“© 2015 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.”

Mining Actionable Combined Patterns of High Utility and Frequency

Jingyu Shao, Junfu Yin, Wei Liu and Longbing Cao
Advanced Analytics Institute
Faculty of Engineering and Information Technology, University of Technology Sydney
{Jingyu.Shao, Junfu.Yin}@student.uts.edu.au, {Wei.Liu, Longbing.Cao}@uts.edu.au

***Abstract*—In recent years, the importance of identifying actionable patterns has become increasingly recognized so that decision-support actions can be inspired by the resultant patterns. A typical shift is on identifying high utility rather than highly frequent patterns as frequent patterns are usually not actionable. Accordingly, *High Utility Itemset (HUI) Mining* methods have become quite popular as well as faster and more reliable than before. However, the current research focus has been on improving the algorithm efficiency while treating items and itemsets independently. It is important to study item and itemset coupling relationships inbuilt in the data. For example, the utility of one itemset A might be lower than the user-specified threshold; when an additional itemset B takes part in, the utility of AB may be higher than the threshold. Instead, an item's utility might be high, while the joint utility of an itemset combining another items may be low. Although some absolutely high utility itemsets can be discovered, sometimes it is often to find out that quite a lot of redundant itemsets sharing the same item are mined (e.g., if the utility of a diamond is high enough, all its supersets are proved to be *HUI*s). Such individual high utility itemsets are not actionable, and sellers cannot make higher profit if marketing strategies are created on top of such findings. To this end, here we introduce a new framework for mining actionable high utility association rules, called *Combined Utility-Association Rules (CUAR)*, which aims to find high utility and strong association of itemset combinations, which consist of multiple itemsets connected in terms of item/itemset relations. The algorithm is proved to be efficient per experimental outcomes on both real and synthetic datasets.**

Keywords - high utility itemset mining, actionable combined pattern mining, association rule, pattern relation analysis

I. Introduction

Typical data mining applications such as basket analysis rely on mining interesting patterns, in which two major objectives are addressed progressively: (1) identifying frequently associated items for commercial purpose such as cross-selling, and (2) discovering high utility itemsets [22] towards profitable selling. By using these methods, retailers can discover the most frequent or profitable products or product combinations. For example, a helmet is recommended when someone wants to buy a bicycle by mining associations between purchasing helmets and bicycles. Moreover, the outcomes delivered by high utility itemset mining [15] make it possible to discover the most profitable brand of products or product combinations, and help retailers and shopkeepers build marketing strategies to sell highly profitable goods.

TABLE I. An Example Database

TID	Transaction	TU
T_1	(A, 1) (C, 1) (D, 1)	8
T_2	(A, 1) (B, 6) (C, 2) (F, 5)	24
T_3	(A, 2) (B, 2) (C, 6) (D, 5) (E, 1)	60
T_4	(B, 4) (C, 3) (D, 2) (F, 3)	18
T_5	(B, 2) (C, 2) (F, 3)	9

TABLE II. Profit Table

Item	A	B	C	D	E	F
Profit	5	2	1	2	30	1

While the utility-based framework greatly enhances the actionability [6] of resultant patterns, compared to frequent pattern mining, it alone is still defective in some circumstances. As in utility mining, the unit profit of each item is given, while the quantity of an item depends on the transaction. Mining the utility of an itemset can be regarded as a statistic way to discover the itemsets whose utility is larger than a specific value [15]. However, the use of utility alone makes it ineffective to discover strongly associated items. For instance, selling a *pedigree cat* in a pet store happens maybe once a month or even more rarely. Although the profit is extremely high, it is probably not very wise for a manager to spend too much on designing strategies to promote such kind of pedigree pet with pet foods, because selling the pedigree pet could be just a coincidence. Furthermore, if a customer happens to purchase many other items at the same time, patterns like "pedigree cat, cat food, cattery, troughs" or "pedigree cat, necklace, cattery, cat litter" etc. could be selected as high utility itemsets. Obviously, such itemsets are neither representative nor actionable to the manager.

Such situations are described in Table I and Table II as an example. All subsets belong to T_3, containing {F} are finally proved to be HUIs, while the given threshold is no less than 50. All other itemsets are filtered as they fail to pass the threshold. Obviously, such itemsets are neither representative nor actionable to most of businessmen. However, if the threshold is set too low, new problem would appear. Searching a large number of itemsets on a large dataset may encounter a large search space.

With the examples in the above tables, Table III lists the utility, support and confidence of itemsets (here each rule is also called an itemset). The support of an itemset reduces when the itemset size (length) increases because of the *Downward*

TABLE III. A COMPARISON OF ITEMSET UTILITY, SUPPORT AND CONFIDENCE

Itemset(Pattern)	Utility	Support	Confidence
{A}	20	60%	Nil
{A → E}	40	20%	33%
{C}	14	100%	Nil
{C → B}	41	80%	80%
{BC → E}	40	20%	25%

Closure Property (DCP) [1]. However, some patterns still contain more information than others. For example, the association rule $\{C \to B\}$ is not only of a high confidence, b associated with a high utility increment from {C} (witl 14) to {BC} (41), which should be more interesting tha rules in this table.

To overcome the above issues, a new framework is r to discover the really actionable patterns: they are n succinct in terms of presentation (for a given item, o most profitable itemset instead of many should be c but also actionable (both utility-contrasted and stron sociated). Even though this seems to be very promisi interesting to users, it is critically challenging to build framework. The major challenges are below.

- The *downward closure property* [1] cannot be in the utility-based framework. The utility itemset is neither monotonic nor anti-monotoni the length of the itemset changes, which means the utility of the itemset might be either higher or lower compared with its superset or subset. In addition, for a tree-structured algorithm, it is hard to assert which branch is with the highest utility until all the branches are calculated. Thus, fast algorithms such as in [7] could not be applied to mining HUI.
- In high utility itemsets mining, a large number of candidates would be generated if a lower threshold is given. On the contrary, if the threshold is set too high, only the absolutely high utility itemsets can be discovered, and it is then hard to identify the most profitable itemsets for a given item. Subsequently, when utility is considered as the only metric to select patterns, high utility pattern mining may result in findings that are not typical and do not consider the couplings between items.
- While the combination of utility mining with frequent pattern mining is promising, the question is how to combine the utility framework with association rule mining. Association rule mining cares about the co-occurrence relationship between items based on the supports of items, while it is not clear how to measure the itemset associations for high utility items. An item's utility depends on not only the quantity of an item in a transaction, but also its item utility.

