Journal of Theoretical and Applied Information Technology
28th February 2013. Vol. 48 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A SHORTEST ANTECEDENT SET ALGORITHM FOR MINING ASSOCIATION RULES*

[1]QING WEI, [2]LI MA [1]ZHAOGAN LU

[1]School of Computer and Information Engineering, Henan University of Economics and Law, Zhengzhou 450002, Henan, China

[2] Department of Information Technology Engineering, Yellow River Conservancy Technical Institute, Kaifeng City, China

E-mail: weiqing.huel@gmail.com

ABSTRACT

Aiming at the redundancy problem of association rules in the mining process of mining association rules, a kind of shortest antecedent set algorithm (SASA) for mining association rules is proposed in this paper, and the algorithm is supported by the set-enumeration tree structure. The proposed algorithm is able to record a kind of subset for association rules without any information loss, and this kind of subset contains the entire generated association rule nondestructive. Experimental results show that the algorithm can significantly reduce the number of association rules, and it also improves the efficiency of analyzing the association rules for users.

Keywords: *Association Rule, Shortest Antecedent Set, Key Rule*

1. INTRODUCTION

Association rule is a very important branch in the area of data mining. The mining of association rule is implemented by two steps, the first one is mining the frequent items; the second one is generating association rules that meet the condition of min-confidence (mc) [1-3]. The first step is usually the core step in the mining process of association rules, and the efficiency of data mining is affected mainly by finding of frequent item. In experience, there are a great number of association rules that generated via frequent items. For instance, there are more than 10 thousands of association rules results in one mining case on medical treatment, therefore, users have to spend a great deal of time on analyzing so many rules if they want to analyze an association relation on an illnesses case, that means, it waste not only time but also space. Actually, it is not necessary to generate all the rules, on the contrary, it is enough for us to generate some key rules only, which contain all the others information, that is to say, those key association rules are able to deduce all rules without any data loss.

GRSET is a kind of algorithm based on set-enumeration, its function is to generate all the association rules which based on one frequent item, and this method generates large numbers of association rules. For example, in table 1, the item I3I2I8I5 is frequent item because it meets the condition of min-confidence (mc) and min-support (ms). And according to GRSET algorithm, only one frequent item generates 10 association rules, they are

{I5⇒I3I2I8 ， I3I2⇒I8I5 ， I2I8⇒I3I5 ， I2I5⇒I3I8，I3I5⇒I2I8，I8I5⇒I3I2，I3I2I8⇒I5 ，I3I2I5⇒I8，I3I8I5⇒I2，I2I8I5⇒I3}.

But in fact, it is a huge waste to generate so many rules. Each expression is a rule, and here, we call the string before the "⇒" a shortest antecedent, and the string after the "⇒" a tail-item. For example, in the second line of the set, 'I3I2⇒I8I5', we call I3, I2 the shortest antecedent of the rule 'I3I2⇒I5I8', and I5, I8 the tail-item of the rule 'I3I2⇒I5I8'.[4] Therefore, the whole set

{I3I5⇒I2I8 ，I2I5⇒I3I8，I8I5⇒I3I2，
I3I2I8⇒I5，I3I2I5⇒I8，I3I8I5⇒I2，I2I8I5⇒I3} (1)

can be deduced by a set only includes 3 rules via a tail-to-head method , the set is

{I5⇒I3I2I8 ，I3I2⇒I8I5，I2I8⇒I3I5} (2)

For instance, the rule 'I5⇒I3I2I8', 'I5I3⇒I2I8' can be obtained via moving I3 to head; and 'I3I8I5⇒I2' can be obtained via moving I3,I8. So, it is not difficult to find that all the rules in (1) can be deduced by its subset (2).That indicates that no information will be lost in the moving process, and

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

any rule in (1) is implied in the subset (key rules set).

SAS (Subset of the Antecedent Set) algorithm introduced in this paper is to reduce the number of association rules without any data loss, so as to save more storage consumption and time for analysis. With the advancement in networking and multimedia technologies enables the distribution and sharing of multimedia content widely. In the meantime, piracy becomes increasingly rampant as the customers can easily duplicate and redistribute the received multimedia content to a large audience. Insuring the copyrighted multimedia content is appropriately used has become increasingly critical. Although encryption can provide multimedia content with the desired security during transmission, once a piece of digital content is decrypted, the dishonest customer can redistribute it arbitrarily[2, 3].

