A Novel Algorithm for Mining Rare-Utility Itemsets in a Multi-Database Environment

Guo-Cheng Lan[a], Tzung-Pei Hong[b,c], and Vincent S. Tseng[a]
[a]Department of Computer Science and Information Engineering
National Cheng Kung University, Taiwan, R. O. C.
[b]Department of Computer Science and Information Engineering
National University of Kaohsiung, Taiwan, R. O. C.
[c]Department of Computer Science and Engineering
National Sun Yat-sen University, Taiwan, R. O. C.
Email: rrfoheiay@idb.csie.ncku.edu.tw, tphong@nuk.edu.tw, tsengsm@mail.ncku.edu.tw

Abstract

Utility mining has recently been an emerging topic in the field of data mining. It finds out high-utility itemsets by considering both the important factors of profit and quantity. In some situations, rarely occurring items may co-occur in a relatively close relationship with specific high-utility items. These utility itemsets with rare items may provide useful information to decision makers as well. Most of the existing methods on utility mining were designed for a single database (centralized database) and not suitable for the environment with multiple data sources such as those in a chain-store enterprise. Moreover, the existing methods did not consider the existing periods and the existing branches of items. In this paper, we have thus proposed a new kind of patterns, named Rare Utility Itemsets, which consider not only individual profits and quantities but also common existing periods and branches of items in a multi-database environment. We have also proposed a new mining approach called TP-RUI-MD (Two-Phase Algorithm for Mining Rare Utility Itemsets in Multiple Databases) to efficiently discover rare utility itemsets. To our best knowledge, this is the first work on mining rare utility itemsets in a multi-database environment. At last, the proposed approach is shown to have good performance under a variety of system conditions through a series of experiments.

1 Introduction

In the fields of data mining, association rules [4] is the most frequently discussed issue due to its wide applications. In [3], Agrawal et al. first proposed the Apriori algorithm that is the most well-known algorithm for mining association rules from a transactional database. In many applications, however, the rare situations may be more significant. Take medical application for an example. The rare combination of symptoms can provide some useful insights for doctors. Take supermarket application for another example, some rare products more frequently occur with specific data in transactional databases. Such information can further provide the useful promotion strategies for decision makers. Therefore, an important issue, rare itemsets mining [26], is extended from the association rules discovery to solve the rare situation problems in various applications.

However, since both of the association rules and the rare itemsets models assume that the significance or profit of each product is the same, we cannot understand the represented significance of each product in a product combination. Moreover, the existing methods on both of the association rules discovery and the rare itemsets discovery may fail to discover product combinations which are composed of items with low frequency and high profit, or those with high frequency and low profit in a transactional database. Hence, frequency is not sufficient to answer a product combination whether it is highly profitable or whether it has a strong impact.

To solve the problems above, Chan et al. [7] proposed a new topic, named utility mining, which its main objective is to discover high utility itemsets from a transactional database. A high utility itemset on utility mining considers both the individual profit of each product (item) in a database and the bought quantity of each one in a transaction simultaneously. Thus each product can represent practical utility value in a product combination. Subsequently, the focus of related studies [10][16][24][25] is how to increase the

efficiency in terms of discovering the high utility itemsets from the databases. However, not all items are existed all periods or all branches in the database. Take chain-store enterprise for an example. A product may be put on-shelf and taken off-shelf multiple times or that one is only sold in some stores in a chain-store enterprise. Hence, the base of existing methods in computing utility value of a product set is throughput a database so the results discovered by these methods may be biased in the aspect of multiple databases.

For the reasons above, we proposed a new kind of patterns, named rare-utility itemsets, which take both of the existing periods and the existing branches into consideration in a multi-database environment. For example, a rule states:" it is less frequent for consumers to buy food processors or cooking pans in supermarkets near the city areas than to buy bread or milk, but the former transactions are more profitable for the chain-store enterprises". The rule may be not a frequent itemset, but it may be a high-utility itemset or a rare-utility itemset because most customers do not frequently buy these products. To discover significantly rare-utility itemsets in the utility mining model, we refer to the framework [26]. That is, our study sets and utilizes two minimum utility thresholds and one minimum relative PT utility confidence threshold to discover rare-utility itemsets in a multi-database environment. The reason is that the main purpose on utility mining is to find the most valuable customers (customers who contribute a major fraction of the profits for the business). These rare-utility itemsets which their utility values are too low are uninterested because decision makers of business think that the information of these itemsets do not gain the more profits to the businesses. Besides, since the downward closure property in the utility mining model does not exist, the Two-Phase algorithm [3] is proposed to solve the problem of mining high-utility itemsets. However, the existing algorithm, the Two-Phase algorithm, is not designed to discover rare-utility itemsets in a multi-database environment. Hence, we proposed a new mining algorithm named TP-RUI-MD (*Two-Phase Algorithm for Mining Rare Utility Itemsets in Multiple Databases*) to increase the efficiency in terms of finding proposed rare-utility itemsets in a multi-database environment. However, our proposed mining framework can not discover all rare-utility itemsets in a multi-database environment, but the TP-RUI-MD algorithm can still discover these rare-utility itemsets undiscovered by existing algorithms in a single database environment. The main contributions of this paper are listed as follows:

1. We have proposed a new itemset named rare-utility itemset that considers the common existing periods and existing branches in a multi-database environment.
2. We have proposed a novel algorithm named TP-RUI-MD to discover efficiently both of the rare-utility itemsets and the high-utility itemsets in a multi-database environment.
3. Detailed simulation experiments on a public data generator IBM were conducted to show the usefulness of proposed itemsets and the merits of proposed mining method in a multi-database environment.

