \le H_2$ if and only if $Y_2 \subseteq Y_1$. H_1 is called child node of H_2, and H_2 is called father node of H_1.

It is clear that the relation $\le$ is a partial order on $L(K)$ and can induce a lattice $(L(K),\le)$ called the concept lattice of the formal context K, also abbreviated as $L(K)$.

2.2 Closed Itemsets

In the following, the basic concept of closed itemsets is introduced. The detail description of closed itemsets can be seen in [17, 18].

Definition 2.4 [18] Let $K=(U,D,R)$ be a context, (f,g) a Galois connection between power sets $P(U)$ and $P(D)$. The operations $h=f\circ g=f(g(Y))$ $(Y\in P(D))$ and $h'=g\circ f=g(f(X))$ are called Galois closure operator.

Definition 2.5[18] Let $K=(U,D,R)$ be a context, and $C\subseteq D$, then C is a closed itemset if and only if $h(C)=C$.

That is to say, closed itemset is the largest set of attributes that have common objects. From the Definition 2.5, we have the following corollaries:

Corollary 2.1 Intent of the concept (X,Y) in $L(K)$ is a closed itemset.

Corollary 2.2 The set of closed itemset generated from $K=(U,D,R)$ is $R=\{Y \mid (X,Y)\in L(K)\}$.

3. Closed Label Lattice and Incremental Building Algorithm

In this section, a new lattice structure for extracting rules, called closed label lattice is proposed.

Definition 3.1 Let $C=(X,Y)\in L(K)$, $extention(C)=X$ be extent of C, and $intention(C)=Y$ be intent of C. The closed label of C is denoted by $Closedlabel(C)$, where $Closedlabel(C)$ is a set, $\forall X\in Closedlabel(C)$ satisfy the follow conditions:

(1) $h(X)=intention(C)$;

(2) $h(X)\ne X$;

(3) $\forall Y\subset X$, $h(Y)\subset h(X)$.

Form Definition 3.1, we know that the elements of closed label set can represent the concept and they have simpler form that possesses the information of the concept. So we can use closed label of a concept to denote the original concept.

Theorem 3.1 Let $C=(X,Y)\in L(K)$, if $C'=\{C_1 \mid C\le C_1\}=\{C_1'\}$, then the closed label of C is

$Closedlabel(C)=\bigcup\{x \mid x\in intention(C)-intention(C_1')\}$.

Proof For $\forall x\in Closedlabel(C)$, we have that, (1) $h(x)=intention(C)$; (2) $h(x)=intention(C)\supseteq x$; (3) is obviously true. So x is a closed label.

Theorem 3.2 Let $C=(X,Y)\in L(K)$, $C'=\{C_1 \mid C\le C_1\}=\{C_1',C_2'\}$.

(1) If $intention(C)-(intention(C_1')\cup intention(C_2'))\ne\varnothing$,

$Closedlabel(C)=\{x \mid x\in\{intention(C)-(intention(C_1')\cup intention(C_2'))\}$.

(2) If $intention(C)-(intention(C_1')\cup intention(C_2'))=\varnothing$,

$Closedlabel(C)=\{x_1x_2 \mid x_1\in intention(C)-intention(C_1'), x_2\in intention(C)-(intention(C_2'))\}$.

Proof Theorem 3.2 is obviously true.

For some concept C, $C'=|\{C_1 \mid C\le C_1\}|>2$, then its closed label can correspondingly be generated based on theorem 3.2.

We can use algorithm 3.1 to produce the closed label set of concept in the lattice.

Algorithm 3.1 Compute closed label $Computeclosed(N)$ of concept N.

Input: $parents(N)$, $parents(N)=\{N' \mid N\le N'\}$.

Output: Closed label $closedlabel(N)$ of N.

Step 1: If $extent(N)=\varnothing$ or $intention(N)=\varnothing$, $Closedlabel(N):=N$, skip to step 6.