Fig. 1 illustrates the utility dynamics of utility-based itemsets when itemset grows.

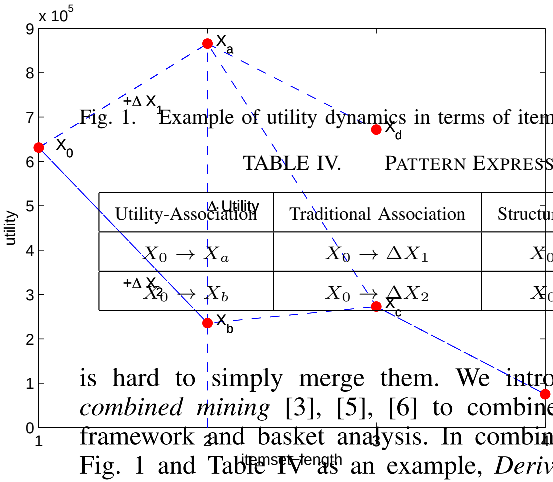


Fig. 1. Example of utility dynamics in terms of itemset growth

TABLE IV. PATTERN EXPRESSION

Utility-Association	Traditional Association	Structure of Pattern Relation
$X_0 \to X_a$	$X_0 \to \Delta X_1$	$X_0 + \Delta X_1 = X_a$
$X_0 \to X_b$	$X_0 \to \Delta X_2$	$X_0 + \Delta X_2 = X_b$

Since there are significant gaps existing in objectives and definitions of utility-based itemsets and association rules, it is hard to simply merge them. We introduce the concept *combined mining* [3], [5], [6] to combine the utility-based framework and basket analysis. In combined mining, taking Fig. 1 and Table IV as an example, *Derivative Itemset (DI) (e.g. X_a, X_b)*, also called combined itemset, is an itemset consisting of two parts. One part is called *Underlying Itemset (UI)*, which is the same part X_0 shared in both X_a and X_b in Equation 1. The other part is called *Additional Itemset (AI)*, which is different in (a) and (b), marked as ΔX_1 and ΔX_2. The combined patterns are called 'Utility-Association Combined Patterns' shown in Table IV.

$$\begin{cases} X_0 \to X_a & (a) \\ X_0 \to X_b & (b) \end{cases} \tag{1}$$

With a single underlying itemset, a cluster of *DI*s might be discovered with respect to different *AI*s. Here we define actionable high utility itemsets as a pair of *DI*s. Furthermore, we define that one of the pairs has the highest utility among all the *DI*s with the same *UI*, whereas the other *DI*s have the lowest utility. This type of combined high utility patterns is informative for decision-making. For instance, in marketing, it may suggest a manager that some products should be sold with the others for high profit, whereas the same products sold with something else may result in a loss. Obviously, such kind of combined patterns incorporate item and itemset relationship and utility, is thus more actionable for decision making.

The contributions of the work are listed below.

- A novel pattern structure, called *Actionable Combined Utility-Association Rule (CUAR)*, is proposed. It enables the generation of patterns that are of both high utility and are strongly associated, by considering the relationships between items, which provides users with actionable knowledge.

- A new interestingness for selecting patterns, called *Associated-Utility Growth (AUG)*, which integrates the relationship (association) and the utility, is proposed. To our knowledge, it is the first method for selecting patterns that have high utility without losing the representativeness (namely strong association).
- Intensive experiments on synthetic and real datasets are conducted to evaluate the proposed methods.

The rest of the paper is organized as follows. In Section 2, the background is introduced. The problem is stated in Section 3. In Section 4, we propose the *UG-Tree*, a baseline approach before introducing the *CUARM* algorithm. Section 5 presents the experimental results, and Section 6 concludes this work.

II. Related Work

A. Frequent Itemset and Association Rule Mining

Mining frequent itemsets [1] is a primary research topic and has been fully extended into diversified directions [8] for a score of years since it was first introduced in 1994 by Rakesh Agrawal et al. It is intended to identify strong rules discovered in databases using different measures of interestingness. Based on the concept of strong rules, Rakesh Agrawal et al. [1] introduced association rules for discovering regularities between products in large-scale transaction data recorded by point-of-sale (POS) systems in supermarkets. After that, a variety of algorithms were proposed to tackle the efficiency issue. Among them, *FP-Growth* [7] is one of the most efficient algorithms.

However, the frequency-based FIM framework often leads to many patterns which are not actionable at all [4]. To enhance pattern actionability, one recent effort is the high utility pattern mining which considers the utilities of unit items and itemsets in addition to their statistical significance. The existing FIM and association rule mining algorithms are incapable of capturing such high utility patterns, because they ignore the business interest [4] of each item and itemset which is essential for decision-making.

B. High Utility Itemset Mining

High Utility Itemset *(HUI) Mining* emerges as a recent solution for enhancing pattern actionability [6], since it was first introduced in 2004 [20]. Then a series of approaches have been proposed based on *Transaction-Weighted Utilization (TWU)* and *Transaction-Weighted Downward Closure (TWDC) Property*, for example *Two-Phase* [13], *IHUP* [2], *UP-Growth* [15], *CHUD* [18], *TKU* [19], d^2*HUP* [11], and *HUIM* [12]. In addition, mining high utility sequential itemsets becomes popular, such as *USpan* [22] and *UP-Span* [17].

The *UP-Growth* algorithm is one of algorithms for utility itemset mining. UP-Growth is a tree-based mining algorithm with four strategies to prune candidate itemsets. Two strategies for constructing a global UP-Tree (a structure for containing the items) called *Discarding Global Unpromising (DGU) Items* and *Decreasing Global Node (DGU) Utilities*, and two strategies for constructing a local UP-Tree called *Discarding Local Unpromising (DLU) Items* and *Decreasing Local Node (DLN) Utilities* are proposed for mining HUIs.

Even though mining algorithms such as UP-Growth extract high utility itemset efficiently, some HUIs may take place just as coincidences and are not representative and reliable for action taking. This is caused by the fact that utility is the only interestingness in the HUI mining algorithms, while utility is a subjective matter.

C. Combined Pattern Mining

The concept of "Combined Association Rules" was introduced in [25], and then extended in [23]. Combined rule mining provides a new way of merging knowledge typically for such scenarios that two features are not in the same dataset and it is not feasible to merge two datasets. For example, different features from two sets cannot be merged into one set as one is with customer IDs, age, address, gender, living region, nationality and such details, while the other contains gender, annually incomes, debt, debt repaying method, and repaying period. In this way, rather than integrating two datasets into one, a more feasible way is to build patterns consisting of constituents from respective data sets.