The remaining parts of this paper are organized as follows. The related work is stated in Section 2. The problem definition is described in Section 3. The proposed approach TP-RUI-MD is described in Section 4. Experiments demonstrate the differences in number, relative utility ratio and relative PT utility confidence between original and proposed rare-utility itemsets by varying various parameters and the performance of proposed method TP-RUI-MD in dealing with large databases are described in Section 5. Conclusions and future work are given in Section 6.

2 Related Work

In the fields of data mining, association rules [4] is the most frequently discussed issue due to its wide applications. In reality, however, the rare situations may be significant information in many practical applications, such as healthy analysis, drug detection, medical application, etc. Hence, the important topic, rare itemsets mining [26], is proposed by Ha et al. In [26], the authors first proposed an algorithm RSA to solve the problem of rare itemsets mining. The RSA algorithm adopts the relative support criterion to discover the rules of significant rare items with specific items co-occurring in relatively high proportion in the databases. Subsequently, there are still many studies [1][9][12][20][21] to investigate the problem of rare itemsets mining.

However, since the frequency is not sufficient to measure the significance of association rules or rare itemsets, Cai et al. [6] proposed the weighted association rules method in 1998 and Agrawal et al. [19] proposed the quantitative association rules in 1996. On the other hand, temporal association rules mining [5][13][15][18] has been proposed to solve the dynamic association rules problem, but the discovered results may be incorrect because they consider the periods of transactions occurred but not the on-shelf periods of products. Besides, the on-shelf and off-shelf periods of products may

be switched multiple times in the database. Since previous data mining models are mostly based on a single database and both of the traditional association rules and the rare itemsets are not sufficient to provide knowledge inherent in data across the stores in a chain-store enterprise, it is necessary to develop data mining techniques on multiple databases [1][8][20][26][28]. Among these studies, Chen et al. [8] proposed a novel approach, named Apriori_TP, which is different from other approaches because the approach discovers temporal association rules across the stores in a multi-stores environment. In [8], the rules take both selling periods and selling stores of products into consideration simultaneously. Besides, the algorithm uses the common selling periods and stores of all products in a product combination as relative content of a product set to compute the relative support, and then discovers frequent relative itemsets in a multi-stores environment.

Nevertheless, the association rules on multiple database mining may fail to find some important rules containing high profit but lower frequency of items across stores. The reason is that the significances or profits of items are the same. In order to overcome the problems, Chan et al. [10] proposed an importance topic, named utility mining. The concept behind the utility mining is to discover the high utility itemsets whose utility values satisfy the minimum utility threshold from a transactional database. Utility mining model considers simultaneously both individual profit of each product in a database and bought quantity of each product in a transaction in the mining processes. In [25], the authors proposed the definitions of utility mining and theoretical model, named MEU. However, the theoretical model MEU has to examine complete sets of all items to find all high utility itemsets. Thus, the MEU method is not sufficient to solve the problem. Subsequently, Liu et al. [16] proposed a novel algorithm named Two-Phase to increase the performance in terms of finding high utility itemsets. However, since the Two-Phase algorithm is based on the Apriori algorithm [3], developing an efficient approach is crucial for utility mining. Thus the focus of subsequent studies [6][7][18][22] is how to improve the performance in terms of discovering the high utility itemsets from a transactional database. Similarly, the existing methods on utility mining are not suitable for the environment with multiple data sources because they do not consider the common selling periods of items across stores. Subsequently, the study [9] proposed the algorithm TRUI-Mine to discover the temporal rare utility itemsets from the transactional databases. However, the algorithm does not consider common selling periods and branches of all items in an itemset in a multi-database environment. Therefore, the TRUI-Mine algorithm does not be applied to discover rare utility itemsets in a multi-database environment.

As described above, there exists no work for discovering rare utility itemsets in a multi-database environment. This motivates our exploration of the issue of efficiently mining rare utility itemsets in a multi-database environment.

3 Problem Statement

1. The PT Table

In this study, we assume that the information about whether an item is on shelf of a branch within a period is known. A table, called the PT (Place and Time) Table [8], is thus used to keep this kind of information of an item. Take a chain-store enterprise as an example. Assume there are six stores (branch) and six time periods for sale. Table 1 shows the on-shelf information of a product A at each store in each period, where '1' represents "on shelf" and '0' represents "off shelf".

Table 1. The PT table of the product A.

	t_1	t_2	t_3	t_4	t_5	t_6
b_1	1	1	1	0	0	0
b_2	0	1	1	1	0	0
b_3	1	1	1	1	1	0
b_4	0	0	1	1	1	0
b_5	0	1	1	1	0	0
b_6	1	1	1	1	0	0

In real implementation, the PT table of an item is compressed into a compact format to increase the memory usage [8]. First, each row in a PT table (representing a branch) is represented as a binary string. For example, the bit string for the branch p_4 is (0, 0, 1, 1, 1, 0). It is then further encoded in the following way. If the first bit value in the string is '0', the number '1' is added to the encoded result; otherwise, nothing is done. Besides, when a different bit value is met, the position for the different value is recorded in the encoded result. The bit string (0, 0, 1, 1, 1, 0) for p_4 will thus be transformed into the string "136" according to the rule above. It means an off-shelf event occurs at t_1, an on-shelf event at t_3, and an off-shelf event again at t_6. In this way, each odd position in the encoded string represents the begin time of an off-shelf period, and each even position represents that of an on-shelf period. Take p_1 as another example in which the first bit is '1'. Its bit string is (1, 1, 1, 0, 0, 0). It is then encoded as the string "4", representing an off-shelf event occurs

in t_4.

