A new incremental updating algorithm for association rules*

WANG Zuo-cheng [1], XUE Li-xia [2]

(1. Software Institute, Chongqing University of Posts and Telecommunications, Chongqing 400065, P. R. China;
2. College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, P. R. China)

Abstract: Incremental data mining is an attractive goal for many kinds of mining in large databases or data warehouses. A new incremental updating algorithm rule growing algorithm (RGA) is presented for efficient maintenance discovered association rules when new transaction data is added to a transaction database. The algorithm RGA makes use of previous association rules as seed rules. By RGA, the seed rules whether are strong or not can be confirmed without scanning all the transaction DB in most cases. If the distributing of item of transaction DB is not uniform, the inflexion of robustness curve comes very quickly, and RGA gets great efficiency, saving lots of time for I/O. Experiments validate the algorithm and the test results showed that this algorithm is efficient.

Key words: assocation rules; incremental updating; aprion growing
CLC number: TP311 **Document code**: A **Article ID**: 1673-825X(2007)03-0309-05

1 Introduction

Discovery of association rules is an important data mining task, several algorithms have been proposed to solve this problem, Apriori is the ancestor[1]. The most important step in mining association is the Generation of frequent itemsets. Most of these algorithms require repeated passes over the database, which incurs huge I/O overhead and high synchronization expense in parallel cases[2-5]. So algorithms which are trying to reduce costs are expected. Several algorithms [6-8] have been proposed in the literature to solve this problem; most of them are based on the Apriori approach. In this paper, we propose another algorithm of association rule data mining—RGA. The seed rule grows in database until the strong threshold. Association rules that satisfy a user-specified minimum robustness threshold are referred to as strong association rules and are considered interesting. The key points are lies: finding the seed rule, selecting where inseminating the seed rules, how to bring up the rule, when to stop the rule growth and get the robust rule.

2 Looking for seed rules

For a given set of task-relevant data, the data mining process may discover thousands of rules, many of which are uninteresting to the user. Generally, we can give some constraints to data mining, which named constraint-based mining. The constraints include the following: knowledge type constraints, data constraints, dimension/level constraints, interesting constraints, rule constraints etc. In order to looking for the seed rules, we can make use of the rule constraints. The rule constraints mining allows users to specify the rules to be mined according to their intention, thereby making the data mining process more effective. In addition, a sophisticated mining query optimizer can be used to exploit the constraints specified by the user. We call the rule constraints metarules, it allow users to specify the syntactic form of rules that they are interested in mining. Metarules may be based on the analyst's experience, expectations, or intuition regarding the data, or automatically generated based on the database schema.

The following example, market basket analysis illustrates how to finding the seed rules making use of the metarules. Let's look at AllFruits database:

AllFruits(*transaction_id*, *item*1, *item*2, *item*3, *item*4, *item*5)

The metarule can be used to specify this information describing the form of rules you are interested in finding. An example of such a metarule is

$$A(X,Y) \rightarrow B(X,W)$$

Where A and B are predicate variables that are instantiated to attributes from the given database during the mining process, X is a variable representing a customer, Y and W take on values of the attributes assigned to A and B, respectively. Typically, a user will specify a list of attributes to be considered for instantiation with A and B. Otherwise, a default set may be used.

In all the transactions, the maximum item is five. If there are rule constraints: the rule antecedent Y is *i*-

* **Received date**: 2007-02-10
Foundation item: The work is supported by Chongqing Municipal Education Commission Science Research Program (No. KJ060511)

$item_1$. The metarule may allow the generation of association rules like the following:

$\text{buys}(X, item_1) \rightarrow \text{buys}(X, item_2)$
$\text{buys}(X, item_1) \rightarrow \text{buys}(X, item_3)$
$\text{buys}(X, item_1) \rightarrow \text{buys}(X, item_4)$
$\text{buys}(X, item_1) \rightarrow \text{buys}(X, item_5)$

So the four rules can be the seed rules. General speaking, according to the DMQL, we can always find the seed rules.

