Mining High Utility Itemsets from Large Dynamic Dataset by Eliminating Unusual Items

Switi C. Chaudhari
M. Tech. IV Sem. (CSE)
Lord Krishna College of Technology
Dept of CSE, Indore

Vijay Kumar Verma
Associate Professor
Lord Krishna College of Technology
Dept of CSE, Indore

ABSTRACT

Utility-based data mining is a new research area interested in all types of utility factors in data mining processes [1]. The basic meaning of utility is the quantity sold, interest, importance & profitability of items to the users. Utility of items in a transaction database consists of two aspects:

1. The importance of distinct or unique items, which is called external utility.
2. The importance of the items in the transaction, w is called as internal utility.

Mining high utility itemsets from the databases is not an easy task. Pruning search space for high utility itemset mining is difficult because a superset of a low utility itemset may be a high utility itemset. Existing studies [2,4,9] applied overestimated methods to facilitate the mining performance of utility mining. In these methods, first we will get potential high utility itemsets, and then an additional database scan is performed for identifying their utilities. However, the existing methods often generate a huge candidate itemsets and the mining performance is degraded consequently. In this paper we proposed Eliminating Unusual Itemset by Eliminating item set which is low utility item set to reduce search space. Proposed methods not only reduce the number of candidate itemsets, but also significantly increase the performance of the mining process.

Keywords

High Utility Mining, Frequent Itemset Mining, Eliminating Unusual Itemset, Profit, Quantity

1. INTRODUCTION

1.1 Data Mining

Data mining is concerned with analysis of large volumes of data to automatically discover interesting regularities or relationships which in turn leads to better understanding of the underlying processes. The primary goal of Data mining is to discover hidden patterns, unexpected trends in the data. Data mining activities uses combination of techniques from database technologies, statistics, artificial intelligence and machine learning.

Data mining has been used in the analysis of customer transactions in retail market research where it is termed as market basket analysis. Market basket analysis has also been used to identify the purchase patterns of the costumer, which play a key role behind the inception and design of a product.[26]

1.2 Frequent Pattern Mining

Frequent Pattern Mining plays an essential role in many data mining tasks that try to find interesting patterns from databases, such as correlations, sequences, association rules, episodes, classifiers and clusters. Frequent pattern mining is beneficial for association rule mining. Association-rule mining discovers unordered correlations between items from a given database[3,5].

In general, the process of mining association rules can roughly be decomposed into two tasks:

(1) Finding frequent itemsets satisfying a user-specified minimum-support threshold from a given database.

(2) Generating interesting association rules satisfying a user specified minimum confidence threshold from the frequent itemsets found.

1.3 Utility Mining

Frequent itemset mining approach may not always satisfy a sales manager's goal. Because a retail businessman may be interested in identifying its most valuable customers (customers who contribute a major fraction of the profits to the business). The limitations of frequent itemset mining motivated researchers to conceive a utility based mining approach.

2. BACKGROUND

The goal of utility mining is to discover all the itemsets whose utility values are greater than or equal to a user specified threshold in a transaction database. We start from the definition of a set of terms that leads to the formal definition of utility mining problem. Consider a simple transaction table which contain item set and quantity[6,7,8]

Table 1. Transaction table

TID	Transactions
T01	(C,18),(E,1)
T02	(B,6),(D,1),(E,1)
T03	(A,2),(C,1),(E,1)
T04	(A,1),(D,1),(E,1)
T05	(C,4),(E,2)
T06	(B,3),(C,2),(D,1)
T07	(B,10),(D,1),(E,1)
T08	(A,3),(C,25),(D,3)(E,1)
T09	(A,1),(B,1)
T10	(B,6),(C,2),(E,2)

Table 2. Utility table

Items	Profit($)
A	3
B	10
C	1
D	6
E	5

Let

I set of items I={i1, i2, i3……in }

DB database DB={T1,T2,T3……Tn}

Tq is a transaction in DB and is a subset of

I ∈ Tq ∈ DB, Tq∈ I

Let X be a set of items, called an itemset

A k-itemset X has an associated set of transactions in DB, denoted as DBX, where DBX = {Tq ∈ DB | X ∈ Tq ∈I}. For example, in Table 1, DB{C, D} = {T06,T08}.