It is also very easy to extend the PT table from an item to an itemset. The AND operation can be used to achieve the purpose. For example, assume the bit strings of the three products *A*, *B* and *C* at branch p_2 are (1, 1, 1, 0, 0, 0), (1, 1, 1, 1, 0, 0) and (0, 0, 1, 1, 1, 1), respectively. By the AND operation on them, the common selling periods for the itemset {*ABC*} on the shelf at branch p_2 is (0, 0, 1, 0, 0, 0), which is encoded as "134". It represents the time period for all the three items on the shelf is only t_3.

By the PT table of each item, the common on-shelf statement of all items in an itemset can be obtained to help for discovering the rare-utility itemsets in a multi-database environment.

2.**Definition**

In order to describe our problem clearly, a set of terms leading to utility mining and multiple database mining are formally defined as follows.

Definition 1. $T = \{t_1, t_2, ..., t_i, ..., t_n\}$ is a set of mutually disjoint time periods, where t_i denotes the *i*-th time period in the whole set of periods, *T*.

Note that there may be several transactions occurring in a time period.

Definition 2. $B = \{b_1, b_2, ..., b_y, ..., b_z\}$ is a set of branches (places), where b_y denotes the *y*-th branch in a multi-database environment. Each branch has its own database, which may be integrated into the centralized database.

Definition 3. $I = \{i_1, i_2, ..., i_m\}$ is a set of items, which may appear in transactions.

Definition 4. An itemset *X* is a set of items, $X \subseteq I$. If $|X| = k$, the itemset *X* is called a *k*-itemset.

Definition 5. A transaction (*Trans*) is composed of a set of occurring items *Y*, an occurring time period *t* in *T*, and an occurring branch *b* in *B*.

Note that each transaction actually occurs at a certain point of time. It is recorded in the corresponding discretized time period in this paper. Besides, the notation of *Trans.Y*, *Trans.t*, *Trans.b* is used to represent the three components in a transaction.

Definition 6. A centralized database is composed of the transactions from all the branches. That is, $D = \{Trans_1, Trans_2, \ldots, Trans_j, \ldots, Trans_n\}$, where $Trans_j$ is the *j*-th transaction in *D* and comes from a certain branch.

Definition 7. In a centralized database *D*, the set of transactions containing the itemset *X* are denoted as *W*(*X, D*). That is,
$W(X, D) = \{Trans_j | Trans_j \in D \wedge X \subseteq Trans_j.Y\}$.

Definition 8. Let PT_i be the PT table for the *i*-th item in *I*. The PT table (PT_x) for an itemset *X* is the intersection of the PT tables of the items contained in *X*.

For example, assume an itemset *X* has two items *a* and *b*. The PT table for the itemset *X* is calculated as $PT_X = PT_a \cap PT_b$.

Definition 9. The local transaction utility $q(i, Trans_j)$ of an item *i* in a transaction $Trans_j$ is the quantity of *i* in $Trans_j$ [16].

Definition 10. The external utility $s(i)$ of an item *i* is the corresponding utility value of each item in the utility table [16].

Note that the users can set the utility value of each item in the utility table to reflect the importance of the item.

Definition 11. The utility $u(i, Trans_j)$ of an item *i* in a transaction $Trans_j$ is the external utility $s(i)$ of *i* in the utility table multiplied by the local transaction utility $q(i, Trans_j)$ of *i* in the transaction $Trans_j$. That is,

$$u(i, Trans_j) = s(i) * q(i, \quad Trans_j).$$

Definition 12. The utility $u(X, Trans_j)$ of an itemset *X* in a transaction $Trans_j$ is the sum of the utility values of all the items in *X* in the transaction. That is,

$$u(X, Trans_j) = \sum_{i \in X} u(i, \quad Trans_j).$$

Definition 13. The utility $u(X, D)$ of an itemset *X* in the whole database *D* is the sum of the utility values of *X* in all transactions of *D*. That is,

$$u(X, D) = \sum_{Trans_j \in D \wedge X \in Trans_j} u(X, \quad Trans_j).$$

Definition 14. The transaction utility $tu(Trans_j)$ of a transaction $Trans_j$ is the sum of the utility values of the items contained in $Trans_j$. That is,

$$tu(Trans_j) = \sum_{i \in Trans_j} u(i, \quad Trans_j).$$

Definition 15. The PT utility $ptu(X, b_i, t_j)$ of an itemset *X* in a time period t_j at a branch b_i is the sum of the utility of the itemset *X* appearing in the transactions from the branch b_i in the time period t_j. That is,

$$ptu(X, b_i, t_j) = \sum_{Trans_j.b = b_i \wedge Trans_j.t = t_j \wedge X \in Trans_j} u(X, \quad Trans_j).$$

Definition 16. The PT transaction utility $pttu(X, b_i, t_j)$ of an itemset *X* in a time period t_j at a branch b_i is the sum of the transaction utilities of all the transactions in the branch b_i within the time period t_j. That is,

$$pttu(X, b_i, t_j) = \sum_{Trans_j.b = b_i \wedge Trans_j.t = t_j} tu(Trans_j).$$

Definition 17. The relative utility ratio *rur(X)* of an itemset *X* is the summation of all the PT utilities of *X* over the summation of all the PT transaction utilities of *X*. That is,

$$rur(X) = \sum_{b_i \in Bt_j \in T} \sum ptu(X, b_i, t_j) \Big/ \sum_{b_i \in Bt_j \in T} \sum pttu(X, b_i, t_j).$$

Three thresholds, *min_1st_util*, *min_2nd_util and min_rptuc* are also given here to define high-utility itemsets and rare-utility itemsets.