3 Inseminating the seed rules

3.1 Asymmetry of transaction database

Let $I=\{i_1, i_2, \cdots, i_m\}$ be a set of items. Let $D=\{d_1, d_2, \cdots, d_n\}$ be a collection of transactions, where each transaction d has a unique identifier and contains a set of items such that $d \subseteq I$, $d_i = \{item_1, item_2, \cdots, item_m\}$. Giving enough d, the appearance of an item in the DB obeys the uniformity distributing. Supposing $P_i \subseteq D$, is the subset of D, probability of appearance of an item in P_i, denoted by Pro_item_i,

$$Pro\_item_i = \frac{count\_item}{count\_P_i} \tag{1}$$

$Count\_item$ is the count of one special item in P_i, $count\_p_i$ is count of D in P_i.

In the example of market basket analysis, total 10 000 transactions. We divide it into 20 subsets, and count the Pro_item_i respectively, as shown in Fig. 1, The Pro_item_i of an item in P_i obey normal distributing, as shown in Fig. 2.

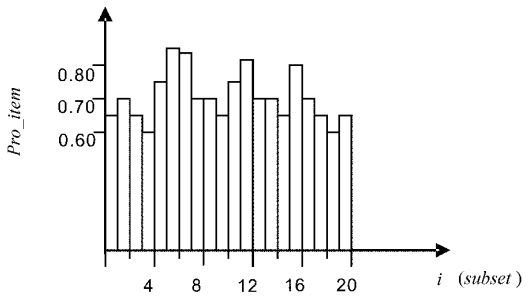

Fig. 1 *Pro_item* in subsets

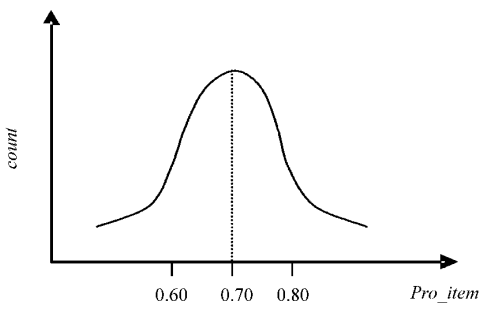

Fig. 2 Distribution of *pro_item*

The distributing predicates that the average Pro_item is 0.70, i. e. the probability of appearance of this item is 0.70 in the transaction DB. The peak value is 0.85, and the lowest value is 0.6. It implies that the distributing of item is not uniformity in the transaction DB. Sometimes a large number of this item is sold, but sometimes only a few. In fact, because of season, weather or sales promotion and so on, the distribution of item is not uniformity in the whole transaction DB, and it is the basis of RGA.

3.2 Significance of appearance probability of rule antecedent

Each discovered pattern should have a measure of certainty associated with it that assesses the validity or "trustworthiness" of the pattern. A certainty measure for association rules of the form "$A \rightarrow B(C\%)$", where A and B are sets of items, C is confidence. Given a set of task-relevant data tuples(or transactions in a transaction database) the confidence of "$A \rightarrow B$" is defined as:

confidence

$$(A \rightarrow B) = \frac{\#\_tuples\_containing\_both\_A\_and\_B}{\#\_tuples\_containing\_A} \tag{2}$$

In fact, the confidence can be converted into conditional probability $P(B|A)$,

$$P(B \mid A) = \frac{P(AB)}{P(A)} \tag{3}$$

The concept of confidence indicates that the appearance probability of rule antecedent influences the interesting of rule obviously, so the growing point can be found according to the distributing of rule antecedent.

3.3 Selecting growing point

First of all, we propose one theorem.

Theorem1: if $P(A)$ tends to large, then $P(AB)$ tend to large too.

Observing the distributing of Pro_item, Fig. 1, we can find that Pro_item_6 is the maximum value; the next is Pro_item_7, and then Pro_item_{12}, Pro_item_{16}. According to the theorem1, we select the subset that the maximum of Pro_item belong, that is 6-th subset, P_6; the next is P_7, and then P_{12}, P_{16}. if the seed rule is not still robust after growing in these subsets, the other subsets, such as P_5, P_{11}, etc. must be considered.

