TELKOMNIKA, Vol. 11, No. 7, July 2013, pp. 3604 ~ 3610
e-ISSN: 2087-278X

Ming Non-redundant Associations From the Frequent Concept Sets on FP-tree

Wang Hui
Information Security Engineering Department, People's Public Security University of China, Beijing, China, 100038
e-mail: wanghui0330@gmail.com

Abstract

The classical algorithm for mining association rules is low efficiency. Generally there is high redundancy between gained rules. To solve these problems, a new algorithm of finding non-redundant association rules based on frequent concept sets was proposed. The Hasse graph of these concepts was generated on the basis of the FP-tree. Because of the restriction of the support most Hasse graphs have lose lattice structure. During building process of the Hasse graph, all nodes were formatted according to the index of items which were found in the frequent-item head table. At the same time these nodes were selected by comparing supports. In the Hasse graph, the intention of node is frequent itemset and the extension of node is support count of this item set. And the non-redundant association rules were gained by scanning the leaf nodes of the graph. The simulation shows the feasibility of the algorithm proposed.

Keywords*: data mining, non-redundant, association rules, frequent concept, FP-tree*

1. Introduction

Data mining is one of the important means for data analysis in many fields [1-3]. Now many algorithms have applied to generate better results successfully. Among these algorithms mining association rules is most popular in business field.The classical algorithms for mining association rules are Aprior and FP-growth [4-6]. There are two steps to gain high confidence association rules for the two algorithms. The first step is finding frequent itemsets. This is the key measure and influences the efficiency of the whole algorithm. It takes many times to scan database in this process usually. Exploring appropriate knowledge model and reducing the number of scanning database are the main ways to improve algorithms. The second step is extracting association rules on the frequent itemsets. Compared with the first step, this step is simpler. But the association rules generated from the frequent itemsets are high redundant. Especially when the support and confidence are low the number of rules generated will increase exponentially. This reduses the predictive ability of the rules gained. So it is essential to study compact representation and find non-redundant rules in the mining process.

The concept lattice describes the relationship between transaction and attribute and reflects the generalization and specialization between concepts [7, 8]. At the same time the Hasse graph of the concept lattice is simple and easy to realize. It is more intuitive to discover association rules based on the Hasse graph. But the efficiency of constructing concept lattice directly determines the applicability of mining association rules. Furthermore, compare with the Aprior algorithm, the FP-growth algorithm only need scan database twice for building a FP-tree. And all the frequent itemsets are found on the tree. The candidate itemsets are also avoided. However, there are exponential growth of the association rules with the the frequent itemsets increasing. The large amount of redundancy exists between rules, espetially with the small support and low confidence. So the rules are difficult to understand and utilize. To solve these problems and obtain reliable association rules for the database, a new algorithm of finding non-redundant association rules based on frequent concept set is proposed. The algorithm is composed of two sub-algorithms. The one is DFCSA (Discover Frequent Concept Set Algorithm) to build Hasse graph of the frequent itemsets on the basis of the FP-tree. Another is NAREA (Non-redundant Association Rules Extraction Algorithm) to gain non-redundant association rules from Hasse graph. In the building process of the graph, the pruning is

Received January 16, 2013; Revised April 5, 2013; Accepted April 15, 2013

completed simultaneously. All the information about generating association rules can be found in the Hasse graph of the frequent itemsets.

Here, it is clear that the advantage of the classical FP-growth algorithm is integrated into the constructing process of frequent concept sets for the new algorithm. The non-redundant association rules gained on the Hasse graph are ensured by rule screening. After non-redundant extraction the rules will be more understandable. The simulations show that the algorithm is intuitive and efficient.

2. Related Concepts

The related concepts and definitions of association rules and concept lattice are described as follows.

Definition 1 [7] Giving a background as $T=(D,I,R)$, it is the group with three elements. Where, D is the transaction sets. I is the attribute sets. R is a relation and $R \subseteq D \times I$. If there is only one partial order relation to generate the lattice structure, the background is called as concept lattice.

Definition 2 [7] The node of lattice L is a ordered pairs and expressed as $\langle X,Y \rangle$. Where, X is a collection of transactions and called the extension. Y is the common attribute of all instances in X and called the connotation. Each pair is complete, expressed as

$$\begin{aligned} X &= \alpha(Y) = \{x \in D \mid \forall y \in Y, xRy\} \\ Y &= \beta(X) = \{y \in I \mid \forall x \in X, xRy\} \end{aligned} \quad (1)$$

Definition 3 [9] Let item set $I_1 \subseteq I$, and the support of I_1 in the transaction sets D is expressed as

$$Support(I_1) = \|\{t \in D \mid I_1 \subseteq t\}\| / \|D\| \quad (2)$$

where, the support is the percentage of affairs in D, which contain I_1.