2.1 Internal Utility

The internal utility value of item ip in transaction Tq, denoted as iu (ip ,Tq), is the value of ip in Tq.

For example, in Table 1, iu (B; T02) = 6.

2.2 External utility

The external utility of item ip in a transaction database, denoted as eu(ip), is the value of ip in the utility table of the database.

For example, in Table 2, eu(C) = 1 and eu (D) = 6.

2.3 Utility value

The utility value of item ip in transaction Tq, denoted as util(ip,Tq), is the product of iu(ip,Tq) and eu(ip). util (ip, T q) = iu (ip, Tq) × eu (ip), where ip ∈ Tq.

For example, in Tables 1 and 2, util (B, T02) = 6 ×10= 60. This can be viewed as when a dealer sells 6 Bs and yields a profit of 10 dollars per item in the transaction T02.

The utility value of itemset X in transaction Tq, denoted as util(X, Tq), is the sum of the utility value of each item of X in Tq, where util(X, Tq)=∑ ip∈ X∈T util(ip, Tq).

For example, in Tables 1 and 2, util ({B; D; E}, T02)

= util (B, T02) + util (D, T02) + util (E , T02)

= 6 × 10 + 6 × 1 + 5 × 1

= 71.

util ({B; D; E}, T02)=71

Table 3. Transaction utility value

Transactions	Transaction utility
T01	23
T02	71
T03	12
T04	14
T05	14
T06	38
T07	111
T08	57
T09	13
T10	72

2.4 Local utility value

The local utility value of an itemset X in DB, denoted as Lutil(X), is the sum of the itemset utility values of X in DBX.

For example, in Table 1,

Lutil({C;D}) =util({C;D},T06)+ util({C;D},T08)

= 8 + 43

= 51.

2.5 Total utility value

The total utility value of DB, denoted as Tutil(DB), is the sum of all transaction utility values in DB.

Tutil (DB) = ∑Tq∈DB util (Tq,Tq).

For example, Tutil(DB)= 425

as shown in Table 3

2.6 Utility value of itemset

The utility value of itemset X in DB, denoted as UTIL(X), is the ratio of the local utility value of X to the total utility value in DB. That is, UTIL(X) = Lutil(X) /Tutil (DB). In other words, UTIL(X) indicates the percentage of the utility value that itemset X contributed in DB.

Table 4.Utility value of one item set

One item set X	Lutil(X)
A	21
B	260
C	52
D	42
E	50

2.7 Minimum local utility

Given a minUtil value, if UTIL(X) ≥ minUtil, the itemset X is a high utility itemset; otherwise X is a low utility itemset. The local utility value of the threshold is called the minimum local utility value, denoted as minLutil. minLutil=minUtil× Tutil(DB).

Consider the transaction database presented in Table 1 and minUtil =30%. Table 4 lists the local utility value and the utility value of each 1-itemset, where Tutil(DB) = 425.

Let X = {B; D; E};

Lutil(X)= util(X,T02)+ util(X,T03)

= 71 + 111

= 182.

Therefore, UTIL(X)= Lutil(X)/ Tutil(DB)

=182/425

$= 42.28\% \geq 30\%.$

The itemset X is a high utility itemset.