Definition 18. A high-utility itemset X in a multi-database environment is the one with its $rur(X) \geqq min\_1st\_util$.

Some other terms are given below for defining rare utility patterns.

Definition 19. The transaction-weighted utilization $twu(X)$ of an itemset X is the summation of the transaction utilities of all the transactions containing the itemset X in D [16]. That is,

$$twu(Trans_j) = \sum_{Trans_j \in D \wedge X \in Trans_j} tu(X, Trans_j).$$

Definition 20. The PT count $ptc(X, b_i, t_j)$ of an itemset X in a time period t_j at a branch b_i is the number of transactions which contain X in the branch and within the time period. That is,

$$ptc(X, b_i, t_j) = \sum_{Trans_j.b = b_i \wedge Trans_j.t = t_j \wedge X \in Trans_j} \delta(Trans_j),$$

where $\delta(Trans_j)$ is 1 if $Trans_j$ satisfies the condition, and is 0 otherwise.

Definition 21. If X includes an item i, then the PT utility confidence $ptuc(X, i)$ of a pattern $i \rightarrow X - i$ is the summation of all the PT counts of X in all branches and within all time periods over the

$$ptuc(X, i) = \sum_{b_i \in Bt} \sum_{t_j \in T} ptc(X, b_i, t_j) \Big/ \sum_{b_i \in Bt} \sum_{t_j \in T} ptc(i, b_i, t_j).$$

summation of all the PT counts of i. That is,

Definition 22. The relative PT utility confidence $rptuc(X)$ of an itemset X is the maximum PT utility

$$rptuc(X) = \max(putc(X, i_1), putc(X, i_2), ..., putc(X, i_m)).$$

confidence of X with each item i in X. That is,

Note that the $rptuc(X)$ value lies between 0 and 1 and is the largest confidence value among those for the itemset X against each item i in X. When an itemset X has a high $rptuc(X)$ value, it means some close associations exist within X. Now, rare-utility itemsets can be defined below.

Definition 23. A rare-utility itemset X is the one satisfying the following two conditions:

(a) $min\_1st\_util > rur(X) \geqq min\_2nd\_util$,

(b) $rptuc(X) \geqq min\_rptuc$.

Note that the condition "$min\_1st\_util > min\_2nd\_util$" must be satisfied. From the above definition, it can be observed that a rare-utility itemset does not satisfy the utility criterion (above the first utility threshold), but have high association of items within X. These itemsets are important and can be further promoted together because they possess high associations and can bring some acceptable profits.

For example, assume that the itemset $\{AB\}$ has satisfied the *min_2nd_util* threshold, but not get to the *min_1st_util* threshold. Also assume the PT count $ptc(\{AB\})$ of itemset $\{AB\}$ is 100 and the PT counts $ptc(\{A\})$ and $ptc(\{B\})$ of items A and B are 200 and 1,000. Thus, $rptuc(\{AB\})$ = max {100/200, 100/1,000} = 0.5. If the *rptuc* value of itemset $\{AB\}$ is equal to or larger than the *min_rptuc* threshold, then $\{AB\}$ is a rare-utility itemset with common periods and branches.

Next, since utility mining does not have the downward-closure property, the transaction-weighted-utilization *(TWU)* model [16] is commonly used to find actual high-utility itemsets in a database. The model uses the transaction utility $tu(Trans)$ of a transaction *Trans* containing X as the overestimated utility of X in *Trans* such that the downward-closure property can be achieved. The proposed TP-RUI-MD algorithm in this paper is also based on the *TWU* model to prune unnecessary itemsets. Since the proposed approach would like to find rare-utility itemsets, instead of high-utility ones, the pruning is based on the second threshold, *min_2nd_util.* Thus, rare-transaction-weighted-utilization *(RTWU)* itemset X is defined below to achieve this purpose.

Definition 24. A rare-transaction-weighted-utilization *(RTWU)* itemset X is the one with its $twu(X) \geqq min\_2nd\_util$.

4 The Proposed Mining Algorithm

In this section, we describe the proposed mining algorithm, *TP-RUI-MD* (*Two-Phase Algorithm for Mining Rare Utility Itemsets in Multiple Databases*), for discovering rare-utility itemsets in a multi-database environment in detail. In order to easily understand the whole process of the TP-RUI-MD mining algorithm, the two tables (the TIU table and the IS table) used for increasing the execution efficiency are first illustrated as follows.

1. The TIU Table

The TIU (Time Interval Utility) table is designed in the algorithm to increase the execution efficiency. An entry in the table records the total transaction utility of all the transactions occurring in a branch within a time interval. For example, assume there are four transactions which all occur at the time period t_1 in branch b_1. Also assume their transaction utilities are 16, 21, 17 and 13, respectively. The sum of the total utility for the four transactions in the time period t_1 is thus 67. It is then filled into the corresponding entry of the TIU table. An example of the TIU table is shown

in Table 2.

Table 2. An example of the TIU table.

	t_1	t_2	t_3	t_4	t_5	t_6
b_1	*67*	*78*	*99*	101	110	123
b_2	78	*89*	*88*	*107*	98	110
b_3	93	*110*	*167*	*150*	*134*	129
b_4	105	121	*118*	*105*	*122*	114
b_5	80	*90*	*103*	*97*	88	120
b_6	*145*	*130*	*138*	*110*	99	101

The proposed algorithm can easily build the TIU table by scanning the whole dataset only once.