4 Bringing up seed rule

In the example of market basket analysis, the seed rules is $\text{buys}(X, item_1) \rightarrow \text{buys}(X, item_2)$, $\text{buys}(X, item_1) \rightarrow \text{buys}(X, item_3)$, $\text{buys}(X, item_1) \rightarrow \text{buys}(X, item_4)$, $\text{buys}(X, item_1) \rightarrow \text{buys}(X, item_5)$. For example, we bring up the first seed rule.

Supposing C_{item_1} denotes the count of tuples containing $item_1$, $C_{item_1 \& item_2}$ denotes the count of tuples containing both item_1 and item_2. $P_i \subseteq D$, $D=\{d_1, d_2, \cdots,$

$d_n\}$ is a collection of transactions, C denotes the count of tuples scaned. First scanning P_6, the next is P_7, and then P_{12}, P_{16}. The process of scanning P_6 as follow:

```
Begin
C=0, C_item1=0, C_item1&item2=0;
DBSize=COUNT (P6);
for (i=1;i<=DBSize;i++){
  if (di containing item1)
    C_item1++;
  else if(di containing both item1 and item2)
    C_item1&item2++;
C++;
}
Return C, C_item1, C_item1&item2;
```

The process will be lasting until the seed rule maturing. And we get the table containing C, C_{item_1}, $C_{item_1 \& item_2}$, showed in Tab. 1.

Tab. 1 Count in the process of seed rule growing

No_subset	*C*	C_{item_1}	$C_{item_1 \& item_2}$	*Support*	*robustness*
6	100	85	70	0.70	0.70
7	100	80	68	0.68	0.69
12	100	76	64	0.64	0.67
16	100	73	56	0.56	0.65
…	…	…	…	…	…

5 Getting mature and robust rule

The seed rule is growing up while the tuples in DB is being scanned and counted. Then, how to recognize the seed rule is mature, and is robust or not. If we consider a seed rule is mature, we can stop the process of rule growing, and then check robustness of the rule.

5.1 Interesting measure of rule

There are many kinds of ways to measure the interesting of rule. We introduce two interesting measures to RGA, the one is support of pattern in subset, according with user-specified maximum support degression threshold, the other is robustness of pattern, according with user-specified minimum robustness threshold.

1) Maturity measure of rule

The support of pattern measures the potential usefulness of pattern. For association rules of the form "$A \rightarrow B$" where A and B are sets of items, it is defined as:

Support $(A \rightarrow B) =$

$$\frac{\#\_tuples\_containing\_both\_A\_and\_B\_in\_subset}{total\_\#\_of\_tuples\_in\_subset} \quad (4)$$

In chapter 3.1, we divide the transaction DB into many subsets, the numerator and denominator is counted in subset, of course the Support is rule Support in subset.

2) Robustness measure of mature rule

We propose another interesting measure—robustness, the robustness of "$A \rightarrow B$" is defined as

robustness $(A \rightarrow B) = \dfrac{\#\_tuples\_containing\_both\_A\_and\_B}{\#\_tuples\_scaned}$ (5)

$\#\_tuples\_scaned$ denotes the count of tuples that have scanned during the process of rule growing. So the robustness is a dynamic indicator, it indicates the seed rule growing process.

5.2 Getting the mature and robust rule

In chapter 3.1, we divide the transaction DB into many subsets, Support is counted in each subset, shown in Tab1. Because the scanning order in the transaction DB is corresponding to decline order of the appearance probability of rule antecedent, the Support of subset is decline, shown in Fig. 3 (a). We propose two user-specified measures to decide whether the seed rule is mature or not. One is maximum support degression threshold, if the degression rate exceeds the maximum support degression threshold, we consider the seed rule is mature, such as point i_1 in Fig. 3(a). If the degression rate does not exceed the maximum support degression threshold during scanning in DB, we adopt the second measure—Number of minimum scanning tuples, in case of number of scanned tuples exceeds the number of minimum scanning tuples, we consider the seed rule is mature, too, such as point i_2 in Fig. 3(b).