3. TWO PHASE METHODS

To address the drawbacks in MEU, Ying Liu Wei-keng Liao Alok Choudhary proposes a novel Two-Phase algorithm that can effectively prune the candidate itemsets and simplify the calculation of utility. Two phase algorithms not only reduces the search space and the memory cost but also reduce computation complexity. In Phase I they define a transaction-weighted upward Closure Property". Those itemsets are High transaction-weighted utilization itemsets are identified in this phase The size of candidate set is reduced by only considering the supersets of high transaction-weighted utilization itemsets. In Phase II, one database scan is performed to filter out the high transaction-weighted utilization itemsets that are indeed low utility itemsets. This algorithm guarantees that the complete set of high utility itemsets will be identified correctly.[10,11,12]

Table 5.With one itemset

One itemset	Transaction-weighted Utilization	Count	High utility item set
A	96	4	N
B	305	5	Y
C	216	6	Y
D	291	5	Y
E	374	8	Y

We can understand the working of Two Phase methods as follow. Consider the transactional database given in table 1 and items utility given in table2. We first find the weight of every transaction

Then start level by transaction-weighted upward Closure Property. Generate one frequent item set and find one high utility item set, then two, three and so on For two candidate item set we use joining for each item with every other item

Table 6 .With two item set

Two itemset	Transaction-weighted Utilization	Count	High utility item set
AB	13	1	N
AC	69	2	N
AD	71	2	N
AE	83	3	N
BC	110	2	N
BD	220	3	Y
BE	254	3	Y
CD	95	2	N
CE	178	5	Y
DE	253	4	Y

For three candidate item set we join two candidate item set with every other two item set

Table 7.With three item set

Three itemset	Transaction-weighted Utilization)	Count	High utility item set
ABC	0	0	N
ABD	0	0	N
ABE	0	0	N
ACD	57	1	N
ACE	69	2	N
ADE	71	2	N
BCD	38	1	N
BCE	72	1	N
BDE	182	2	Y
CDE	57	1	N

From the table5,6,7 it is clear that high utility item set are

{(B), (C), (D), (E) }

{(B, D), (B, E), (C, E), (D, E) }

{ (B, D, E)}[12]

4. RELATED WORK

In 2004 Yao et al defined the problem of utility mining, a theoretical model called MEU, which finds all itemsets in a transaction database with utility values higher than the minimum utility threshold. The Mathematical model of utility mining was defined based on utility bound property and the support bound property. This laid the foundation for future utility mining algorithms [13, 27]

In 2005 Y. Liu, W. Liao, and A. Choudhary proposed Two-Phase algorithm that can discover high utility itemsets with a high efficiency. Utility mining problem is at the heart of several domains, including retailing business, web log techniques, etc. In Phase I algorithm calculate a term transaction-weighted utilization, and proposed the transaction-weighted utilization mining model. In Phase II to filter out the overestimated itemsets. This algorithm requires fewer database scans, less memory space and less computational cost. [14, 15]

In 2006 H. Yao et al formalized the semantic significance of utility measures in. Based on the semantics of applications, the utility-based measures were classified into three categories, namely, item level, transaction level, and cell level. The unified utility function was defined to represent all existing utility-based measures. The transaction utility and the external utility of an itemset was defined and general unified framework was developed to define a unifying view of the utility based measures for itemset mining.[16,25]

In 2008 Alva Erwin1, Raj P. Gopalan, and N.R. Achuthan proposed Efficient Mining of High Utility Itemsets from Large Datasets High utility itemsets mining extends frequent pattern mining to discover itemsets in a transaction database with utility values above a given threshold. Mining high utility itemsets presents a greater challenge than frequent itemset mining. Transaction Weighted Utility (TWU) mining proposed recently

by various researchers, but it is an overestimate of itemset utility and therefore leads to a larger search space. Many proposed algorithm uses TWU with pattern growth based on a compact utility pattern tree data structure. These algorithm implements a parallel projection scheme to use disk storage when the main memory is inadequate for dealing with large datasets. [6,17,18]