2. CONCEPTS AND THEOREMS

Some new concepts, theorems and symbols are introduced here to make it easy to explain the new algorithm, besides, we also inherit some concepts in GRSET algorithm in paper[1].

Definition 1 The support of itemset A is denoted as sup (a), which means the number of transactions that include the itemset. Take quasi association rule [4] c⇒ (l-c) as an example, l is a frequent itemset, and c is the subset of l.

Definition 2 If quasi association rules c⇒ (1-c) meets the condition of the min-confidence.

Theorem 1 If itemset C is the subset of itemset D, then sup(c)⩾sup(d).

Theorem 2 If quasi association rule c⇒ (l-c) meets the condition of min-confidence (mc), then all the quasi association rules that shortest antecedent include C and be the subset of L meets the condition of mc.

Proof: Let c be a proper subset d, and d is the subset of l, if c⇒ (l-c) is true, then association rule d⇒（l-d）is true.

∵c⇒（l-c）is true, ∴sup(l)/sup(c) ⩾ mc, and because c is the proper subset of d, ∴sup（c）⩾ sup（d）, ∴ sup(l)/sup(d) ⩾ mc, ∴ quasi association rule d⇒（l-d）is true too. QED.

Therefore, by the theorem 2, given a frequent itemset l, if one of its proper subset c, as a shortest antecedent, meets the condition of association rule c⇒ (l-c), then for any itemset d, which is the subset of l and includes c, as a shortest antecedent, meets the condition of association rule d⇒ (l-d).

By the property of Theorem 1 and 2, given an association rule, if move some tail-items to shortest antecedents, then the new quasi association rules meet the condition of mc. For example, let “abcde” be a frequent itemset, if ab⇒cde meets the condition of mc, and move the tail-item c to head, then the new association rule abc⇒de meet the condition of mc as well.

Given a frequent itemset l, x denotes the shortest antecedent of quasi association rules that generated by l, if quasi association rule x⇒ (l-x) is tenable, then sup(l)/sup(x) ⩾ mc, namely sup(x) ⩽ sup(l)/mc. According to the analysis above, the problem to find the shortest antecedent that makes the association rules which generated by frequent itemset l tenable, in fact, is the problem to find the subset of item set（ sub-item set for short） whose support is less than or equal to sup(l)/mc according to theorem 1 and 2. Therefore, in the process of finding the sub-itemset, if the support of a certain sub-itemset less than or equal to sup (l)/mc, then it is not necessary to judge whether the support of the superset of sub-itemset meets the condition. However, if the support of the sub-itemset is bigger than sup (l)/mc, it is still necessary to judge the support of the superset of the sub-itemset by depth first algorithm.

Example 1: in table 1, I3I2I8I5 is a frequent itemset, and its support is 9, because mc=80%, the condition meet x⇒I3I2I5I8-x is just sup(x) ⩽ sup(l)/mc, and sup(l)/mc = 9/80% = 11.5. So sup(x) ⩽11.5 is necessary and sufficient condition in this case.

The SAS algorithm in this paper is based on the theorem 1 and 2, and its main idea is: firstly, generates the subset of the shortest antecedent set that makes quasi association rules that generated by frequent itemset l tenable; and secondly, adds the subset of the shortest antecedent set to the Subset of the Antecedent Set(AS).

3. GENERATE SUBSET OF THE ANTECEDENT SET ALGORITHM

SAS is acronyms of Generate subset of the Antecedent Set, and it is a recursive algorithm for the function of generating a minimal expression-a subset of the antecedent set.

Algorithm 1 SAS

Journal of Theoretical and Applied Information Technology
28th February 2013. Vol. 48 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Input: frequent itemset l, min-confidence mc;

Output: subset of the Antecedent Set (AS) of association rules

Method:

（1） m=||l||

（2） AS=∅;

（3） get AS（∧，1）;

procedure getAS(h，s)

Parameters:

H: substring of l, the quasi association rule with the shortest antecedent h, and the rule meets the condition of mc.