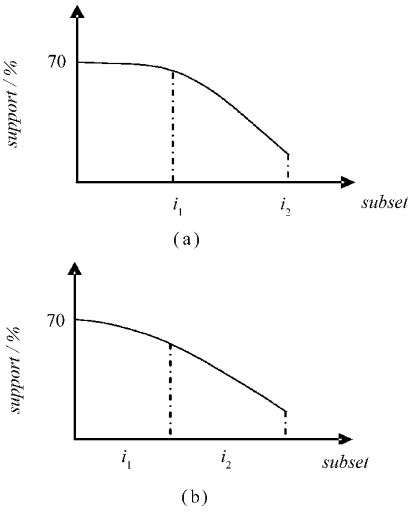

Fig. 3 *Support* in process of rule growing

When the seed rule is mature, we must decide it is robust or not. During the process of rule growing, robustness is descending, shown in Fig. 4. When the seed rule is mature, if robustness of the seed rule exceeds user-specified minimum robustness threshold, we consider the seed rule is robust or strong and considered interesting. Seed rule with low support likely

represent noise, or rare or exceptional cases.

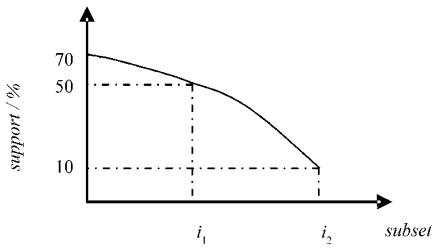

Fig. 4 *Robustness* in process of rule growing

The curve of support indicates that the support decreases very slowly before i_1. we can see that the seed rule is growing in the subsets whose Pro_item is higher than the average value. After i_2, the support decreases rapidly, and faster and faster.

If we finished scanning the transaction DB and get the whole robustness curve, just like Fig. 5, we can find that the minimum robustness, 10%, is the support in whole transaction DB.

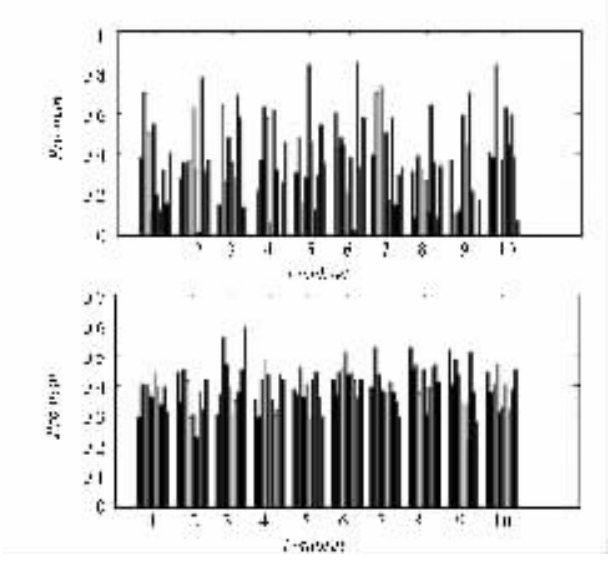

Fig. 5 *Pro_Milk* in subsets of DB_1 and DB_2

We can confirm metarules are strong rules or not, and need not scan all the transaction DB in most case. If the distributing of item of transaction DB is not uniform, the i_1 comes very quickly, and RGA gets great efficiency, saving lots of time for I/O. the worst situation is that the tuples which containing rule antecedent and consequent are centralization, but the tuples which containing rule antecedent are decentralization, we must scan almost all the transaction DB in order to confirming the metarule is strong or not.