Step 2: Computing $L=\bigcup\{intention(N') \mid N'\in parents(N)\}$.

Step 3: If $L\subset intention(N)$, i.e. there have new added attributes, $Closedlabel(N):=\{x \mid x\in intention(N)-L\}$, skip to step 6.

Step 4: If $L=intention(N)$, i.e. there have no new added attributes, for any two element M_i and M_j in $parents(N)$.

Computing $L_i=intention(N)-intention(M_i)$,

$L_j:=intention(N)-intention(M_j)$,

$Closedlabel(N):=Closedlabel(N)\cup\{x_ix_j \mid x_i\in L_i, x_j\in L_j \text{ and } x_ix_j\ne intention(M_i), x_ix_j\ne intention(M_j)\}$.

Step 5: If $Closedlabel(N)=\varnothing$, then get any three element from $parents(N)$, computing as step4, until $Closedlabel(N)\ne\varnothing$.

Step 6: End, return $Closedlabel(N)$.

Based on the concept of closed label, we can define a new framework of concept lattice.

Definition 3.2 Every concept of closed label lattice is a triple $(extention(C),\ Closedlabel(C),\ intention(C))$, called concept of closed label lattice.

There have been a series of algorithms for building concept lattice based on binary relation. These algorithms can be parted into two species [21]: batch algorithm and incremental algorithm. R. Godin [22] proposed an algorithm of incrementally building lattices and its Hasse graph, and Z. Xie [23] gave an algorithm for building association rules lattice. Algorithms 3.2 has similar idea as above algorithms, but does some corresponding modification as to the difference concept structure.

Algorithm 3.2 Incremental algorithm of building lattice based on closed label.

Input: Given lattice L, added an object x, $f(x)$ denote the set of attributes that x satisfied, i.e. add concept （x, Φ, $f(x)$） into L.

Output: Lattice L' after update.

Step 1 : Initialization $Mark := \phi$.

Step 2: For every concept C in lattice, rearrange as $Card(intention(C))$ ascending.

Step 3: For C, if $intention(C) \subset f(x)$, add x to $extention(C)$, add C to $Mark$, skip to step 7. If $intention(C) = f(x)$, add x to $extention(C)$, add C to $Mark$, skip to step 8.

Step 4: Let $Int = intention(C) \cap f(x)$, if there dose not exists a father node $C_i \in Mark$ such that $intention(C_i) = Int$, then create new node N, intention(N):= Int; $extention(N) := extention(C) \cup x$, add N to the set of his father nodes $parents(C)$, and add C to the set of his child nodes $Children(N)$, otherwise skip to step 7.

Step 5: Get a element MP from $Mark$.

(1) If $intention(MP) \subset intention(N)$, then $Parent := true$;

(2) For every $M \in Children(MP)$, if $intention(M) \subset intention(N)$, then $Parent := false$;

(3) If $Parent = true$, then add N to $Children(MP)$, and add MP to $parents(N)$. If $MP \in parents(C)$, then delete MP from $parents(C)$, and delete C from $Children(MP)$. Repeat step 5.

Step 6: Run, $Computeclosed(N)$ compute closed label of new node N, and re-compute closed label of every child node of N.

Step 7: Get next node, skip to step 3.

Step 8: End, output lattice L'.

4. Mining Implication Rules

One can directly mining implication rules from concept lattice according to its relationship of generalization and specialization among concepts, but the number of rules is very large [16, 24]. Here we extract generation set of implication rules using closed label lattice. Generation set needs small storage space, and other implication rules can be deduced from it. Sometimes what user interest is only a part of rules in the whole set of rules, then he can extract some rules selectively, and generate other rules. But the generation set extracted is also not the smallest rules set, rules generate from different concept may have redundancy.

Theorem 4.1 Let X_1, X_2 be closed label of a concept lattice, then rule $X_1 \to X_2$ is true if and only if $h(X_1) \to h(X_2)$.