S: the beginning location to be connected in l, and p (l, h[||h||])+1≤ s≤ m.

Method:

（1） n=||h||;

（2） if n < m-1;

（3） if sup (h+l[s]) ≤ sup (l)/mc

（4） if (! (Superset Check (AS, h+l[s]) then

（5） AS=AS ∪ {h+l[s]}

（6） else AS=AS ∪ {SuperSet of h+l[s]}-{h+l[s] }

（7） else {

（8） j=s,

（9） h=h+l[s]

（10） do

（11） j=j+1

（12） while sup (h+l[j]) > sup (l)/mc

（13） if (! (Superset Check (AS, h+l[j]))

（14） AS=AS ∪ {h+l[s]}

（15） else AS=AS ∪ {SuperSet of h+l[s]}-{h+l[s]

（16） }

Procedure supsetcheck(AS, a)

Parameters:

AS: subset of the Antecedent Set

A: an itemset found just recently

Method:

(1) If Superset (a)∈AS

(2) {

(3) Sus=Superset (a);

(4) Prune (sus) according to Theorem 1, 2;

(5) Return 1;

(6) }

(7) Else

(8) Return 0

Explanation: In the line 4,5,6,13,14,15 of algorithm SAS, the Superset Check(AS, a) is run because if the sub-itemset in a certain recursive procedure meets the condition of sup(x) ≤sup(l)/mc, it needs to judge whether AS has superset, if it is true, then the superset will be pruned according to Theorem 1 and 2, and at the same time, adds the new shortest antecedent to the set AS, and a new set AS will be obtained.

4. EXAMPLE AND ANALYSIS

Example 2 :

Given a data set, table 1,

Table 1: Transactional Data Set

TID	Item
T1	I1I2I3I4I5I7I8
T2	I2I3I5I6I7I8
T3	I3I4I6
T4	I1I2I3I4I5I6I7I8
T5	I1I2I3I5I6I7I8
T6	I1I3I4I5I7I8
T7	I2I3I4I6I7I8
T8	I1I2I3I5I6I8
T9	I1I2I3I4I5I6I7I8
T10	I3I5I7I8
T11	I1I2I4I6
T12	I1I2I4I6
T13	I3I4I7
T14	I1I2I3I4I5I6I7I8
T15	I2I3I4I5I6I7I8
T16	I2I3I5I6I8

Let ms=50%，mc=80%. Sup (I3I2I8I5) =9.

Question: output the minimal expression of association rules generated by frequent item sets f=I3I2I8I5.

The mining process is as follows：

m=4

AS=Φ

Sup(f)/mc=11

Sup(I3)=14 > 11

Sup(I3I2)=10 < 11

Journal of Theoretical and Applied Information Technology
28th February 2013. Vol. 48 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AS= AS ∪ {I3I2}={I3,I2}
Sup(I3I8)=12 > 11
Sup(I3I8I5)=11 ⩽ 11
AS= AS ∪ { I3I8I5}={I3I2,I3I8I5}
Sup(I3I5)=11 ⩽ 11
AS= AS ∪ { I3I5}-{I3I8I5}={I3I2,I3I5}
Sup(I2)=12 > 11
Sup(I2I8)=10 ⩽ 11
AS= AS ∪ { I2I8}={I3I2,I3I5,I2I8}
Sup(I2I5)=9 < 11
AS= AS ∪ { I2I5}={I3I2,I3I5,I2I8,I2I5}
Sup(I8)=12 > 11
Sup(I8I5)=11 ⩽11
AS= AS ∪ { I8I5}={I3I2,I3I5,I2I8,I2I5,I8I5}
Sup(I5)=11 ⩽ 11
AS=AS∪{I5}-{I3I5,I2I5,I8I5} ={I3I2,I2I8,I5}
According to SAS, the subset {I3I2,I2I8,I5} of shortest antecedent set of association rule can be generated via the frequent itemset I3I2I8I5, and the association rules with the shortest antecedents in {I3I2,I2I8,I5} are { I3I2⇒I8I5， I2I8⇒I3I5 ， I5⇒I3I2I8}.