We use a measure to denote the efficiency of RGA, denoted by Pscan:

$$Pscan = \frac{\#\_tuples\_scaned}{\#\_tuples\_DB} \qquad (6)$$

$\#\_tuples\_scaned$ denote the count of tuples which be scaned during seed rule process. $\#\_tuples\_DB$ is the count of total tuples in transaction DB.

6 Example

Some experiments have been carried out to evaluate the performance of RGA. All the experiments were performed on a Pentium 586 personal computer running windows 98 with main memory of 32 MB. We applied the algorithms to two relational databases, the two DB contain 10 000 transaction records of two small supermarkets respectively. Most of the transaction records have five items or so. The supermarket manager want to know which items are frequently purchased together with milk by customers.

According to RGA, we divide each DB into 100 subsets. *Pro_Milk* of item milk shown as Fig. 5. The *Pro_Milk*$_i$ of in P_i obey normal distributing, but the variance σ^2 is not the same, $\sigma_1^2 = 0.05$, $\sigma_2^2 = 0.005$, the distribution was shown in Fig. 6. Directed by the metarule of "buys(X,*item*)→ buys(X,*item*)", we bring up the seed rule "buys(X,*milk*)→ buys(X,*bread*)"in the two transaction DB. The curve of support is shown in Fig. 7.

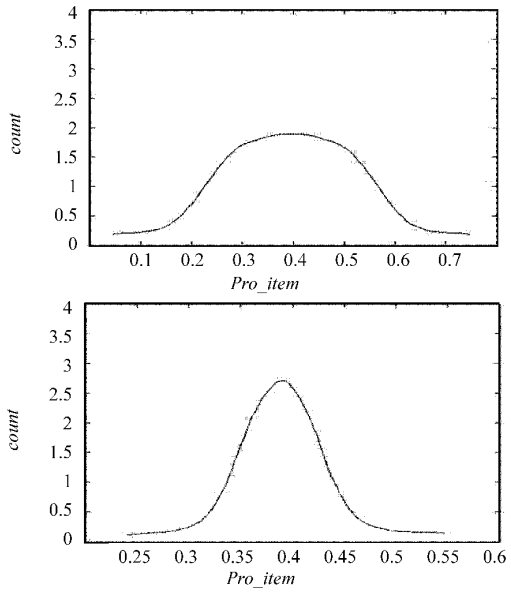

Fig. 6 Distribution of *pro_item* of DB_1 and *Pro-item* of DB_2

We can see that in DB_1, the curve of support exists a turning point at the 300th subset or so, and the degression rate exceed the user-specified minimum support threshold, 10%, so we consider the seed rule is mature, and the robustness is 96%, exceeding user-specified minimum robustness threshold, we consider the seed rule is robust or strong and considered interesting.

In DB_2, the curve of support is descending uniformly, and the degression rate does not exceed the user-specified minimum support threshold, 10%, so we must adopt the second measure—Number of minimum scanning tuples, user-specified 7 000, when the 7 000th tu-

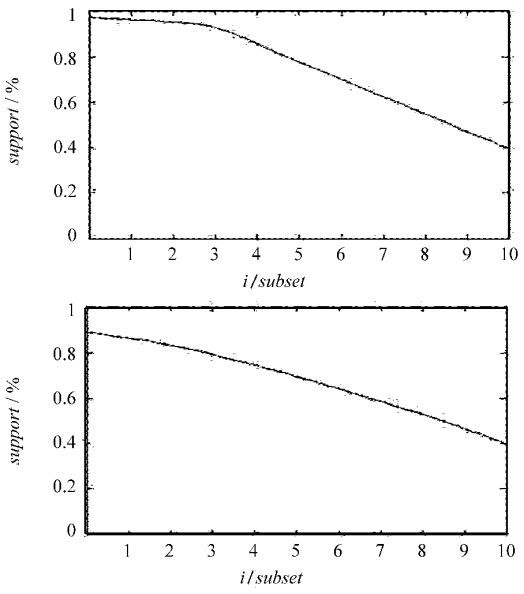

Fig. 7 *Support* in process of rule growing in DB_1 and DB_2

ple is be scanned, the robustness of seed rule is 73%, satisfying user-specified minimum robustness threshold, we consider the seed rule is robust or strong and considered interesting too.