2. The IS Table

Another table used in the proposed algorithm is the IS (Item Support) table. It is designed to efficiently catch the count values of each item. Take the item *A* as an example. Assume that the PT table and the IS table of item *A* are shown in Table 3 and Table 4, respectively. Since the on shelf bit string of item *A* at the second branch b_2 is [1,1,0,1,0] and its corresponding supports in the IS table are 1, 2 and 1. It means that the corresponding periods of all supports of item *A* at the second branch b_2 are t_1, t_2 and t_4, respectively. Such way can increase the memory usage during performing the process of mining task. The IS table can be constructed in the first scan of the database.

Table 3. The PT table of item *A*.

Branch	On Shelf Bit String
b_1	[1,1,1,1,1]
b_2	[1,1,0,1,0]
b_3	[0,1,1,0,0]

Table 4. The IS table.

Item	b_1	b_2	b_3
A	[1,1,1,2,1]	[1,2,1]	[1,2]
B	[1,2,1,1]	[1,2]	[2,3]

3. TP-RUI-MD Mining Algorithm

INPUT:
1. A centralized database *D* with multiple data sources;
2. A utility table of items;
3. PT tables of the items;
4. Three thresholds, *min_1st_util*, *min_2nd_util* and *min_rptuc*.

OUTPUT: A set of rare-utility itemsets and a set of high-utility itemsets with common on-shelf time periods and branches.

STEP 1: Load all related data and encode the *PT* tables of the items for saving memory as mentioned above.

STEP 2: Initialize the TIU table as the zero table, in which the row number is the number of branches, the column number is the number of time periods, and each entry is 0.

STEP 3: Initialize the IS table as the zero table, in which the row number is the number of branches, the column number is the number of items, and each entry is a zero vector with the dimension equal to the number of time periods.

STEP 4: Scan the database *D* to find the transaction-weighted utilization of each *1*-itemset and to build the TIU table and IS table. That is, for each transaction $Trans_j$ in the database *D*, do the following subsets.

(a) Calculate the transaction utility $tu(Trans_j)$ of $Trans_j$. That is:

$$tu(Trans_j) = \sum_{Trans_j \in D \wedge i \in Trans_j} u(i, Trans_j).$$

(b) Generate all the possible *1*-itemset in $Trans_j$.

(c) For each possible *1*-itemset *X* in $Trans_j$, set its transaction utility $tu(X) = tu(Trans_j)$.

(d) If *X* has appeared in the previous transactions which have been processed, add *tu(X)* to the transaction-weighted utilization *(twu(X))* of *X*. Otherwise, set *twu(X)* as *tu(X)*.

(e) Find in the TIU table the entry which has the same branch as $Trans_j.b$ and the same occurring time of $Trans_j.t$; Add $tu(Trans_j)$ to the value of the entry.

(f) For each item *x* in $Trans_j$, find in the IS table the entry which has the same branch as $Trans_j.b$ and the same item as *x*; Add 1 to the value of the element representing the time period of $Trans_j.t$ in the entry.

STEP 5: Check whether the transaction-weighted utility of each possible *1*-itemset is larger than or equal to *min_2nd_util*. If it is, put it in the set of rare-transaction-weighted-utilization *(RTWU) 1*-itemsets. That is,

$$RTWU_1 = \{i_k \mid twu(i_k) \geq \min\_2nd\_util, 1 \leq k \leq m\}$$

STEP 6: Set $r = 1$, where *r* is used to represent the number of items in the current candidate rare-transaction-weighted-utilization itemsets to be processed.

STEP 7: Generate the candidate set C_{r+1} from $RTWU_r$ with all the *r*-subitemsets in each candidate in C_{r+1} must be contained in $RTWU_r$.

STEP 8: Calculate the transaction-weighted-utilization *(twu(X))* of each candidate *(r+1)*-itemset as the summation of the

transaction utilities of the transactions which include *X*. That is:

STEP 9: Check whether the transaction-weighted-utilization of each possible *(r+1)*-itemset is larger than or equal to *min_2nd_util*. If it is, put it in the set of rare-transaction-weighted-utilization *(RTWU) (r+1)*-itemsets. That is:

$$RTWU_{r+1} = \{X \mid twu(X) \geq \min\_2nd\_util,\ X \in original\ C_{r+1}\}$$

STEP 10: IF C_{r+1} is null, do the next stop; otherwise, set $r = r+1$ and repeated STEPs 7 to 9.

STEP 11: Scan the database *D* to find the utility *u(X, D)* and the count of each rare-transaction-weighted-utilization itemset *X*.

STEP 12: Check whether a rare-transaction-weighted-utilization itemset *X* is a rare-utility itemset or a high-utility itemset. That is, for each rare-transaction-weighted-utilization itemset *X*, do the following substeps.

(a) Use AND operation to obtain common on-shelf branches and periods of all items in the itemset *X* by the PT table for each item.

(b) Find in the TIU table each entry which has the same branches and periods of *X;* calculate the sum $pttu(X, b_i, t_i)$ of values of these entries.

(c) Calculate the relative utility ratio *rur(X)* of *X* by its $ptu(X, b_i, t_i)$ and $pttu(X, b_i, t_i)$.

(d) Find in the IS table each entry which has the same branches and periods of each item *i* in *X*; find the *ptc(i)* of each *i*.

(e) Calculate all the confidence values *ptuc(X, i)* for the itemset *X* against each item *i*.

(f) Select the largest confidence value as the relative PT utility confidence *rptuc(X)* of *X*.

(g) If *rur(X)* of *X* satisfies the min_1st_util threshold, then the itemset *X* is identified as high utility itemset; otherwise, do the substep (h).