Definition 4 [9] The confidence of association rule $(I_1 \Rightarrow I_2)$ which is defined in the attribute set I and transaction sets D, is expressed as

$$Confidence(I_1 \Rightarrow I_2) = Support(I_1 \cup I_2) / Support(I_1) \quad (3)$$

where, $I_1, I_2 \subseteq I \cdot I_1 \cap I_2 = \varnothing$. The confidence is the ratio of affairs number which is respectively included in $I_1 \cup I_2$ and I_2.

Definition 5 [9] The Strong Association Rule (SAR) is the association rule which satisfies with $Minsup$ (Min-support) and $Minconf$ (Min-confidence) in D and I. When SAR is a nonempty set, it is called frequent item sets. If any element of SAR doesn't contained by the others, it is called the maximum frequent item sets.

3. Mining Non-Redundant Association Rule

In the mining process for association rules, the rules must be satisfied with the minimum support threshold. Compared with the specific transactions contained by the frequent itemset, the support calculation only concerned about the quantity of these transactions. So the specific information contained by the extension X of the concept $\langle X,Y \rangle$ is ignored. Here, X is replaced by the cardinal number $|X| \cdot |X|$ means the number of the transactions involved by itemset Y. The concept $\langle |X|, Y \rangle$ becomes concept quantified. This concept is more concise and easier to calculate support for mining association rule. Moreover, because of support threshold's limit, most of Hasse graphs lose the structure of lattice. Because all subsets of the maximum frequent are still frequent itemsets. So the Hasse graph of frequent itemsets contain all frequent concept. Non- redundant association rules can be found on this graph.Building Hasse graph is the most important step at begining.

3.1. Discover Frequent Concept Sets Algorithm (DFCSA)

According to Definition1 and Definition 2, DFCSA algorithm uses concept node and builds Hasse graph. But the Hsaae graph is generated on the basis of the FP-tree which is constructed by the classical FP-growth algorithm. Considering the Hasse graph with the value of the 1 frequent itemsets, the layers $L_i(i \geq 2)$ are generated by indexing Htable (Header-table) of the FP-growth algorithm. Other nodes are selected by comparing with the minimum support threshold. So each node is frequent. And the connotation of each node is frequent itemset. The maximum frequent item sets are composed of the connotations of leaf nodes. At the same time, the Sub-Hasse with frequent node value 1 is not cross each other. The constructing process is described below.

Input: transaction database D, minimum support threshold $Minsup\text{-}count$.

Output: the Hasse diagram of frequent concept set which is corresponding to D.

Step 1: through scanning the database D once, the 1 frequent item sets are generated. The support count number is recorded. And then, 1 frequent items list T_F is obtained by descending the count number of the frequent items. Let the number of frequent item set with value 1 is N.

Step 2: Constructing the Htable and the FP-tree of T_F. Each node of the FP-tree is consisted of node name, node count number, node-link and pointer of parent node [6].

Step 3: The root of L_0 layer node is created directly, which is marked as $\langle \|D\|, \varnothing \rangle$. According to T_F, the L_1 layer is created. Its node is expressed as $\langle \|A_i\|, \{A_i\} \rangle$. Where, $A_i (i \leq N)$ was frequent items in T_F, $\|A_i\|$ is the support count number of A_i.

Step 4: $i = 1$. Based on the Htable's item order, each nodes A_i of FP-tree is respectively executed by depth-first searching.

Step 5: If $N \neq 0$, turn to Step 6 else Step 8.

Step 6: If $A_i.node-link \neq \wedge$, then generate Sub-hasse graph of the node A_i,else Step 7.

Step 7: $i = i + 1$, $N = N - 1$,turn to Step5. // Layer $L_j (j > 1)$ of Hasse diagram is generated in above steps.

Step 8: Join the nodes with cover relations and output the Hasse graph corresponding to D and support threshold.

According to the Hasse diagram of the frequent concept constructed with the method described above, the frequent itemsets with value 1 were considered during the constructing process of FP-tree. For all transactions contained by each frequent concept have been sorted comparing with the support number, the related nodes of the Hasse-diagram's L_1 layer appear orderly. And the inner nodes above are no-repeat, the Sub-Hasse diagram of frequent item sets with value 1 can be generated independently.