Further, the high impact combined pattern mining was proposed in [3]. These patterns are either frequent or infrequent, but show important business impact that is crucial for solving business problems. These patterns are exceptional because they won't be detected by traditional frequent pattern mining methods which can only find patterns with high frequency. The resultant combined patterns are represented in Equation 2. For each equation in Equation 2, *A* is a feature, an itemset, a sequence, or a sub-pattern, *B* is another, and *C* is the impact associated with the pattern consisting of *A* and *B*. *C* may refer to different risk levels, fraud or not, outlier or not. Yet, this approach is only used in the frequency-based framework.

$$\begin{cases} A_1 + B_1 \rightarrow C_1 \\ A_1 + B_2 \rightarrow C_2 \\ A_1 + B_3 \rightarrow C_3 \\ \ldots \end{cases} \tag{2}$$

More comprehensive discussions about combined mining and pattern relation analysis are in [5], [6], which address new perspectives of analyzing pattern relations and mining heterogeneous sources.

D. Frequency-Utility Mining Model

Seeking for both high profit and effective itemset combinations, Yeh et al. proposed a novel utility-frequent mining model *BU-UFM* [21]. A new definition of support called *QSupport* was proposed to measure the interestingness of the combination of both utility and frequency. Years after that, algorithm called *S-UFPM* [10] with a shared tree structure was proposed and proved to be faster. Several other algorithms such as *WUARM7* [9], *FUFM* [14] with small missing rate and less running time compared with UMining (Foundational) [20] and *MWIT* [16] were proposed to reduce the execution time compared to traditional HUI mining method.

However, none of the above methods are applicable for solving our proposed problem due to the following reasons: 1) although it is also called "utility" in [21], but the definition is different [15], and 2) the measurements and proposed algorithms perform as just filters for selecting patterns, they do not combine patterns considering their utilities and frequencies.

III. Problem Statement

Here, we define *Associated-Utility Growth (AUG) Pattern*, which introduces both association and utility growth into combined mining and mining actionable combined pattern pairs.

A. Definitions

Taking Table I and II as an example, let $D = \{T_1, T_2, ..., T_m\}$ be a transaction database, and $\mathcal{I} = \{i_1, i_2, ..., i_n\}$ be a set of finite and discriminative items. Each transaction $T_c \in D$ $(1 < c < m)$ is a subset of $\mathcal{I}$ with a distinct identifier called *TID*. In a given transaction T_c, each item i_k appears with a positive integer, $q(i_k, T_c)$ is called i_k's *quantity utility* in T_c. Also, each item in $\mathcal{I}$ is associated with a positive number $p(i_k, \mathcal{I})$, which is called i_k's *profit utility* in $\mathcal{I}$.

Definition 1: The *frequency* of an itemset X counts the times it appears in all transactions and is denoted as $SC(X)$. The *support* of X is $SC(X)$, divided by the number of transactions in D, and is denoted as $supp(X)$.

Based on Table I, $SC(A)$ is 3, $SC(AB)$ is 2, and *Supp(AB)* = 40%.

Definition 2: The *utility of item* i_k in a transaction T_c is the profit utility of the item times its quantity utility in a transaction, defined as

$$u(i_k, T_c) = p(i_k, \mathcal{I}) * q(i_k, T_c) \tag{3}$$

An item i_k with its utility in a transaction T_c is denoted as $u(i_k, T_c)$ $(i_k \in T_c)$.

Definition 3: The *utility of an itemset* X in a transaction T_c is the utility sum of all items belonging to the itemset, defined as

$$u(X, T_c) = \sum_{i_k \in X} u(i_k, T_c) \tag{4}$$

An itemset contains l discriminative items is called an *l-length* itemset, where $X \subseteq \mathcal{I}$. The utility of the same item in different transactions might be different considering the quantity of each item purchased.

Definition 4: The *utility of an itemset* X in the whole database D is the sum of the utility of this itemset in all transactions. It is denoted as $U(X)$, and defined as

$$U(X) = \sum_{X \subseteq T_c \wedge T_c \in D} u(X, T_c) \tag{5}$$

In addition, one item can be regarded as an 1-length itemset.

Definition 5: The minimum utility threshold is denoted as *min_util*, and a set of all itemsets whose utilities are higher than *min_util* is denoted as $f_H(D, min\_util)$. The goal of HUI mining is to find such itemset, $f_H(D, min\_util)$.

Definition 6: The *transaction utility* of the transaction T_c is denoted as $TU(T_c)$ and defined as

$$TU(T_c) = \sum_{i_k \in T_c} u(i_k, T_c) \tag{6}$$

Definition 7: The *transaction-weighted utility* of the itemset X is the sum of the transaction utilities of all the transactions that X belongs to. It is denoted as $TWU(X)$ and defined as

$$TWU(X) = \sum_{X \subseteq T_c \wedge T_c \in D} TU(T_c) \tag{7}$$

Definition 8: The *high transaction-weighted utility itemset (HTWUI)* consists of those itemsets whose TWU is no less than *min_util*.

Property 1: The *transaction-weighted downward closure property* holds for HUI, says that if an itemset X is not an HTWUI, all its supersets are not HUIs because $U(X) \leq TWU(X)$.

B. Mining Combined Utility-Association Rules

HUI Mining is to discover itemsets with high utility whose utilities are higher than the minimum threshold, in which their frequencies are not concerned. HUI could be regarded as an extension of FIM towards addressing business interest [4] represented by utility. Large number of itemsets would come out if the threshold is not proper. As shown in Fig. 1 of the utilities in a cluster of incremental itemsets, we can see the utility is changing dynamically and irregularly. As the basic rule in social marketing is to gain profit, businessmen might only care about products that can make profit for them, and are also interested in converting those less popular goods to be more preferred. It is helpful for business purposes to figure out those itemsets which 1) are low utility itemsets, but become high utility after one additional item (or itemset) is added; 2) are high utility, but become low utility after one item (or itemset) added. An itemset whose length increases with adding new items is called *Incremental Itemset*, which means the number of items it contains would grow but never reduce.