Algorithm analysis

In example 2, in regard to the frequent itemset {I3I2I8I5}, if by GRSET algorithm, then all the 10 association rules should be generated, namely {I5⇒I3I2I8 ， I3I2⇒I5I8 ， I3I5⇒I2I8 ， I2I8⇒I3I5，I2I5⇒I3I8， I8I5⇒I3I2，I3I2I8⇒I5 ，I3I2I5⇒I8，I3I8I5⇒I2，I2I8I5⇒I3}.

So as to the whole database, the number of association rules will be huge, and the consumption of time, space and power will be enormous.

However, the SAS algorithm here just generates 3 rules with shortest antecedent in set {I3I2，I2I8 ，I5}, the rules are

{I3I2⇒I8I5，I2I8⇒I3I5 ，I5⇒I3I2I8},

and other 7 rules generated by I3I2I8I5 in GRSET can be obtained by moving tail-items to shortest antecedents via the 3 rules above. For example, 2 rules {I3I2I8⇒I5，I3I2I5⇒I8} can be obtained by tail-to-head method based on rule I3I2⇒I8I5; and {I2I8I3⇒I5, I2I8I5⇒I3} can be deduced by I2I8⇒I3I5; and rules {I5I3⇒I2I8 ， I5I2⇒I3I8 ， I5I8⇒I3I2 ， I5I3I2⇒I8，I5I3I8⇒I2，I5I2I8⇒I3} can also be deduced by I5⇒I3I2I8.

So it is not difficult to find that the 3 key rules {I3I2⇒I8I5 ， I2I8⇒I3I5 ， I5⇒I3I2I8} implies totally all other 7 rules without any data loss.

A Hasse diagram, Figure 1, is able to indicate the structure of Boolean association rules set[5] that generated by the frequent itemset I3I2I8I5, the first line in brackets indicates the shortest antecedents of association rules, and the second line in brackets means the tail-items of association rules. It is not difficult to get a conclusion that the association rules with shortest antecedents {I3I2， I2I8， I5} constitute the maximal element in Figure 1.

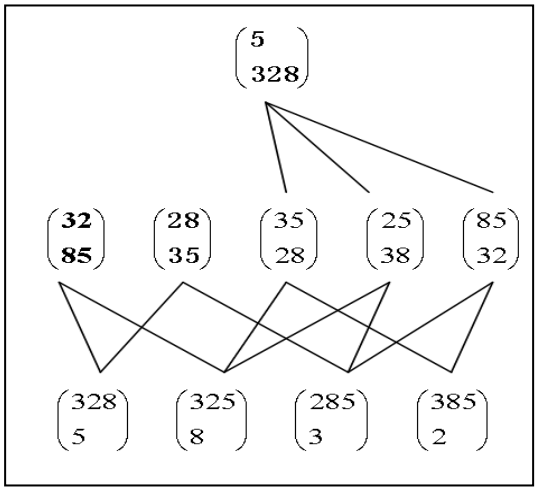

Figure 1. Hasse Diagram

5. PERFORMANCE ANALYSIS

The performance of the SAS algorithm was tested comparing with the GREST and Apriori algorithm with the same data set. The algorithm is implemented by Java, and the running environment mainly includes: a CPU of P4 2.0GHz, storage memory of 1.0G and the Windows XP system. We have tested the number of rules that generated via GRSET algorithm and SAS algorithm respectively, the data set used in this case are shown in table 1.
Let ms=50% ,and mc=80%, then Table1 and Figure 2 show the number of rules in GRSET and SAS algorithm for the data in Table 2.