(h) If *rur(X)* of *X* satisfies the *min_2nd_util* threshold but not the *min_1st_util* threshold, do the next step.

(i) If *rptuc(X)* satisfies the user-specified relative PT utility confidence *min_rptuc*, then the itemset *X* is identified as rare utility itemset.

STEP 13: Output a set of rare-utility itemsets and a set of high-utility itemsets.

4. Generation of Utility Itemsets

In order to illustrate how to generate the utility itemsets, we use the process of candidate sets generated by TP-RUI-MD algorithm to illustrate the process. A rare-utility itemset must be the members of rare-transaction-weighted-utilization (*RTWU*) itemsets which satisfy *min_2nd_util*. Hence, candidate itemsets of length *k+1* are generated from rare-high-transaction-weighted-utilization itemsets of length *k*. However, if we use rare-utility itemsets of length *k* to generate candidate sets of length *k+1*, it will cause the candidate sets loss. Hence, if we use rare-transaction-weighted-utilization itemsets of length *k*, which satisfy *min_2nd_util*, to generate candidate sets of length *k+1*, then it still satisfies the downward closure property.

5 Experimental Evaluation

In this section, we conduct a series experiments to evaluate the differences between the rare-utility itemsets in a single database environment and the ones in a multiple databases environment under different user-specified parameters. Also the efficiency of the TP-RUI-MD algorithm is evaluated by varying various parameters. The simulation is implemented in J2SDK 1.5.0 and conducted in a machine with 3.0 GHz CPU and 1GB memory.

1. Description of Experimental Datasets

Since it is very difficult to obtain the real databases from the chain-store enterprise, our synthetic datasets in the experiment are generated by IBM data generator [11]. Furthermore, we develop a simulation model which is similar to the model used in [8], and then synthetic datasets generated by IBM data generator [11] are newly made up via the simulation model. In the experiments, the used factor definitions are showed as Table 5 on IBM data generator, and other parameters are still default values.

Table 5. Factor Definitions.

D	Total number of transactions
P	The number of branches (stores)
T	The number of periods
N	Total number of different items
L	Average length of items per transaction
I	Average length of maximal potentially frequent itemsets
S_u	The maximum size of branches (stores)
S_l	The minimum size of branches (stores)
min_1st_util	The first utility threshold
min_2nd_util	The second utility threshold
min_rptuc	The minimum relative PT utility confidence

However, since our main purpose is to discover rare-utility itemsets in a multi-database environment, we also develop a simulation model which is similar to the model used in [16]. Preceding synthetic datasets are then newly made up via the simulation model once again. The main reason is that the IBM data generator only generates the quantity of 0 or 1 for item in a transaction. In these datasets generated, we generate randomly the quantity of each item in each transaction, and the quantity ranges between 1 and 5. Furthermore, for each dataset generated, we also generate the corresponding utility table in which a profit value is randomly assigned to each item and the profit value ranges between 0.01 to 10.00.

2.Performance Measures

In order to evaluate the differences between the original and proposed itemsets, we define three measurements to measure the change rate. First, the type *A* change is to measure the average difference in relative utility ratio between original and proposed itemsets which must exist both them simultaneously. Second, the type *B* change is to measure the difference in number between original and proposed itemsets. And finally, the type *C* change is to measure the difference in relative PT utility confidence between original and proposed itemsets. Hence, type *A* change, type *B* change and type *C* change are defined as (1), (2) and (3), respectively.

$$Change\ Rate\ A = \sum \left(\frac{rur(MP) - rur(SP)}{rur(MP)}\right) / |SP| \quad (1)$$

$$Change\ Rate\ B = \frac{|MP| - |SP|}{|MP|} \quad (2)$$

$$Change\ Rate\ C = \sum \left(\frac{rptuc(MP) - rptuc(SP)}{rptuc(MP)}\right) / |SP| \quad (3)$$

Note that *SP* indicates the rare-utility itemsets under a single database environment, and *MP* indicates the proposed rare-utility itemsets under a multi-database environment. Besides, *rur*(*SP*) represents the relative utility ratio of rare-utility itemsets under a single database environment and *rur*(*MP*) represents the relative utility ratio of proposed itemsets in a multi-database environment. Similarly, *rptuc(SP)* represents the relative PT utility confidence of rare-utility itemsets under a single database environment and *rptuc*(*MP*) represents the relative PT utility confidence of proposed itemsets under a multi-database environment.

3.Experimental Results

(a) Impact on varying the numbers of periods and branches

In this experiment, we use the synthetic dataset L10.I4.N2K.D200K.S_u100.S_l50 to evaluate the impacts on type *A* change rate, type *B* change rate and type *C* change rate under different the numbers of periods and branches, respectively. The results of differences between original and proposed rare-utility itemsets on L10.I4.N2K.D200K.S_u100.S_l50 with *min_2nd_util* varied from 0.3% to 0.05% are shown in Figure 1, Figure 2 and Figure 3, respectively.

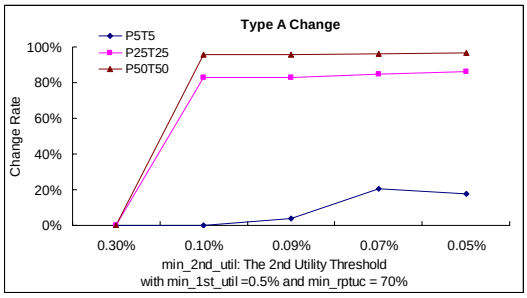


Figure 1. Impact on type *A* change rate by varying the numbers of periods and branches.