Even though the utility metric provides reasonable evidence for selecting patterns of business interest, it does not provide sound insurance about how sound a high utility pattern could be, which can be complemented by frequency-based filters. Therefore, we propose a new strategy which combines both utility and frequency, named as *Associated-Utility Growth (AUG) Pattern Mining*, to identify *Association-Maximum Incremental Itemsets (AMII)* and *Utility-Increasing Incremental Itemsets (UIII)*. The utility of an incremental itemset is dynamic, meaning that the utility evolves when additional items are added to the underlying items, which forms a utility curve. A UIII structure is necessary to discover those utility increment-oriented itemsets. Also, it is no doubt that the frequency of an incremental itemset monotonically decreases. Here maximum association does not refer to those itemsets with the highest frequency, but those itemsets where items share a reasonable relationship with each other. In some way, it also means that the frequency would not change too much after adding one or several items. Subsequently, the measurement of *AUG* is considered for candidate pruning to

TABLE V. Reorganized Transactions with Their Reorganized-TUs

TID	Transaction	RTU
T_1'	(C, 1) (A, 1) (D, 1)	8
T_2'	(C, 2) (B, 6) (A, 1) (F, 5)	24
T_3'	(C, 6) (B, 2) (A, 2) (D, 5) (E, 1)	60
T_4'	(C, 3) (B, 4) (D, 2) (F, 3)	18
T_5'	(C, 2) (B, 2) (F, 3)	9

TABLE VI. Items with Their TWUs

Item	A	B	C	D	E	F
Profit	92	111	119	77	60	51

find out those having both high utility growth and highly associated items. In this way, only one significant combined pattern is selected for each UI.

C. An Abstract Model: 2-length Combined Utility-Association Pattern Pair

Here we illustrate the application of *actionable Combined Utility-Association Rules* through identifying *2-length Combined Utility-Association Pattern Pair*. Take Fig. 1 as an example (here we suppose they hold the minimum confidence threshold, or X_a and X_b share the same relation with X_0), the 1-length itemset X_0 is firstly treated as a *UI*, then two items added separately form two 2-length itemsets: ΔX_1 is added and forms one new itemset X_a with higher utility, ΔX_2 is added and forms the other new itemset X_b with lower utility. X_a and X_b are two supersets of X_0. The pattern pair is shown in Equation 8.

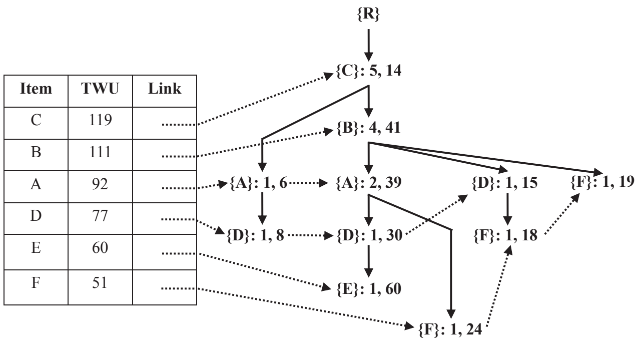
$$\begin{cases} X_0 \to X_a \Rightarrow U - Increase & (a) \\ X_0 \to X_b \Rightarrow U - Decrease & (b) \end{cases} \tag{8}$$

Definition 9: Positive/Negative Impact Rule (PIR/NIR): referring to rules structured as Equation 8(a)/(b) which is called positive/negative impact rules, whose right-hand side is associated with utility higher/lower than X_0 on the left-hand side.

If such rules are used for marketing purposes, a retailer should know what promotion mixtures make more profit and what would lead to low profit if they are put together, based on the positive and negative rules.

IV. The CUARM Approach

We aim to provide patterns to retailers for promotion strategies include increasing high utility product combinations which are highly associated with each other. We name them as *Utility-Association Rules*, which cannot be discovered by traditional association rule methods or utility mining algorithms alone. The algorithm for identifying interesting utility-association rules is called *Combined Utility-Association Rules Mining (CUARM)*. To this end, two factors, *Contribution* and *Weight*, are proposed to select combined patterns of both high utility growth and strong association.



Item	TWU	Link
C	119	
B	111	
A	92	
D	77	
E	60	
F	51	

Fig. 2. Header Table and A UP-Tree When *min_util = 0*

TABLE VII. Composition in Each Node

Formation	Explanation
N.na	the item name of this node
N.sc	the support count of this node
N.u	the node utility of this node
N.p	the parent node of this node
N.nl	the node link which points to a node whose name is same as N.na

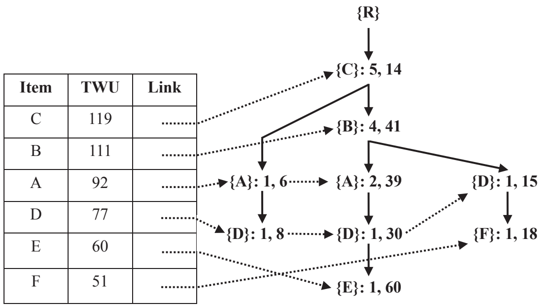
A. The Baseline Approach

As stated above, the purpose of HUI mining is to find the set of all high utility itemsets $f_H(D, min\_util)$ efficiently. One way to obtain the target patterns is to obtain $f_H(D, 0)$ first, which can be achieved by using UP-Growth with *min_util* = 0, and then extract the UARs from it. In essence, the baseline approach is a strategy to maintain all itemsets with their utilities. Readers can refer to [15] for the detailed structure and examples about UP-Growth and UP-Tree. In addition, the UG-Tree we proposed is built with *min_util* = 0 based on Table V and Table VI.

B. The Proposed Approach

1) UG-Tree: Our algorithm is based on utility growth and association rule mining, thus all branches whose utilities decreasing should be pruned from the UP-Tree. Since the utility growth of each node in a given branch could be either positive or negative, we prune branches from its external nodes until the first node whose utility growth becomes positive. Such re-organized tree structure is called *UG-Tree*, as shown in Fig. 3.

The composition of each node N is listed in Table VII. The *Rebuilt Table* is displayed to demonstrate the traversal of a UG-Tree. In the downward table, each row is composed of an *item name*, a *transaction-weighted utilization value* and a *link*. Each link points to the node having the same item name as shown in the UG-Tree. The nodes with the same item names can be traversed efficiently by following the links between downward table and the nodes in UG-Tree.