In 2010 Vincent S. Tseng1, Cheng-Wei Wu1, Bai-En Shie1, and Philip S. Yu2 proposed UP-Growth: An Efficient Algorithm for High Utility Itemset Mining high utility itemsets from a transactional database. In this paper, they proposed an efficient algorithm, namely UP-Growth (Utility Pattern Growth), for mining high utility itemsets with a set of techniques for pruning candidate itemsets. The information of high utility itemsets is maintained in a special data structure named UP-Tree (Utility Pattern Tree) such that the candidate itemsets can be generated efficiently with only two scans of the database. [18,19]

In 2011 S. Kannimuthu Dr. K. Premalatha proposed the improved version of FUM algorithm, (Improved Fast Utility Mining) iFUM for mining all High Utility Itemsets. The proposed algorithm is compared with existing popular algorithms like UMining and FUM using real life data set. iFUM algorithm is faster than other existing algorithms. iFUM avoid recalculation for generating high utility item set. The iFUM algorithm also scales well as the number of distinct items increases in the input database.[20,21,26]

In 2012 Cheng Wei Wu, Bai-En Shie, Philip S. Yu, Vincent S. Tseng proposed Mining Top-K High Utility Itemsets. They proposed an efficient algorithm named TKU for mining top-k high utility itemsets from transaction databases. TKU guarantees there is no pattern missing during the mining process. The mining performance is enhanced significantly since both the search space and the number of candidates are effectively reduced by the proposed strategies [22,27].

5. PROBLEM STATEMENT

In the literature review there are several approaches have been proposed in recent years, they are unable to solve the problem of producing a large number of candidate itemsets for high utility itemsets. Such a large number of candidate itemsets degrades the mining performance in terms of execution time, search space requirement and in term of memory requirement. The situation may become worst when the database contains lots of transactions or long high utility itemsets.[23,24]

6. PROPOSED METHOD LIMINATING UNUSUAL ITEMS (EUI)

Proposed Eliminating Unusual Items (EUI) as an efficient way of eliminating unusual item set from the transaction to find out high utility item set. Existing level-wise utility mining method including Expected Utility mining (EUM) ,Two Phase methods(TP) ,Sharing Frequent items Set Mining (ShFSM) , Direct Candidate Generation (DCG), and Fast Utility Mining(FUM) generate huge number of candidate. Proposed Eliminating Unusual Items (EUI) methods not only reduce search space but also increase performance. Consider the transactional database given in table 1 and utility of items in table 2..

Table 8 .With one candidate itemset

One itemset	Transaction-weighted Utilization	Count	High utility item set
A	96	4	N
B	305	5	Y
C	216	6	Y
D	291	5	Y
E	374	8	Y

Now from the table 8 we eliminate those item set which has the utility value less than the given minimum utility value in this case item A is deleted from the table. So the remaining item are those item which satisfy the given minimum high utility threshold value.

Table 9. With one high utility itemset

One itemset	Transaction-weighted Utilization	Count	High utility item set
B	305	5	Y
C	216	6	Y
D	291	5	Y
E	374	8	Y

Now for two item set we are using self joining one high utility item set

Table 10.With two candidate itemset

Two itemset	Transaction-weighted Utilization	Count	High utility item set
BC	110	2	N
BD	220	3	Y
BE	254	3	Y
CD	95	2	N
CE	178	5	Y
DE	253	4	Y

Now we delete those two item set which has utility value less than the given minimum utility minimum utility threshold value. So we delete tow item set (B,C),(C,D). Remaining itemset are high utility item set

Table 11.With two high utility itemset

Two itemset	Transaction-weighted Utilization	Count	High utility item set
BD	220	3	Y
BE	254	3	Y
CE	178	5	Y
DE	253	4	Y

For three item set we joining two high utility item set

Table 12.With three candidate itemset

Three itemset	Transaction-weighted Utilization	Count	High utility item set
BCE	72	1	N
BDE	182	2	Y
CDE	57	1	N

Now we delete those item set which has utility value less than the given minimum utility threshold value. So only (B, D, E) satisfy the minimum threshold and is high utility item set. The entire working process of Eliminating Unusual Items (EUI) methods has shown in the figure 1. From the figure it is clear that EUI not only reduce candidate set but also increase performance.