Proof According to definition of closed itemsets, X_1 and $h(X_1)$ have same objects set, X_2 and $h(X_2)$ have same objects set. So the theorem is apparently true.

Meanwhile, for any closed label X in concept lattice, $X \to h(X)$, $h(X) \to X$ are true.

In closed label lattice, the closed itemsets of closed label is intent of the concept, and closed label set is the simplest attributes set that can denote the concept. So we only consider closed label when mining generation set.

Theorem 4.2 Let X_1, X_2, X_3 be closed label of a concept lattice, and $h(X_1) \subseteq h(X_2) \subseteq h(X_3)$. If $X_1 \to X_2$, $X_2 \to X_3$, we have $X_1 \to X_3$.

Proof According to the relationship of generalization and specialization among concepts, one easily knows that the theorem is true.

Theorem 4.2 shows that mining implication rules only need to think about neighborhood concept.

Theorem 4.3 If the closed label of a concept includes more than one new added attributes, these attributes implicate each other.

Proof Because these attributes appear in the same concept at the same time, they must have same objects set. Hence, they implicate each other.

Let a, b be two new added attributes, then $a \to b$, $b \to a$ are two implication rules.

Form Theorem 4.3, we have that:

Corollary 4.1 Let $C=(X,Y)\in L(K)$, $C'=\{C_1 \mid C \le C_1\}=\{C_1'\}$, for $\forall x \in Closedlable(C)$, $\forall y \in Closedlable(C')$, $x \to y$ is true.

Theorem 4.4 Let $C=(X,Y)\in L(K)$, $C'=\{C_1 \mid C\le C_1\}=\{C_1',C_2'\}$, for $\forall X \in Closedlable(C)$, $\forall Y \in Closedlable(C_1')$ or $Y \in Closedlable(C_2')$, if $Y \not\subset X$, we have that $X \to Y$ is an implication rule.

Proof If $Y \not\subset X$, because every concept related to its father node is specialized, $X \to Y$ is true. If $Y \subseteq X$, $X \to Y$ is clearly true and hence redundancy.

Theorem 4.5 Based on generation set, we can deduce all other implication rules.

Proof Because generation set of implication rules is produced from closed label (not intent of concept), and closed label can denote the concept. Therefore, all other rules can produced based on theorem 4.1-4.5.

User can use the following methods to generate implication rules from generation set.

(1) Adding some attributes to the former of the implication rules (added attributes should be in the same closure as the former).

(2) Transfer according to theorem 4.2.

(3) Unite some rules together, for example, if $W \to C$, $D \to C$, then $DW \to C$ [20].

Algorithm 4.3 Mining generation set of implication rules based on closed label lattice.

Input: Given closed label lattice L.

Output: Generation set of implication rules Σ.

Step 1 : Initialization $\Sigma := \varnothing$.

Step 2 : For every concept C in lattice, rearrange as $Card$ (intention(C)) ascending, and put into $Mark$, let $\mathrm{Mark}' := Mark$.

Step 3: Get a concept C from Mark', $\mathrm{Mark}' := \mathrm{Mark}' - C$. If $Closedlabel(C) = \varnothing$, skip to step 10.

Step 4: Define a set $Parents'(C)$, and let $Parents'(C) := Parents(C)$.

Step 5: Get a concept N from $Parents'(C)$, let $Parents'(C) := Parents'(C) - N$. If $Closedlabel(N) = \varnothing$, skip to step 8.

Step 6: Let $S := Closedlabel(C)$, if $Closedlabel(N)$ has two or more new added attributes, then move these attributes from S.

Step 7: For every element X in S and every element Y in $Closedlabel(N)$, generate rule $X \to \{Y - X\}$, $\Sigma := \Sigma \bigcup \{X \to \{Y - X\}\}$. If S has two or more new added attributes, then for arbitrary two new added attributes, generate rules $a \to b$, $b \to a$, $\Sigma := \Sigma \bigcup \{ a \to b,\ b \to a \}$.