Item	TWU	Link
C	119	
B	111	
A	92	
D	77	
E	60	
F	51	

Fig. 3. Downward Table and A UG-Tree When *min_util = 0*

2) UG-Tree Construction: A UG-Tree can be constructed by only scanning the original database twice. During the first database scan, items and their TWUs are captured via the calculation of the TU of each transaction and TWU of each item. The downward table is then formed by items inserted in the TWU-decreasing order. In the second scan, the reorganized transactions as well as their utilities are inserted into the UG-Tree, as shown in Table V. In addition, all utility-decrement branches are pruned and shown in Fig. 3. Finally, a UG-Tree is created with root R. The composition of each node is listed in Table VII.

Definition 10: A transaction after the process of the above reorganization is called *Rebuilt Transaction* and its TU is called *rebuilt transaction utility (RTU)*, denoted as $RTU(T_c)$.

The construction of UG-Tree will be completed after all RTs are inserted with their RTUs. Fig. 3 presents a UG-Tree when the minimum utility threshold is set to zero.

By using the UG-Tree, which contains enough information, we can generate the utility-association rules. While the UG-Tree is retrieved, the utility of each itemset can be discovered and prepared for the calculation of *Factor C*. At the same time, the support count of each item can also be found through *item.nl* and used for *Factor W*. These two factors will be discussed in the next subsections.

C. Impact Factor of Utility Growth across Combined Itemsets

Both PIRs and NIRs are not difficult to be acquired, since just scanning the database one more time with a comparison added will help. However, this belongs to post-processing approach which is less efficient, and ignores item relation analysis during the mining process. For these, a measurement called *contribution* is proposed below to discuss the relationship among these three itemsets (UI, AI and DI), on top of PIRs and NIRs.

Definition 11: The *contribution* of Additional Itemset (ΔX) to make utility change (increase or decrease) from the Underlying Itemset (X_0) to Derivative Itemset (X), denoted as $C(\Delta X|X_0)$, is defined as:

$$C(\Delta X|X_0) = \begin{cases} \frac{2}{1+e^{-\mathcal{R}}} - 1 & , \quad U(X) > U(X_0) \\ \mathcal{R} & , \quad U(X) \leq U(X_0) \end{cases} \tag{9}$$

In Equation (9),

$$\mathcal{R} = \frac{U(X)}{U(X_0)} \tag{10}$$

This equation is proposed to measure whether *itemset* ΔX, as an additional part of X_0, plays an important role in transforming itemset X_0 to X in the utility perspective. Here, $U(X)$ is the utility of X in the whole database. The first equation is associated with utility increase, corresponding to PIR, while the second corresponds to NIR. In the first function, a logistic function is used to converge the contribution to the range of [0,1], which will be discussed later in the next section.

As demonstrated in Fig. 1, the function of utility and item-length is neither monotonic nor anti-monotonic, meaning the utility of an itemset is dynamic with its length increasing. The utility of each item is taken into consideration to analyze the contribution. Furthermore, the contribution of additional itemset provides influence within the itemset sharing the same support counts, which is not suitable for the operation of contribution. However, even though the contribution is presented to measure whether ΔX plays a significant role to promote the utility from X_0 to X, it is still measured by the rate $\mathcal{R}$ because ΔX might appear in other itemsets, and the utility should be calculated in another way.

Two conditions might appear: 1) The contribution of this ΔX is high enough to make $X(DI)$ a significant PIR, which also means itemset X_0 has a strong utility-association with itemset X; 2) The contribution of this ΔX is so low that this $\{X_0 \rightarrow X\}$ is proved to be a significant NIR thus itemset X_0 is rarely utility-associated with itemset X.

D. Co-occurrence Association between Underlying and Additional Itemsets

Definition 12: The weight of additional itemset to measure the co-occurrence frequency of underlying itemset and additional itemset, denoted as $W(\Delta X|X_0)$, is defined as:

$$W(\Delta X|X_0) = \frac{Supp(X)}{Supp(X_0 \cup \Delta X)} \tag{11}$$

It is a reduction of the Jaccard similarity coefficient [1]:

$$J(X_0, \Delta X) = \frac{|X_0 \cap \Delta X|}{|X_0 \cup \Delta X|} \tag{12}$$

This equation is to examine whether itemset ΔX has a high or low association by measuring their co-occurring frequency with itemset X_0.

In Equation (11), *Supp(X)* is the support of X_0 and ΔX appearing together, and *Supp(*$X_0 \cup \Delta X$*)* is the support of either X_0 or ΔX appearing:

[1] http://en.wikipedia.org/wiki/Jaccard_index

$$Supp(X_0 \cup \Delta X) = Supp(X_0) + Supp(\Delta X) - Supp(X) \quad (13)$$

E. Impacted Coefficient of the Additional Itemset

Definition 13: The impacted coefficient of an additional itemset is to describe how effective this itemset is to manufacture the derivative itemset from underlying itemset, denoted as $AUG(\Delta X|X_0)$, defined as:

$$AUG(\Delta X|X_0) = \sqrt{\frac{C^2(\Delta X|X_0) + W^2(\Delta X|X_0)}{2}} \quad (14)$$

This equation averages the value of $C(\Delta X)$ in Equation 9 and $W(\Delta X)$ in Equation 11. Here we use the *Quadratic Mean (QM) (also known as Root-Mean Square)* to measure the significance of the itemset ΔX in terms of both utility and relationship perspectives because it represents the sample standard deviation of the difference between *W* and *C*, thus the result cannot be affected heavily by the smaller value. It is easy to prove:

$$QM^2(X) = (\overline{X})^2 + \sigma^2(X) \quad (15)$$

Here, $\overline{X}$ and $\sigma(X)$ stand for the arithmetic mean and the standard deviation of *W* and *C*. We also tried another measurement by *Harmonic Mean (HM)* as a baseline, which is proven to be less effective in our experiments.

For a specific X_0, for each itemset ΔX to be considered, the higher AUG means this itemset is likely to impel the underlying itemset into higher utility itemset. On the contrary, the lower the AUG is, the lower utility that derivative itemset might be. As all the AUG would be calculated, only the largest AUG value itemset will be chosen.

F. The CUARM Algorithm

In this section, an algorithm named *Combined Utility-Association Rule Mining (CUARM)* is proposed to discover all the actionable combined utility-association rules. At the beginning of the algorithm, it picks all UIs as candidates. For each UI, all the combined patterns are discovered with their AUGs which form a combined pattern cluster as in Equation 2, and only the most effective pattern would be selected. In addition, if two patterns are coupled with utility increment and decrement, a combined pattern pair forms.