In order to verify the effectiveness of DFCSA, the sample database and the minimum support count number is same as the reference [10]. The database is shown in table1 below. The minimum support count number is 2. At first the Htable and FP-tree of the sample database are getted by FP-growth algorithm and shown in Figure 1. The Hasse diagram of frequent concept corresponding with FP-tree is shown in Figure 2.

Seen from the Figure 2, the Hasse graph above isn't lattice structure already. But to abtain the non-redundant association rules will be easier with this structure. And the comparison of nodes' connotation with direct and indirect relationship will be executed by bottom-up principle.

Table 1. Sample database

TID	Transaction itemsets
1	ABC
2	ACD
4	ABCD
5	A
6	BCD

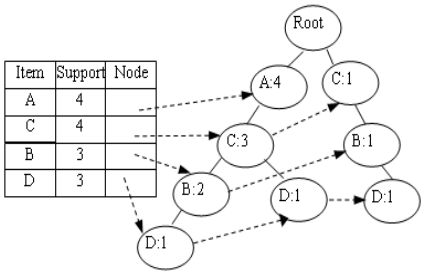

Figure 1. A frequent tree of the sample.

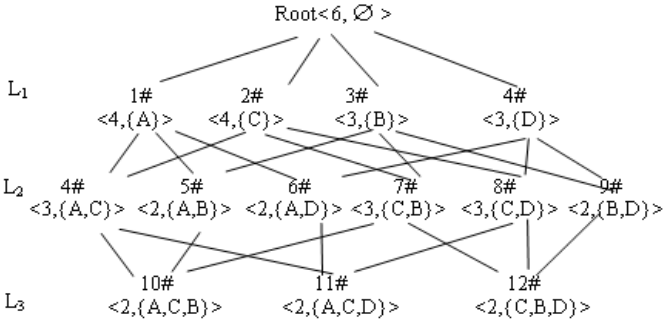

Figure 2. Frequent concept Hasse diagram**.**

3.2. Non-redundant Association Rules Extraction Algorithm (NAREA)

After generating the Hasse diagram of frequent concept set, extracting rules becomes easier. Considering the Hasse diagram and letting the Minimum reliability as $Minconf$, in addition to root node the content of other nodes is frequent item sets. And the extension of the node is the support count number in transaction database D. In the Hasse graph, the leaf nodes contain all the maximum frequent itemsets for a support threshold given. The association rules can be obtained through scanning the cross-layers except the root node. But the rules extracted can not be avoided redundancy with the classical algorithms. Because the non-redundant rules will provide more information under the minimum confidence threshold given. The definitions about non-redundant association rules are as follows.

Definition 6 According to the rule $A \Rightarrow B$ and $C \Rightarrow D$, if $C \Rightarrow D$ will be gained from $A \Rightarrow B$ by some inference rules that $C \Rightarrow D$ is a redundant rule relative to $A \Rightarrow B$.

Definition 7 Let $R: X \Rightarrow Y$ comes from itemset I. $X \cup Y = I$. If there isn't rule $R': X' \Rightarrow Y'$ in I that the rule $R: X \Rightarrow Y$ is called the smallest non-redundant rules in itemset I. Where, $X' \cup Y' = I$, $X' \subset X$, $Y \subset Y'$.

Definition 8 Let $R: X \Rightarrow Y$ comes from itemset I_j. $X \cup Y = I_j$. The rule $R': X' \Rightarrow Y'$ comes from itemset I_k ($I_k \subseteq I_j$). $X' \cup Y' = I_k$. If $X \subset X'$ that rule $R': X' \Rightarrow Y'$ is called strict non-redundant rules relative to $R: X \Rightarrow Y$.

The definition 6 shows that there is no cross between non-redundant rules about the transaction database. The definition 7 shows that minimum non-redundant association rules have the characteristics with minimum antecedent and maximum consequent. So in order to avoid redundancy the rules that contain fewer projects of antecedent in the same itemset need to be calculated at first. The definition 8 shows that the rules generated among the the frequent itemsets and its subset are redundant. To avoid redundancy the rules generated by the maximum frequent itemsets must be calculated firstly.