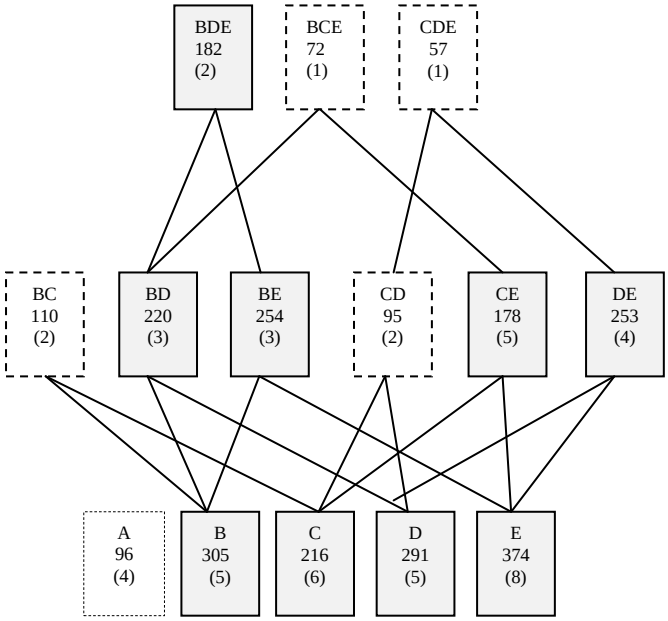

Fig 1: Search space for mining high utility item set using EUI

7. PROPOSED ALGORITHMS

Task: Discovery of High Utility Itemsets

Input: Database DB {Set of Transactions}

Transaction T ∈DB i=1,k=1,

i_p internal utility value of item

H: High utility item set

C_k: Candidate's item set

Minimum Utility value threshold minUtil

Output: High Utility Itemsets H

[1]Begin For each T ∈ DB // scan data DB

[2] Compute the utility value ∀ single itemset

[3] While (Ck !=Null)

[4] Begin For each $i_p \in C_k$ // scan data DB and generate Candidate set

[5] Accumulate ∀ Lutil (i_p)

[6] If Lutil (i_p) ≥ minLutil // high utility

[7] H.add (C);

[8] If Lutil(i_p) ≤ minLutil

[9] $C_k := C_k - i_p$ //delete useless itemset

[10] End

[10] End

[11] End

[11] return (H);

8. EXPERIMENTAL EVALUATION AND PERFORMANCE STUDY

We evaluate the performance of Two-Phase (TP) algorithm and Mining using Expected Utility (MEU) and proposed Eliminating Unusual Items (EUI) by varying the size of the search space. We also analyze the scalability and result accuracy.

All the experiments were performed on a Pentium 3i 2GHz processor 2 GB Main memory, running the window 7 operating system.

The program is implemented in VB.Net version (10). For the database we have used SQL server. Due to its simplicity, we also design simple GUI for user interactions. We use synthetic data and real world data for our evaluation purpose. We are using data set of electronics product for our experiments.

Comparison table between Two Phase Algorithm and Eliminating Unusual Items Algorithms

Table13 .Level by level comparison between TP and EUI

Level	Number of Candidate used by TP	Number of Candidate used by EUI
1	5	5
2	10	6
3	10	3