The input is the transaction database, including all transactions with the utility of each item, and the output is the combined pattern pairs, their underlying itemset and the corresponding utilities. In line 1, we prepare all the itemsets with their utilities in the alphabetical order and the length of longest itemset. In lines 2-5, we start with each of the UIs named $itemset_0$ with its utility U_0. In lines 6-11, the DIs are ready and we calculate their AUGs. In line 12-13, we select the pattern with max AUG values as CUAR.

Algorithm 1: CUARM

```
   Input: Transaction database $D$, including the utility
          $U(X)$ of each item in $D$
   Output: All actionable combined utility-association
           rules
1  Get all itemsets' utilities via UG-Tree ;
2  Get the length of longest itemset: lmax ;
3  for len = 1, len < lmax, len++ do
4      for Itemset whose length is equal to len do
5          Get $itemset_0$ with $U_0$(itemset-utility);
6          for itemset.length > len do
7              Check inclusive and utility changes;
8              Get $itemset_1$ with $U_1$;
9              Calculate C;
10             Scan the database, get W;
11             Calculate AUG;
12         Selected max one;
13         Present this utility-association rule;
```

TABLE VIII. CHARACTERISTICS OF DATASETS

Dataset	Number of Transactions	Number of Items	Average Length
Retail[2]	88162	16470	10.3
Chainstore[3]	1112949	46086	7.3
t20i6d100k	100000	658	13.7
c20d10k	10000	187	13

V. Experiments

In this section, we conduct intensive experiments to evaluate the proposed methods. Our experiments were run on a PC with a 2.30 GHz Intel Core, 16 gigabyte memory. CUARM is implemented in Java. Two real datasets and two synthetic datasets are used for the experiments. The real datasets are *Retail*[2] and *Chainstore*[3], and the synthetic datasets are *t20i6d100k* and *c20d10k*. The parameters of the datasets are listed in Table VIII.

A. Comparison of Two Functions for Calculating Impacted Coefficient

Here we propose two functions for calculating the impacted coefficient. One is the quadratic mean (QM), which is adopted in this paper, the other function is the harmonic mean (HM), which is proved to be less accurate in the experiments. Those itemsets with a good coefficient measurement should be associated with both high frequency and high utility growth, we thus can separate the database randomly. If the output itemsets discovered in each sub-database are stable, we can assume that this measurement is suitable.

The experiments are conducted on the Retail dataset for the sake of simply examining the QM function. The top 100 experimental results are selected and shown in Fig. 4. The figure on the left shows the comparison between UP-Growth and QM, while the figure on the right shows the result of QM

[2] http://www.philippe-fournier-viger.com/spmf/index.php?link=datasets.php

[3] http://cucis.ece.northwestern.edu/projects/DMS/MineBench.html

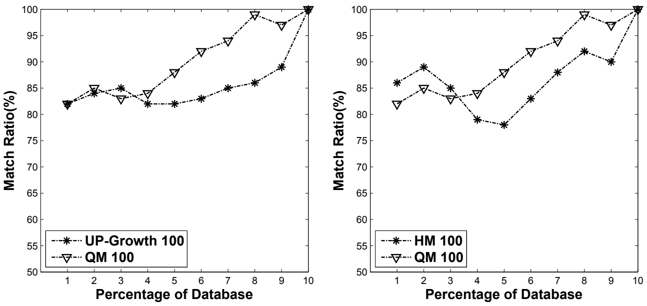


Fig. 4. The Comparison of HM, QM and UP-Growth

and HM on $C(\Delta X)$ and $W(\Delta X)$. The database is split into 10 parts randomly. The first part contains 10% transactions in the database and each later part contains 10% more transactions than the former part (such that the second part contains 20% and the last part is 100%). The X axis is the k_{th} $(1 \leq k \leq 10)$ part of the database, and the Y axis is the match ratio, which means the ratio of the exact patterns found in the k_{th} part matching with the $(k+1)_{th}$ part. As seen from the figures, the QM method outperforms both HM and UP-Growth.

B. Experimental Evaluation of CUARM

Next, we present the experimental results of comparing derivative itemsets with the traditional HUIs, FIs and UIs (Underlying Itemsets) respectively. The statistic values of each dataset are shown in Table VIII. The experiment is conducted as follows. Firstly, we collect all the utility itemsets with their utilities and frequencies in each dataset respectively. Secondly, we also collect all the frequent itemsets with their frequency and utility. Then we calculate the utilities of the frequent itemsets, frequencies of the HUIs and both utilities and frequencies of the derivative itemsets. At last, we plot the frequency of itemsets discovered via UP-Growth and CUARM, the utility of itemsets discovered by FP-Growth and CUARM and the utility changes from each underlying itemset to derivative itemset as shown in Fig. 5. Such exhibition is made for the comparison of our algorithm with FIM and HUI to demonstrate the Utility-Association Rules we discovered have both high utility and high frequency.

Here, all the frequent and utility itemsets we compare with contain at least two items because the derivative itemsets our algorithm discover contain no less than two items.

Experiments on Real Datasets

We first present the outputs of dataset *Retail* in Fig. 5(a) and Fig. 5(b). Top 50 patterns of each algorithm are selected for experiments. By analyzing the frequencies and utilities of patterns, many of them are without much difference in both two experiments, which means the association rules we found via traditional AR algorithms are also high utility-association rules via our method. In addition, such rules are also with high utilities. This explains why some parts of the curves overlap. In addition, customers prefer to buy a few products at one time, that is, most of FIs and HUIs contain only one or two items, which also explains the observations.

In datasets *Chainstore*, the differences are much clearer, because customers usually prefer a variety of products in each of transactions, and high utility itemsets are always low in frequency, while highly frequent itemsets are with low utility. For example, in Fig. 5(d), the CUARM performs much better than that of FP-Growth, while in Fig. 5(e), even at some points, the performance is not so good, the global performance is much better. To sum up, we can assert that the performance of our algorithm CUARM is much better than the others.

Experiments on Synthetic Datasets

Experimental results on synthetic datasets *t20i6d100k* and *c20d10k* are shown in Fig. 5(g), Fig. 5(h), Fig. 5(j) and Fig. 5(k). The results are much clearer than those from real datasets because the items included are much neat and with orderliness. For most of the patterns discovered via CUARM, the frequencies are much higher than those traditional high utility itemsets. At the same time, most Utility-associated rules are also with much higher, i.e., twice the utility, than traditional association rules, especially in Fig. 5(h) from dataset *t20i6d100k*.