In the Hasse graph, the intension of the leaf node is the maximum frequent itemset. So the rule $p \Rightarrow Y - p$ generqted by any leaf node $C\langle sup(Y), Y\rangle (p \in Y)$ is the smallest non-redundant

rule. If rule $p \Rightarrow Y - p$ is ture then all the other rules involving p in antecedent can be derived by this rule. If the rule does not meet the confidence then the rule involving p in antecedent can be found by the following method.

Firstly, the non-redundant rules involving p in antecedent will be generated by the proper subset of Y. These rules are gained by the upper layer of the leaf nodes of the Hasse diagram.

Secondly, the non-single item rule involving p in antecedent will be generated by Y. For example, If rule $A \Rightarrow BCD$ is not ture then rule $AB \Rightarrow CD$ may be ture. These rules are gained by the lower layer of the node $C\langle sup(\{A\}),\{A\}\rangle$.

Here, the rules generated by the described method above are no cross. So the non-redundant rules of the frequent concept set quantified can be completed interactively by two processes above. The algorithm of non-redundant association rules extraction is described below.

Input: The Hasse graph of frequent concept set (Let the number of leaf nodes is L) and minimum confidence $mincof$.

Output: The non-redundant strong association rule set $NSAR$

Step 1: All the leaf nodes $C_i\langle sup(Y_i),Y_i\rangle$ enter the queue Q_1. $i = 1,2,\cdots,L$.

Step 2: $i = 1$. $NSAR = \varnothing$.

Step 3: If the queue Q_1 is empty then turn to Step 11. Else $C_i\langle sup(Y_i),Y_i\rangle = Outqueue(Q_1)$, $j = 1$, $Y_i = \{p_1, p_2, \cdots, p_m\}$.

Step 4: If $|Y_i| = 1$ then turn to Step 10.

Step 5: If $j > m$ then turn to Step 10.

Step 6: $confidence = sup(Y_i) / sup(\{p_j\})$.

Step 7: If $confidence \geq mincof$ then $NSAR = NSAR \cup \{p_j \Rightarrow Y_i - p_j\}$, $j = j+1$ and turn to Step5. Else Step 8.

Step 8: Call $\text{Gen-Rules-Subsets}(C_i, p_j)$.

Step 9: Call $\text{Gen-Rules}(C_i, p_j)$.

Step 10: $i = i+1$. If $i \leq L$ then Step3.

Step 11: Output $NSAR$.

Procedure $\text{Gen-Rules-Subsets}(C_i, p_j)$

$EnQueue(Q_2, parent(C_i))$

if $QueueEmpty(Q_2)$ then

return

else

$N = Queuelength(Q_2)$

for ($k = 1, k \leq N, k++$)

$C_i = Outqueue(Q_2)$

if $Y_i \neq \{p_j\}$ then

if $Y_i \cap \{p_j\} \neq \varnothing$ then

$confidence = sup(Y_i) / sup(\{p_j\})$

if ($confidence \geq mincof$) then

$NSAR = NSAR \cup \{p_j \Rightarrow Y_i - p_j\}$

else

call $\text{Gen-Rules-Subsets}(C_i, p_j)$

endif

endif

endif

endfor

endif
Procedure $\text{Gen-Rules}(C_i, p_j)$
$EnQueue(Q_2, Child(\langle \sup\{p_j\}, \{p_j\} \rangle))$
if $QueueEmpty(Q_2)$ then
return
else
$N = Queuelength(Q_2)$
for ($k = 1, k \le N, k++$)
$C = Outqueue(Q_2)$
if $Y \neq Y_i$ then
$confidence = sup(Y_i) / sup(Y)$
if ($confidence \ge mincof$) then
$NSAR = NSAR \cup \{Y \Rightarrow Y_i - Y\}$
else
call $\text{Gen-Rules}(C_i, Y)$
call $\text{Gen-Rules-Subsets}(C_i, Y)$
endif
endif
endfor
endif

All the non-redundant association rules can be generated by the algorithm above. The procedure Gen-Rules-Subsets calculates the rules with a single set antecedent of the frequent itemsets. The procedure Gen-Rules produces the non-single item rule in antecedent.

According to figure 2, there are three leaf nodes such as $\{ACB\}, \{ACD\}, \{CBD\}$. When $mincof = 50\%$, the result is best for non-redundant rules extraction. There are nine smallest non-redundant rules such as $A \Rightarrow CB$, $\Leftrightarrow AB$, $B \Rightarrow AC$ and so on. The other rules are redundant. If non-redundant extraction doesn't exist, there will be $2^6 \times 3 = 192$ rules of this graph.