In the two DB, the efficiency of RGA are counted, $P_{scan_1}=0.3$, $P_{scan_2}=0.7$. In DB_1, the rule antecedent and consequent are centralized, so we get the high efficiency. In DB_2, the rule antecedent and consequent are decentralized, in order to get the mature seed rule, large user-specified number of minimum scanning tuples leads too many tuples be scanned, and decrease the efficiency.

7 Conclusion

We have proposed new algorithms for efficient mining of association rules, different from all existing algorithms. We introduce a concept of rule growing, consider that the seed rule is growing in database, when the seed rule is mature and robustness, the rule is considered strong rule and interesting. In order to let the seed rule become mature as soon as possible, we analyze the DB at first, divide the DB into many subsets, select subset whose *Pro_item* is the largest in the subsets as the jumping-off point. When the seed rule is growing in DB, we propose two measures to decide the seed rule is mature and robustness or not, *support* and *robustness*. If *robustness* of the mature seed rule exceeds user-specified minimum robustness threshold, we consider the seed rule is robust or strong and considered interesting.

Our algorithm has several advantages. First, the I/O costs are quite limited because after analyzing the DB, only part of DB would be scanned, and high efficiency would be gotten when the rule antecedent and consequent are centralized. Second, RGA can make use of the metarule efficiently; it mines the association rules based on metaerule. Third, our algorithm can be used in incremental mining, suppose that we have gotten the association rules from current DB, when a set of new tuples, △DB, is inserted into the database, RGA can bring up the association rules that have been gotten, and then check these rules is robustness or not. To confirm that, we have done the experiments to validate the algorithm, the test results show that our algorithm is efficient.

References:

[1] HAN J. KAMBER M. Data Mining: Concepts and Techniques[M]. U S: Morgan Kaufmann Publishers, 2000.

[2] CHEN M S, HAN J, YU P S. Data mining: an overview from a database perspective[J]. IEEE Trans. Knowledge Data Eng. 1996, 8 (6): 866-883.

[3] HAN J, CAI Y, CERCONE N. Data-driven discovery of quantitative rules in relational databases[J]. IEEE Trans. Knowledge Data Eng. 1993, 5(1): 29-40.

[4] RAJAMANI K, COX A, IYER B, et al. Efficient mining for association rules with relational database systems[C/BL]. Proceedings of the IEEE International Symposium on Database Engineering and Applications, (1999-02-20) [2007-01-20]. http://citeseer. ist. psu. edu/rajamani99efficient. html.

[5] TSAI P S M, CHEN C M. Discovering knowledge from large databases using prestored information[J]. Inform. Systems, 2001, 26 (1): 3-16.

[6] AGRAWAL R, IMIELINSKI T, SWAMI A. Mining Association Rules Between Sets of Items in Large Databases [C/OL]. SIGMOD-93, (1993-05-23) [2007-01-10]. http://rakesh. agrawal-family. com/papers/sigmod93assoc. pdf.

[7] AGRAWAL R, SRIKANT R. Fast Algorithms for Mining Association Rules [C]. Proceedings of the 20th VLDB Conference, Santiago, Chile: [s. n.]. 1994: 487-499.

[8] SAVASERE A, OMIECINSKI E, NAVATHE S. An efficient algorithm for mining association rules in large databases[C]. Proceedings of the 21st VLDB Conference, Zurich, Switzerland: [s. n.]. 1995: 432-44.

Biographies:

WANG Zuo-cheng(1973-), Bazhong, Sichuan province, Vice-professor of Software Institute, Chongqing University of Post and Telecommunication. Major in Software Engineering, GIS and Digital image processing.

XUE Li-xia(1976-), Xichang, Sichuan province, Teacher of College of Computer Science and Technology, Chongqing University of Post and Telecommunication. Major in GIS, digital image processing.