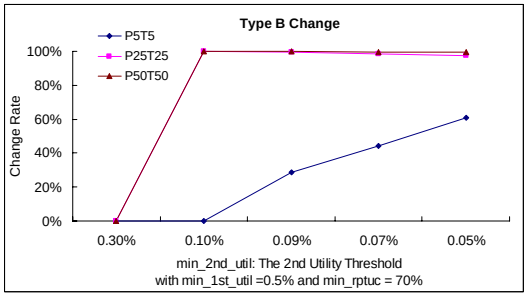


Figure 2. Impact on type *B* change rate by varying the numbers of periods and branches.

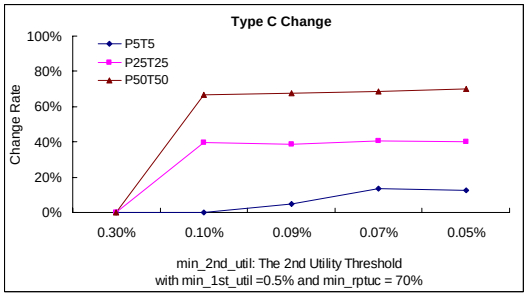


Figure 3. Impact on type *C* change rate by varying the numbers of periods and branches.

In Figure 1, Figure 2 and Figure 3, we can observe clearly that the differences in relative utility ratio, number and relative PT utility confidence between original and proposed rare-utility itemsets are obviously increasing when the numbers of both branches and periods are increasing and *min_2nd_util* is decreasing. That is,

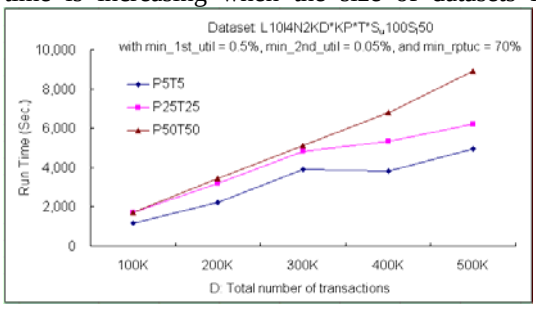
the relative utility ratio and the relative PT utility confidence of most of rare-utility itemsets are underestimated in a multi-database environment. Many of rare-utility itemsets are thus undiscovered by traditional methods in a multi-database environment. However, because our proposed algorithm TP-RUI-MD considers both branches and periods of items in a multi-database environment, we can not only obtain the correct relative utility ratio and the correct relative PT utility confidence values of rare-utility itemsets discovered by traditional methods but also can obtain these rare-utility itemsets which are undiscovered by traditional methods in a multi-database environment. Hence, the variations of three measurements are obviously increasing when the numbers of both branches and periods are increasing.

(b) **Evaluation of execution efficiency**

In this experiment, we evaluate the efficiency of our proposed algorithm TP-RUI-MD. Figure 4 shows the results of execution on dataset L10.I4.N2K.S_u100.S_l50 with different sizes of datasets varied from 100K to 500K when *min_2nd_util* is set at 0.05% simultaneously.

Figure 4. Impact on efficiency of TP-RUI-MD algorithm by varying the size of datasets.

In Figure 4, we can observe that the execution time is increasing when the size of datasets is



increasing. But the performance of the TP-RUI-MD algorithm is still very steady. This reason is that the TP-RUI-MD algorithm takes the level-wise technique to discover the rare-utility itemsets in the mining processes. Hence, the TP-RUI-MD algorithm can increase effectively the efficiency when the size of datasets is increasing.

5 Conclusion

In this paper, we have proposed a new kind of patterns, named rare-utility itemset, which considers not only individual profit and quantity of each item in a transaction but also common existing periods and branches of each one in an itemset in a multi-database environment. Besides, we have also proposed a mining approach named, *TP-RUI-MD* (*Two-Phase Algorithm for Mining Rare Utility Itemsets in Multiple Databases*), to increase the performance in terms of discovering rare-utility itemsets in a multi-database environment. The TP-RUI-MD algorithm is designed to find the rare-utility itemsets in a multi-database environment. The TP-RUI-MD algorithm uses the level-wise technique for discovering the rare-utility itemsets in a multi-database environment. The TP-RUI-MD algorithm can not discover all rare-utility itemsets in a multi-database environment, but the TP-RUI-MD algorithm can still discover these rare-utility undiscovered by existing algorithms in a single database environment.

In conclusions, there are three contributions of this paper. The first one is that we proposed a new kind of itemset named rare-utility itemset in a multi-database environment. The second one is that we have also proposed a data mining approach to discover proposed rare-utility itemsets in a multi-database environment. The last one is that detailed simulation experiments on a public dataset Foodmart were conducted to show the usefulness of proposed rare-utility itemsets and the merits of proposed mining algorithm in a multi-database environment.

As to the future work, we would apply both of the proposed rare-utility itemset and the proposed mining algorithm into other practical applications, such as the data stream, medical application, supermarket promotion application, etc, to discover the more interesting and valuable rules or patterns in a multi-database environment.

References

[1] Adda, Mehdi, Wu, Lei, and Feng, Yi. 2007. Rare Itemset Mining. *In: Proceedings of the Sixth International Conference on Machine Learning and Applications (ICMLA)*. 73-80.

[2] Adhikari, A. and Rao, P.R. 2005. Synthesizing heavy association rules from different real data sources. Pattern Recognition Letters. 29, 1 (January 2008) 59-71.

[3] Agrawal, R. and Srikant, R. 1994. Fast algorithms for mining association rules, *In: Proceedings of the 20th VLDB Conference*. Santiago, Chile, 478–499.