4. Simulation

To further illustrate the accuracy and effectiveness of the NAREA on the Hasse graph, the matlab simulation of NAREA, FP-Growth and Aprior are done respectively. The mushroom data set of machine learning database UCI (http://archive.ics.uci.edu/ml/) is chosen as the simulation data. There are 8124 transactions and 23 properties. The data set is provided by American University of California. Because the mushroom set is dense datasets the association rules of this datasets are redundancy greatly. The test of the processes for generating non-redundant association rules is done by using three algorithms above. The simulation result is shown in Figure 3.

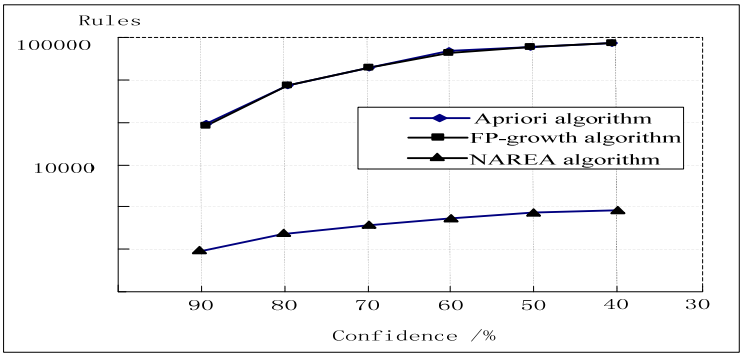

Figure 3. Rules Extraction Algorithm Compares

Figure 3 shows that the number of association rules gained by Apriori and FP-growth are same. So two curves almost repeat in the Figure 3. In contrast, under the same confidence NAREA algorithm removes redundancy and reduces the number of mining rules. This improves the readability and comprehensibility of the rules. Therefore, the NAREA algorithm has high operating efficiency of mining.

5. Conclusion

In this paper, the DFCSA and NAREA algorithms based on FP-tree were proposed. Both algorithms are used to extract non-redundant association rules. The process for pruning branch is executed synchronously during constructing Hasse graph of frequent concepts. Considering the sub-tree corresponding to 1 frequent item, the algorithm reduces the comparing count between the sequence nodes. At the same time, the concept of non-redundant association rules have been put forward through different forms. And the Hasse graph of the frequent concept contains all information for extracting non-redundant association rules. It is shown that the DFCSA and NAREA algorithms are effective and efficient by simulating and comparing with other algorithm.

Acknowledgements

The paper is supported by the teaching project of the People's Public Security University of China named as the feasibility research for network real name.

References

[1] Liu Bin, Qiu Huayong, Shen Yizhen. Realization and Application of Customer Attrition Early Warning Model in Security Company. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2012; 10(5): 1106-1110.
[2] Lijuan Zhou, Hui Wang, Wenbo Wang. Parallel Implementation of Classification Algorithms Based on Cloud Computing Environment. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2012; 10(5): 1087-1092.
[3] Awad Ali, Moawia Elfaki, Dayang Norhayati. Using Naïve Bayes and Bayesian Network for Prediction of Potential Problematic Cases in Tuberculosis. *International Journal of Informatics and Communication Technology (IJ-ICT)*. 2012; 1(2): 63-71.
[4] Agrawal R, Imieliński T, Swami A. *Mining Association Rules between Sets of Items in Large Databases*. Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data. New York. 1993: 207-216.
[5] Agrawal R, Shfer JC. Parallel Mining of Association Rules. I*EEE Transactions on Knowledge and Data Engineering*. 1996; 8(6): 962-969.
[6] Han J, Pei J, Yin Y. *Mining Frequent Patterns Without Candidate Generation*. Proceeding of the 2000 ACM SIGMOD International Conference on Management of Data, New York. 2000; 29: 1-12.
[7] Ganter B, Wille R. Formal Concept Analysis: Mathematical Foundations. Berlin: Springer-Verlag. 1999.
[8] Wille R. *Restructuring Lattice Theory: an Approach Based on Hierarchies of Concepts.* Proceedings of the 7th International Conference on Formal Concept Analysis, Berlin. 2009: 314-339.
[9] Mao G, Duan L, Wang S. Principles and Algorithms of Data Mining. BeiJing: Tsinghua University Press. 2005.
[10] Chen X, Wu Y. Mining Associations Based on Simplified Concept Lattice by Improved Algorithm. *Application Research of Computers*. 2011; 28(4): 1293-1295.