Step 8: If $Parents'(C) \neq \varnothing$, skip to step 5.

Step 9: If there have some element X in S that have not generated rules, then $\Sigma := \Sigma \bigcup \{X \to \{\text{intention}(C) - X\}\}$.

Step 10: If $\mathrm{Mark}' \neq \varnothing$, skip to step 3.

Step 11: End, output Σ.

5. Mining Frequent Itemsets and Association Rules

Let $I = \{i_1, i_2, \cdots, i_m\}$ denote a set that contains m different attributes. n-itemset mean that there have n items. Every transaction in transaction database TD has an only identifier TID and has an itemset $T \subseteq I$.

Definition 5.1[19] (Frequent itemsets) Let $l \subseteq I$, the support of an itemset l is the percentage of objects in transaction database containing l. l is a frequent itemset if $support(l) \geq minsupport$.

Definition 5.2[19] (Association rules) An association rule r is an implication between two frequent itemsets $l_1, l_2 \subseteq I$ of the form $l_1 \to (l_2 \setminus l_1)$ where $l_1 \subset l_2$. The support and the confidence of r defined as: $support(r) = support(l_2)$ and $confidence(r) = support(l_2) / support(l_1)$.

Generally speaking, mining association rules includes two steps:

1. Finding all frequent itemsets,
2. Finding possible association rules from frequent itemsets.

Transaction database TD can be seen as a formal context (U, D, R) easily [23], where U is the set of transaction and D is the set of items in TD. For $x \in U$, $d \in D$, xRd if and only if d belongs to the itemsets of transaction x.

Theorem 5.1 For a concept in closed label lattice $C(extention(C), Closedlabel(C), intention(C))$, and itemset $S = intention(C)$, the support of S is $Support(S) = extention(C) / |U|$

Theorem 5.2 For a concept of closed label lattice $C(extention(C), Closedlabel(C), intention(C))$, all the member of $Closedlabel(C)$ has the same support.

Because for $\forall s \in Closedlabel(C)$, we have that $h(s) = intention(C)$, $g(s) = extention(C)$. So they have same support.

Theorem 5.3 The set of frequent itemsets in a transaction database is
$S=\{intention(C) \mid C \in L(K)\ and\ extention(C) \geq \theta\}$
$\bigcup \{s \mid h(s) = intention(C)\}$.

Because the set of closed label is a subset of frequent itemsets, and all the frequent itemsets can deduced from closed label, such as adding some attributes to the closed label (attributes in the intent of concept). Therefore, we only think about closed label when mining association rules, removed part of redundancy rules.

Definition 5.3 For arbitrary two concept C_1, C_2 in closed label lattice, if $support(intention(C_2)) \geq \theta$, $extention(C_1) \supset extention(C_2)$ and $support(intention(C_2))/support(intention(C_1))=$ $extention(C_2)/extenction(C_1) \geq \delta$, then (C_1, C_2) is called (θ,δ)-candidate binary group.

Theorem 5.4 $intention(C_1) \to (intention(C_2) \setminus intention(C_1))$ is (θ,δ)-association rule if and only if (C_1, C_2) is (θ,δ)- candidate binary group

From above, we know that all of the (θ,δ)-association rules can be produced from (θ,δ)- candidate binary group. According to definition 5.2, for arbitrary two concept C_1, C_2, if $C_1 \succ C_2$, i.e. C_1 is a father node of C_2, $extention(C_2) \geq \theta$ and $extention(C_2)/extenction(C_1) \geq \delta$, then (C_1, C_2) is (θ,δ)-candidate binary group. So when mining association rules, we only consider concept and its child concept.

Theorem 5.5 Let $A \to B, B \to C$ be (θ,δ)-association rule, and confidence is c_1, c_2 respectively, then confidence of $A \to C$ is $c_1 * c_2$.