[4] Agrawal, R., Imielinski, T., and Swami, A. 1993. Mining association rules between sets of items in large databases. *In: Proceedings of 1993 ACM SIGMOD International Conference on Management of Data.* Washington, DC, 207-216.

[5] Ale, J.M. and Rossi, G.H. 2000. An approach to discovering temporal association rules. *In:*

Proceedings of the 2000 ACM Symposium on Applied Computing (Vol. 1). Como, Italy, 294-300.

[6] Cai, C. H., Fu, A.W. C., Cheng, C. H., and Kwong, W.W. 1998. Mining association rules with weighted items. *In: Proceedings of the International Database Engineering and Application Symposium*. Cardiff, Wales, UK, 68-77.

[7] Chan, R., Yang, Q., and Shen, Y. 2003. Mining high utility Itemsets. *In: Proceedings of the Third IEEE International Conference on Data Mining (ICDM)*. Florida, November, 19-26.

[8] Chen, Y.L., Tang, K., Shen, R.J., and Hu, Y.H. 2005. Market basket analysis in a multiple store environment. Decision Support Systems. 40, 2 (August 2005), 339-354.

[9] Chu, Chun-Jung, Tseng, Vincent S., and Liang Tyne. 2008. Mining Temporal Rare Utility Itemsets in Large Databases Using Relative Utility Thresholds. International Journal of Innovative Computing, Information and Control. 4, 8.

[10] Hu, J. and Mojsilovic, A. 2007. High-utility pattern mining: A method for discovery of high-utility item sets. Pattern Recognition. 40, 11 (November 2007), 3317-3324.

[11] IBM Quest Data Mining Project. 1996. Quest Synthetic Data Generation Code. From: http://www.almaden.ibm.com/ cs/quest/syndata.html.

[12] Koh, Y. S. and Rountree, N. 2005. Finding sporadic rules using Apriori-Inverse. In T. B. Ho, D. W.-L. Cheung & H. Liu, eds, 'PAKDD', Vol. 3518 of Lecture Notes in Computer Science (Vol. 3518), 97–106.

[13] Lan, G. C. and Tseng, Vincent S. 2008. A Novel Approach for Discovering Chain-Store High Utility Patterns in a Multi-Stores Environment. *In: Proceedings of the Second ACM KDD Workshop on Mining Multiple Information Sources (KDD/MMIS)*. Las Vegas, USA.

[14] Lee, C.H., Lin, C.R., and Chen, M.S. 2001. On mining general temporal association rules in a publication database. *In: Proceedings of the 2001 IEEE International Conference on Data Mining*. San Jose, California, 337-344.

[15] Li, Y., Ning, P., Wang, X.S., and Jajodia, S. 2003. Discovering calendar-based temporal association rules. Data & Knowledge Engineering. 44, 2 (February 2003), 193-218.

[16] Liu, Y., Liao, W., and Choudhary, A. 2005. A fast high utility itemsets mining algorithm. *In: Proceedings of the Utility-Based Data Mining Workshop*, Chicago, August, 90-99.

[17] Microsoft Corporation, Example Database FoodMart of Microsoft Analysis services.

[18] Roddick, J.F. and Spiliopoilou, M. 2002. A survey of temporal knowledge discovery paradigms and methods. IEEE Transactions on Knowledge and Data Engineering. 14, 4 (July 2002) 750-767.

[19] Srikant, R. and Agrawal R. 1996. Mining quantitative association rules in large relational tables. *In: Proceedings of the ACMSIGMOD Conference on Management of Data (SIGMOD'96)*. Montreal, Canada, 1-12.

[20] Suk, Sang-Kee, Song, Im-Young, and Kim, Kyung-Chang. 2004. MORSA: An Algorithm to Discover Association Rules in Image Data using Recurrent Items and Significant Rare Items. *In Proceedings of the 2004 IEEE International Conference on Information Reuse and Integration (IRI 2004)*. 426-432.

[21] Szathmary, Laszlo, Napoli, Amedeo, and Valtchev, Petko. 2007. Towards Rare Itemset Mining. *In: Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*. October, 305-312.

[22] Tseng, Vincent S., Chu, C. J., and Liang, T. 2006. Efficient Mining of Temporal High Utility Itemsets from Data streams. *In: Proceeding of ACM KDD Workshop on Utility-Based Data Mining*. Philadelphia, USA.

[23] Wu, X. and Zhang, S. 2003. Synthesizing high-frequency rules from different data sources. IEEE Trans. Knowledge Data Eng. 15, 2 (February 2003), 353-367.

[24] Yao, H. and Hamilton, H.J. 2006. Mining itemset utilities from transaction databases. Data & Knowledge Engineering. 59, 3 (December 2006), 603-626.

[25] Yao, H., Hamilton, H.J., and Butz, C.J. 2004. A foundational approach to mining itemset utilities from databases. *In: Proceedings of the 4th SIAM International Conference on Data Mining*, Florida, USA, 482-486.

[26] Yun, H., Ha, D., Hwang, B., and Ryu, K. H. 2003. Mining association rules on significant rare data using relative support. The Journal of Systems and Software. 67, 3, 181-191.

[27] Zhang, C., Liu, M., Nie, W., and Zhang, S. 2004. Identifying global exceptional patterns in multi-database mining. IEEE Comput. Intell. Bull. 3 (1), 19-24.

[28] Zhang, S., Zhang, C., and Yu, Jeffrey X. 2003. An efficient strategy for mining exceptions in multi-databases. Information Sciences. 165, 1-2(September 2004), 1-20.