C. Evaluation of the Utility Increment

We demonstrate the utility increment in a graphic way to show how the utility increases from underlying itemset to derivative itemset. The utility increment is valued based on the same datasets as above. points are ordered by the utility of derivative itemsets. The performance of our algorithm varies from one dataset to another. The performance in chainstore is much better than that in retail because the transaction time in chainstore is 12 times more than that in retail while the item types are only twice more. However, in the synthetic datasets, the performance is better. In conclusion, for each dataset, the performance is different but the utility actually increases.

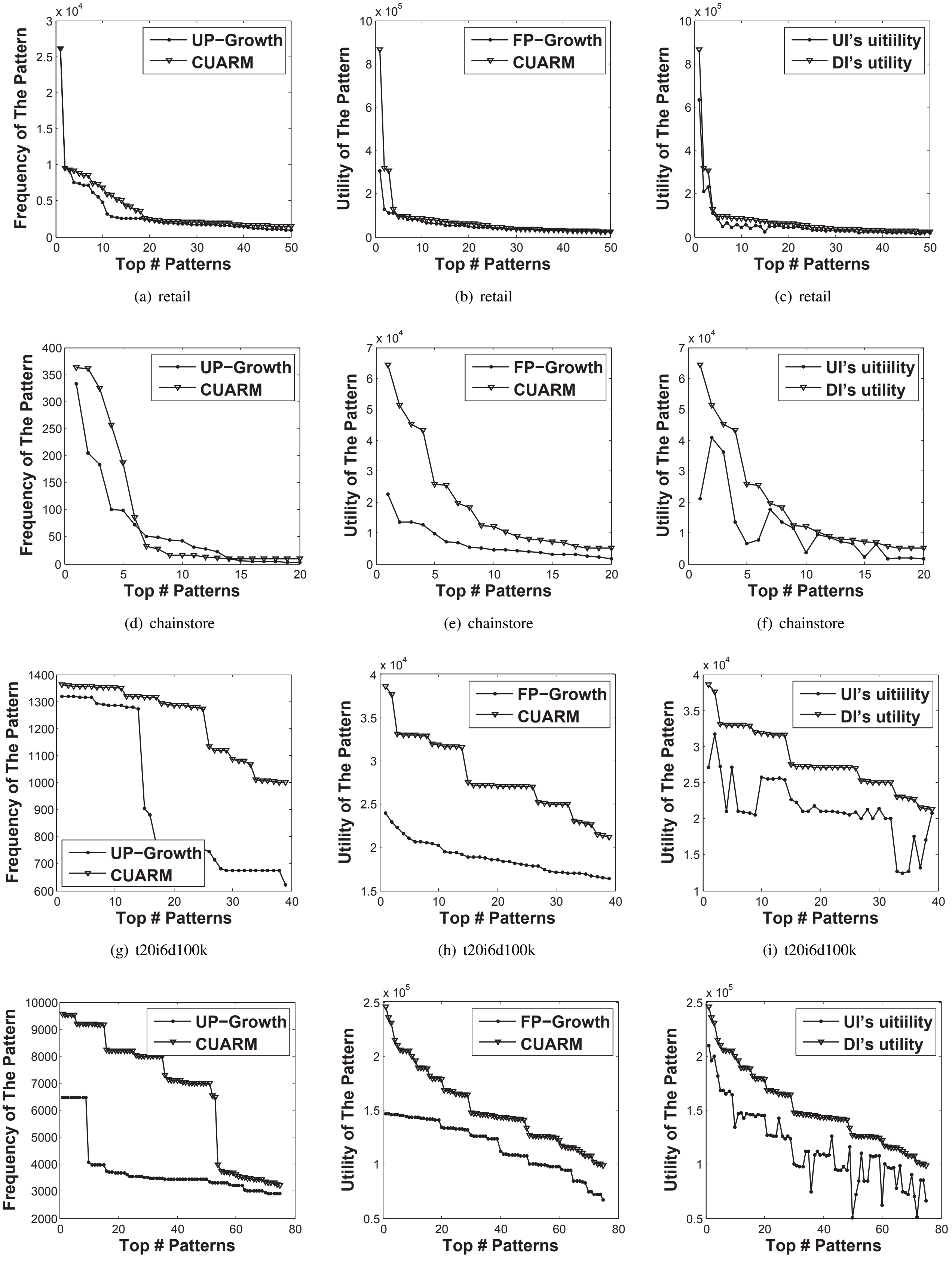
D. Summary of Findings

Based on the above datasets and experimental results, a table is used to demonstrate the conclusion that comes from our experiments and shown as Table IX. This table describes the number of itemsets whose utilities increase or decrease with a given threshold. Also, two kinds of utility incremental forms are listed. One is the utility of derivative itemset is higher than both the utilities of underlying itemset and additional itemset, which is denoted as FA; the other is that the utility of derivative itemset is only higher than the utility of underlying itemset, which is denoted as FB. As for each underlying itemset, only one derivative itemset would be discovered, some FA and FB might be ignored.

For the utility decrement itemsets whose utilities are only lower than the underlying itemsets would not be considered in this table because these itemsets can also be regarded as FBs when the underlying itemsets and additional itemsets exchange.

VI. Conclusions and Future Work

Traditional high utility itemset mining methods face problems if the minimum utility threshold is set too high, the itemsets discovered may contain unrepresentative items; while if the threshold is set too low, too many redundant itemsets will be found. On the other hand, traditional association rule mining ignores the utility hidden among the items. This work



(a) retail

(b) retail

(c) retail

(d) chainstore

(e) chainstore

(f) chainstore

(g) t20i6d100k

(h) t20i6d100k

(i) t20i6d100k

(j) c20d10k

(k) c20d10k

(l) c20d10k

Fig. 5. Experiments for FP, UP and CUARM

TABLE IX. UTILITY VARIATION CONCLUSION

Dataset	Minimum Support	Number of Itemsets	Utility Incremental Rate	Number of FA	Number of FB	Number of Decremental Itemsets
Retail	0.01	89	20.3% - 50.4%	28	22	39
	0.008	135	18.6% - 50.4%	37	46	52
	0.002	1667	8.4% - 50.4%	473	769	425
Chainstore	0.002	79	4.6% - 207.2%	7	19	53
t20i6d100k	0.017	33	25.7% - 78.5%	8	11	14
	0.015	79	22.8% - 78.5%	24	19	36
	0.012	383	1.8% - 78.5%	112	137	134
c20d10k	0.05	120	19.7% - 150.9%	28	48	44

proposes a novel pattern selection method from two aspects. One is the co-occurrence of two (underlying and additional) itemsets; another is the utility increment from underlying itemset to derivative itemset. It is an effective approach for identifying actionable combined utility itemsets, in which, for different items, only one itemset will be selected with the highest association-utility growth, which caters for both high association and high utility. Thus, only the most effectively impacted itemsets will be presented. The results demonstrate that our method can discover patterns that are composed of different item combinations of both utility increment and high representativeness.