Proof Because $A \to B, B \to C$ are (θ,δ)-association rule, there exists three nodes C_1, C_2, C_3, satisfying the following conditions:

$$h(A) = intention(C_1),$$
$$h(A \bigcup B) = intention(C_2)$$

and

$$h(B \bigcup C) = intention(C_3).$$

Also we have $C_1 \succ C_2 \succ C_3$.

Thereby,

$$c_1 = extention(C_2)/extenction(C_1),$$
$$c_2 = extention(C_3)/extenction(C_2),$$

Hence, confidence of $A \to C$ is

$$extention(C_3)/extenction(C_1) = c_1 * c_2.$$

When mining association rules in closed label lattice, we still only consider candidate binary group composed of father and child node.

Algorithm 5.1 Mining generation set of association rules on closed label lattice.

Input: Given closed label lattice L, support θ and confidence δ.

Output: Generation set of association rules Σ.

Step 1 : Initialization $\Sigma := \varnothing$.

Step 2: For every node C in lattice L,
if $extenction(C)/|U| \geq \theta$, rearrange as $Card(extention(C))$ ascending, and put into $Mark$, let $\mathrm{Mark}' := Mark$.

Step 3: Get a concept C in Mark', $\mathrm{Mark}' := \mathrm{Mark}' - C$.
If $Closedlabel(C) = \varnothing$, skip to step 8.

Step 4: $Parents'(C) := Parents(C)$.

Step 5: Get a concept N in $Parents'(C)$,
let $Parents'(C) := Parents'(C) - N$. If $Closedlabel(N) = \delta$, skip to step 7.

Step 6: For every element X in $Closedlabel(N)$ and every element Y in $Closedlabel(C)$, generate rule
$X \to \{Y - X\}$, if
$extention(C)/extention(N) \geq \delta$,
$\Sigma := \Sigma \bigcup \{X \to \{Y - X\}(extenction(C)/|U|, extention(C)/extention(N))\}$.

Step 7: If $Parents'(C) \neq \varnothing$, skip to step 5.

Step 8: If $\mathrm{Mark}' \neq \varnothing$, skip to step 3.

Step 9: End, output Σ.

6. Example and Analysis of Experimental Results

Example 6.1 Table 1 is a given transaction database, and Table 2 is corresponding formal context.

Fig. 1 is closed label lattice based on formal context in Table 2. Generation set of implication rules extracted from the lattice is as follows: $\{f \to e, d \to h, h \to d, a \to i, b \to g, b \to f, cd \to e, ed \to c, ch \to e, eh \to c, ca \to g, ga \to ci, gc \to a, gi \to a, ei \to f, fi \to ec, fc \to ei, di \to a, hi \to a\}$, there are 19 rules. For example, the rules generated from concept $(2,\{gc,ci,ga,ca\},acgi)$ are $\{ca \to g, ga \to ci, gc \to a, gi \to a\}$, then we can produce the following rules: $\{gca \to i, gia \to c, gci \to a, gi \to ac, gc \to ai, ca \to gi\}$, so the rules that we extracted are more simpler. From above we can see that generation set is only a smaller set of rules. Other rules could be generated from generation set if needed.

Let $\theta = 0.4$, δ =0.6, then the generation set of association rules generated from lattice are : { $e \to f$ (0.4, 0.67), $e \to c$ (0.4, 0.67), $c \to e$ (0.4, 0.67), $c \to i$ (0.4, 0.67), $i \to c$ (0.4, 0.67), $i \to a$ (0.4, 0.67}.