Table 2: Number Of Association Rules

	Apriori	GRSET	SAS
Number of Rules	258	140	86

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

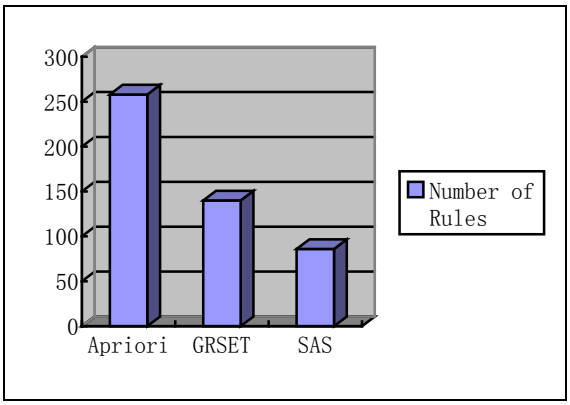

Figure 2. Number of Association Rules

As it shown from the experimental results in Table 2 and Figure 2, for the same dataset and the minimum support threshold, the number of the key rules generated by SAS algorithm are much less than the rules that generated by GRSET algorithm, so the storage usage are decreased than that of GRSET, the SAS algorithm almost saves 40% storage consumption compared with the GRSET, and saves 66% space to Apriori. Therefore, the experiment indicates that the SAS algorithm is more effective in saving storage consumption than the GRSET and Apriori algorithm.

6. CONLUSIONS

In this paper, a new algorithm named SAS is introduced; it only generates some key rules of association rules comparing with GRSET algorithm. It is shown on theory and experiments that SAS has three advantages. Firstly, the key rules generated by SAS are much less than that generated by GRSET, it can save more space consumption for user. Secondly, the subset built by key rules is a maximal element of all association rules which generated by the same frequent itemset, so it implies all the rules. Thirdly, the key rules can deduce all the association rules base on the same frequent itemset without any information loss.

Because the property of frequent closed itemset is helpful for reducing the number of redundant rules in the process of generating the association rules, the studies on generating association rules based on frequent closed itemset will be carried out.

ACKNOWLEDGEMENTS

The authors are grateful to the referee for their valuable comments and suggestions. This work has been supported by the Supported by the Fund of Program for Tackling Key Problems in Science and Technology of the Department of Science and Technology of Henan Government. (Grant No. 12A510001).

REFERENCES:

[1] R Agrawal , Mannila H , Srikant R , et al, “Fast discovery rules”. Fayyad U. Advances in Knowledge Discovery and Data Mining. Menlo Park: AAA I Press, February 25-28,1996. pp. 307-328.

[2] Cercone V, Tsuchiya M. Luesy, “editor’s introduction”, *IEEE Transaction on Knowledge and Data Engineering*, Vol. 5, No. 6, 1993, pp.901-902.

[3] Brin S , Motwani R , Ullman J, etal “Dynamic itemset counting and implication rules for market basket data”. http:// citeseer . njnec.com/ brin97dynamic html, 1999-05-23/2002-05-09 .

[4] Kun Wu, Baoqing Jiang, Qing Wei, “A Depth-First Algorithm of Finding All Association Rules Generated By A Frequent Itemset”, *Proceedings of 2006 International Conference on Intelligent Systems And Knowledge Engineering*, April 6-7,2006, pp.102-109.

[5] Jiang Baoqing, Li Jian, Xu Yang, “A Structure of Boolean Association Rules Set”, *Journals of Henan University*, Vol. 36, No. 1, 2006, pp. 88-90.

[6] Pasquier N, Bastide Y, Taouil R , etal “Discovering frequent closed itmesets for association rules”. http: // citeseer. Njnec.com / pasquier99 discovering html, 1999-11-23/2002-04-18.

[7] Y.-J. Yan, Z.-J. Li, and H.-W. Chen, “Frequent item sets mining algorithms”, *Computer Science*, Vol. 1, No. 5, 2004, pp.112-114.

[8] Das Nandinia, Ghosh Avishekb, Das Prasun. “Mining association rules to evaluate consumer perception: A new FP-tree”, *International Journal for Quality Research*, Vol. 5, No. 2, 2011, pp.89-102.

[9] M. Hahsler, K. Hornik, New “probabilistic interest measures for association rules”, *Intelligent Data Analysis: an International Journal*, Vol.11, No.5, 2007, pp.437-455.

[10] Yin-Fu Huang, Chieh-Ming Wu, “Preknowledge-based generalized association rules mining”, *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology*, Vol.22, No.1, 2011, pp.1-13.