For the future work, we may find some more interesting pattern selection methods. For example, there exists a dependent relationship between two itemsets A and B, which means A might appear frequently alone or with other items, but for most time B appears together with A.

VII. ACKNOWLEDGMENTS

This work is sponsored in part by Australian Research Council Discovery Grant (P130102691).

REFERENCES

[1] R. Agrawal, R. Srikant, 1994, ‘Fast Algorithms for Mining Association Rules’, in Proc. of the 20th Int’l Conf. on Very Large Data Bases, pp.487-499, Santiago, Chile.

[2] C.F. Ahmed, S.K. Tanbeer, B.-S. Jeong, and Y.-K. Lee, 2009, ‘Efficient Tree Structures for High utility Pattern Mining in Incremental Databases’, in Proc. of IEEE Transactions on Knowledge and Data Engineering, Vol. 21, Issue 12, pp. 1708-1721.

[3] L. Cao, Y. Zhao, C. Zhang, 2008, ‘Mining Impact-Targeted Activity Patterns in Imbalanced Data’, IEEE Trans. on Knowledge and Data Engineering, 20(8): 1053-1066.

[4] L. Cao, P. S. Yu, C. Zhang and Y. Zhao, 2010, ‘Domain Driven Data Mining’, Springer.

[5] L. Cao, et al, 2011. Combined Mining: Discovering Informative Knowledge in Complex Data, IEEE Trans. SMC Part B, 41(3): 699-712.

[6] L. Cao, 2013, ‘Combined mining: Analyzing object and pattern relations for discovering and constructing complex yet actionable patterns’, Wiley Interdisc. Rew.: Data Mining and Knowledge Discovery 3(2): 140-155.

[7] J. Han, J. Pei and Y. Yin, 2000, ‘Mining Frequent Patterns without Candidate Generation’, in Proc. of the ACM-SIGMOD Int’l Conf. on Management of Data, pp. 1-12, Dallas, TX, USA.

[8] J. Han, H. Cheng, D. Xin and X. Yan, 2007, ‘Frequent Pattern Mining: Current Status and Future Directions’, DMKD, 15: 55-86.

[9] M. S. Khan, M. Muyeba, and F. Coenen, 2008, ‘A Weighted Utility Framework for Mining Association Rules’, in Proc of the Second UKSIM European Symposium on Computer Modeling and Simulation, pp. 87-92.

[10] X. Lin, Q. Zhu, F. Li, Z. Geng, and S. Shi, 2010, ‘S Share Strategy for Utility Frequent Patterns Mining’, in Proc. of the Seventh International Conference on Fuzzy Systems and Knowledge Discovery’, pp. 1428-1432, Yantai, China.

[11] J. Liu, K. Wang, and B. C. M. Fung, 2012, ‘Direct Discovery of High Utility Itemsets without Candidate Generation’, in Proc. of the IEEE Int’l Conf. on Data Mining (ICDM).

[12] M. Liu and J. Qu, 2012, ‘Mining High Utility Itemsets without Candidate Generation’, in Proc. Of the ACM Int’l Conf. on Information and Knowledge Management (CIKM), pp. 55-64.

[13] Y. Liu, W. Liao, and A. Choudhary, 2005, ‘A Two-Phase Algorithm for Fast Discovery of High Utility Itemsets.’ in Proc. of PAKDD, pp. 689-695.

[14] S. Shankar, T. Purusothaman, S. Kannimuthu, and P. K. Vishnu, 2010, ‘A Novel Utility and Frequency Based Itemset Mining Approach for Improving CRM in Retain Business’, International Journal of Computer Applications, Volume 1, No. 18, pp. 87-94.

[15] V. S. Tseng, C.-W. Wu, B.-E. Shie, and P. S. Yu, 2010, ‘UP-Growth: An Efficient Algorithm for High Utility Itemset Mining’, in Proc. of Int’l Conf. on ACM-SIGMOD, pp.253-262.

[16] B. Vo, B. Le, and J. Jung, 2012, ‘A Tree-Based Approach for Mining Frequent Weighted Utility Itemsets’, in Proc. of ICCCI 2012, Part I, LNAI 7653, pp. 114-123.

[17] C. Wu, Y. Lin, P. S. Yu, and V. S. Tseng, 2013, ‘Mining High Utility Episodes in Complex Event Sequences’, in Proc. of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), pp. 536-544.

[18] C. Wu, P. Philippe, P. S. Yu and V. S. Tseng, 2011, ‘Efficient Mining of a Concise and Lossless Representation of High Utility Itemsets’, in Proc. of IEEE Int’l Conf. on Data Mining (ICDM), pp.824-833.

[19] C. Wu, B. Shie, V. S. Tseng, and P. S. Yu, 2012, ‘Mining top-K high utility itemsets’, in Proc. of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), pp. 78-86.

[20] H. Yao, H. J. Hamilton, and C. J. Butz, 2004, ‘A foundational approach to mining itemset utilities from databases’, in Proc. of the 4th SIAM Int’l Conf. on Data Mining, Florida, USA.

[21] J. -S. Yeh, Y. Li, and C. Cheng, 2007, ’Two-Phase Algorithms for a Novel Utility-Frequent Mining Model’, in Proc. of PAKDD Workshop, LNAI 4819, pp. 433-444.

[22] J. Yin, Z. Zheng and L. Cao, 2012, ‘USpan: An Efficient Algorithm for Mining High Utility Sequential Patterns’, in Proc. of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), pp. 660-668.

[23] H. Zhang, et al., 2008, ‘Combined Association Rules Mining’, in Proc. of PAKDD08, pp.1069-1074.

[24] Q. Zhao and S. Bhowmick, 2003, ‘Association Rules Mining: a Survey’, Journal of Nanyang Technological University, 2003116.

[25] Y. Zhao, et al., 2007, ‘Mining for Combined Association Rules on

Multiple Datasets', in Proc. of the KDD 2007 Workshop on Domain Driven Data Mining, San Jose, CA, USA, pp. 18-23.