The time complexity of incremental building lattice in [22] is $O(2^k \| L \|)$, in which $\| U \|$ is the number of objects and k is the maximum value of the number of attributes $\| f(\{x\}) \|$ $(x \in U)$. $\| L \| = 2^k \| U \|$ is the number of nodes in lattice. In the paper, because the computing closed label of every new added node or of generate node need scan all its father nodes. The maximum number of father nodes is 2^k -1, so the time complexity of the algorithm is

Table 1 Transaction database

TID	T
0	b, e, f, g
1	c, d, e, h
2	$a, c, g\ i$
3	c, e, f, i
4	a, d, h, i

Table 2 Formal context

	a	b	c	d	e	f	g	h	i
0	0	1	0	0	1	1	1	0	0
1	0	0	1	1	1	0	0	1	0
2	1	0	1	0	0	0	1	0	1
3	0	0	1	0	1	1	0	0	1
4	1	0	0	1	0	0	0	1	1

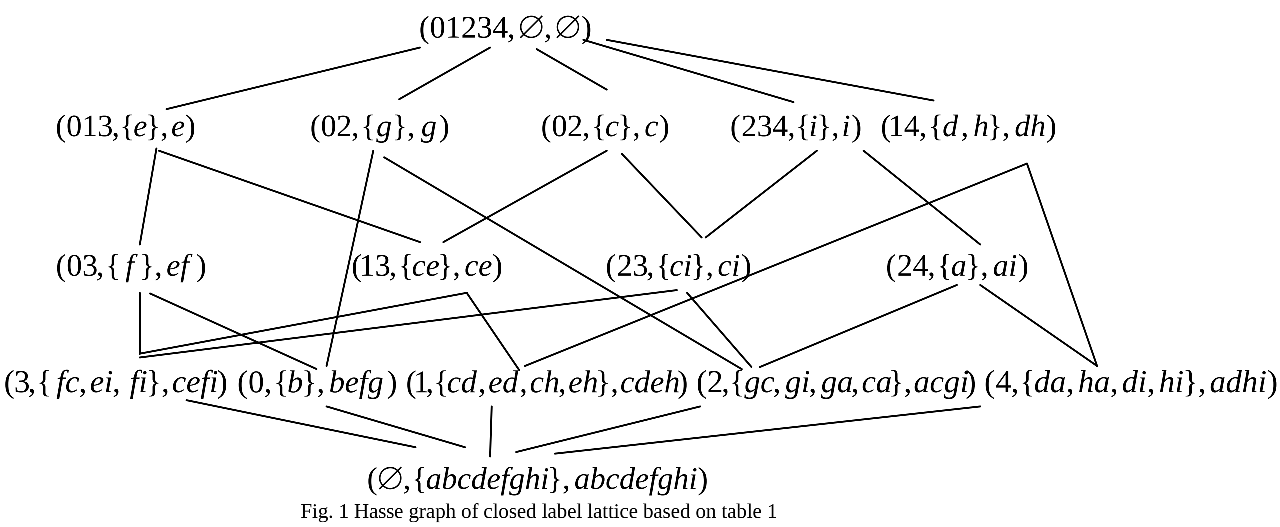

Fig. 1 Hasse graph of closed label lattice based on table 1

$O(\| L \| (2^k + 2^k - 1)) = O(2^k \| L \|)$. The number of repeat times that the algorithm of mining implication rules is the number of father nodes, so the time complication of mining generation set is $O(\| L \| (2^k - 1)) = O(\| L \| 2^k)$. In practice, the number of father nodes is far less that the worst, and we recorded father nodes of every node, so the efficiency of mining rules will has some improvement. As for mining association rules, the scan number is even less because of pruning based on support.

The above algorithms have been realized using Delphi5.0 in Windows XP. We have tested from building closed label lattice to mining rules for some random dataset of given 15 attributes and each object has 5 attributes. Fig. 2 denotes the relationship of average time of running and the number of objects. The results indicate that the algorithm has certain stability.

Fig. 3 denotes the relationship of the number of object and the number of rules in generation set. In the same time, we compared the results with [24]. In Fig. 3, the above curve shows the results of [24] and the below curve denotes the results of our algorithm. We can see from the figure that the set of rules we extracted is smaller than the traditional algorithm. So at certain number of attributes, our algorithm has some advantage.

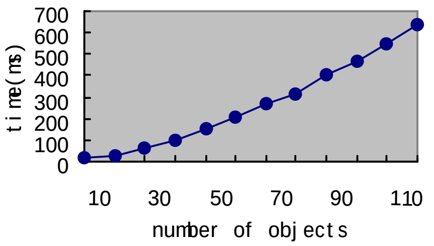

Fig. 2 Relation of execute time of algorithm and objects set

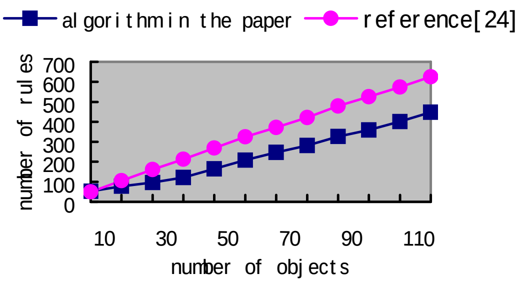

Fig. 3 Relation of the number of rules and objects set

7. Conclusions

In the paper, a new framework for extracting association rules based on concept lattice and the concept of closed itemsets is proposed. The algorithm of incremental building lattice based on closed label lattice is given, and the algorithms of mining generation set of implication rules and association rules are introduced. The number of non-redundant rules produced by the new approach is smaller than the rule set from the traditional approach. Experiments confirm the utility of the framework in terms of reduction in the number of rules, and in terms of time. In future work, we will study the method of extracting classification rules based on the new framework.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 70471003, No. 60275019), the Doctor Authorization Foundation of the Ministry of Education (No. 20050108004), Key Project of Science and Technology Research of the ministry of Education of China (No. 206017) and the Top Scholar Foundation of Shanxi, China.

References

[1] R. Agrawal, T. Imielinski and A. Swami, "Mining association rules between sets of items in large databases", Proc. Sixth International Conference on Extending Database Technology, Vaiencia, Spain, pp. 105-119, 1993.

[2] R. Agrawal and R. Srikant, "Fast algorithm for mining association rules", Proc. of 1994 International Conference on Very Large Data Bases, Santiago, Chile, pp. 487-499, 1994.

[3] D. W. Cheung, J. Han, "Maintenance of discovered association rules in large databases: an incremental updating technique", Proc. of 1996 International Conference on Data Engineering (ICDE'96), New Orleans, Louisians, USA, 1996.

[4] P. Jian, H. Jiawei and M. Runying, "Clost: an efficient algorithm mining frequent closed itemsets", Proc. of the 2000 ACM SIG SIGMOD International Workshop on Data Mining and Knowledge Discovery, Dallas, TX, pp. 11-20, 2000.

[5] Y. Q. Zhu, Z. H. Sun and X. J. Ji, "Incremental updating algorithm based on frequent pattern tree for mining association rules", Journal of Computer (in Chinese), vol. 26, no. 1, pp. 91-96, 2003.

[6] B. Liu, W. Hsu, S. Chen, and Y. Ma, "Analyzing the subjective interestingness of association rules", Intelligence Systems, vol. 15, no. 5, pp. 47-55, 2000.

[7] Y. Q. Zhu, Z. H. Sun, "Fast updating frequent itemsets," Journal of Computer Research and Development (in Chinese), vol. 40, no. 1, pp. 94-99, 2003.

[8] R. Wille, “Restructuring lattice theory: an approach based on hierarchies of concepts”, In: Rival I. Ordered Sets, Dordrecht-Boston: Reidel, pp. 445–470, 1982.

[9] R. Godin, G. Mineau, R. Missaoui, M. St-Germain and N. Faraj, “Applying concept formation methods to software reuse”, International Journal of Knowledge Engineering and Software Engineering, vol. 5, no. 1, pp. 119-142, 1995.

[10] G. W. Mineau and R. Godin, “Automatic structuring of knowledge bases by conceptual clustering,” IEEE Trans. on Knowledge and Data Engineering, vol. 7, no. 5, pp. 824-828, 1995.

[11] C. Carpineto and G. Romano, “A lattice conceptual clustering system and its application to browsing retrieval”, Machine Learning, vol. 24, no. 2, pp. 95-122, 1996.

[12] R. Cole and P. Eklund, “Scalability in formal concept analysis”, Computational Intelligence, vol. 15, no. 1, pp. 11-27, 1999.

[13] R. Cole, P. Eklund and G. Stumme, “CEM – a program for visualization and discovery in email”, Proc. 4th European Conference on Principles and Practice of Knowledge Discovery in Databases. Berlin: Springer Verlag, Sep. 2000.

[14] J. Y. Liang and J. H. Wang, “An Algorithm For Extracting Rules Generating Set Based On Concept Lattice”, Journal of Computer Research and Development (in Chinese), vol. 41, no. 8, pp. 1339-1344, 2004.

[15] K. S. Qu, J. Y. Liang, J. H. Wang, et al, “The algebraic properties of concept lattice”, Journal of Systems Science and Information, vol. 2, no. 2, pp. 271-277, 2004.

[16] R. Missaoui and R. Godin, “Search for concepts and dependencies in databases”, Rough sets, and Fuzzy Sets and Knowledge Discovery. London：Springer-Verlag, pp. 16-23, 1994.

[17] N. Pasquier, Y. Bastide, R. Taouil, and L Lakhal, “Discovering frequent closed itemsets for association rules”, Proc. 7th International Conference on Databases Theory, Jerusalem, Israel, Jan.1999.

[18] N. Pasquier, Y. Bastide, R Taouil, and L Lakhal, “Efficient mining of association rules using closed itemset lattice”, Information Systems, vol. 24, no. 1, pp. 25-46, 1999.

[19] Y. Bastide, N. Pasquier, R Taouil, G. Stumme and L. Lakhal, “Mining minimal non-redundant association rules using frequent closed itemsets”, Computational Logic, pp. 972-986, 2000.

[20] M. Zaki, “Generating non-redundant association rules”, Proc. 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 34-43, 2000.

[21] K. Y. Hu, Y. C. Lu and C. Y. Shi, “Advances in concept lattice and its application”, Journal of Tsinghua University (in Chinese), vol. 40, no. 9, pp. 77-81, 2000.

[22] R. Godin, R. Missaoui and H. Alaoui, “Incremental concept formation algorithms based on Galois (concept) lattices,” Computational Intelligence, vol. 11, no. 2, pp. 246-267, 1995.

[23] Z. P. Xie and Z. T. Liu, “Concept lattice and association rule discovery”, Journal of Computer Research and Development (in Chinese), vol. 37, no. 12, pp. 1415-1421, 2000.

[24] Z. H. Wang, K. Y. Hu, X. G. Hu, Z. T. Liu and D. C. Zhang, “General and incremental algorithms of rule extraction based on concept lattice”, Chinese Journal of Computers (in Chinese), vol. 22, no. 1, pp. 66-70, 1999.

Dr Jiye Liang is a Professor in the School of Computer & Information Technology, Shanxi University, in China. Dr Liang received his PhD in Information Science from Xi’an Jiaotong University, and was engaged on Postdoctoral research about Computer Science and Technology in Institute of Computing Technology, the Chinese Academy of Sciences. Dr Liang has published over 60 refereed papers in the field of rough set theory, concept lattice and granular computing. His current research interests are data mining and knowledge discovery in database, soft computing and granular computing.

Junhong Wang is a PhD candidate in the School of Computer & Information Technology, Shanxi University, in China. Her research interests include data mining and concept lattice.