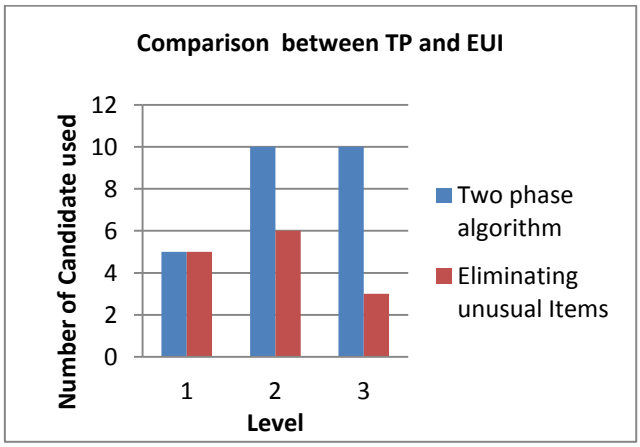

Fig 2: comparison graph between TP and EUI

9. CONCLUSION

Form experimental analysis it is clear that proposed methods EUI methods for mining high utility item set is more efficient as compared to Two Phase method. From the Comparison table between Two Phase Algorithm and Eliminating Unusual Items algorithm it is clear that EUI algorithm generate fewer candidates to find high utility item set as compared to the Two Phase methods. In EUI simple calculation are required where as in Two phase, EUM and other utility mining algorithm uses complex calculation. So form the example it is clear that EUI mining methods perform better.

10. REFERENCES

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In Proc. of the 20th Int'l Conf. on Very Large Data Bases, pp. 487-499, 1994.

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

[3] R. Chan, Q. Yang and Y. Shen. Mining high-utility itemsets. In Proc. of Third IEEE Int'l Conf. on Data Mining, pp. 19-26, Nov., 2003.

[4] Y. L. Cheung, A. W. Fu, Mining frequent itemsets without support threshold: with and without item constraints. IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No. 6, pp. 1052-1069, 2004.

[5] K. Chuang, J. Huang, M. Chen, Mining Top-K Frequent Patterns in the Presence of the Memory Constraint, The VLDB Journal, Vol. 17, pp. 1321-1344, 2008.

[6] A. Erwin, R. P. Gopalan and N. R. Achuthan. Efficient Mining of High-utility Itemsets from Large Datasets. In PAKDD 2008, LNAI 5012, pp. 554-561, 2008.

[7] A. W. Fu, R. W. Kwong and J. Tang, Mining N-Most Interesting Itemsets, In Proc. of ISMIS'00, 2000.

[8] J. Han, J. Pei and Y. Yin. Mining frequent patterns without candidate generation. In Proc. of the ACM-SIGMOD Int'l Conf. on Management of Data, pp. 1-12, 2000.

[9] J. Han, J. Wang, Y. Lu and P. Tzvetkov, "Mining Top-k Frequent Closed Patterns without Minimum Support," In Proc. of ICDM, 2002.

[10] Y. Hirate, E. Iwahashi and H. Yamana, TF2P-Growth: An Efficient Algorithm for Mining Frequent patterns without any Thresholds, In Proc. of ICDM 2004.

[11] H.-F. Li, H.-Y. Huang, Y.-C. Chen, Y.-J. Liu, S.-Y. Lee. Fast and Memory Efficient Mining of High Utility Itemsets in Data Streams. In Proc. of the 8th IEEE Int'l Conf. on Data Mining, pp. 881-886, 2008.

[12] Y. Liu, W. Liao, and A. Choudhary. A fast high-utility itemsets mining algorithm. In Proc. of the Utility-Based Data Mining Workshop, 2005.

[13] Y.-C. Li, J.-S. Yeh and C.-C. Chang. Isolated Items Discarding Strategy for Discovering High-utility Itemsets. In Data & Knowledge Engineering, Vol. 64, Issue 1, pp. 198-217, 2008.

[14] S. Ngan, T. Lam, R. C. Wong and A. W. Fu, Mining N-most Interesting Itemsets without Support Threshold by the COFI-Tree, Int. J. Business Intelligence & Data Mining, Vol. 1, No. 1, pp. 88-106, 2005.

[15] J. Pisharath, Y. Liu, B. Ozisikyilmaz, R. Narayanan, W. K. Liao, A. Choudhary and G. Memik, NU-MineBench version 2.0 dataset and technical report, http://cucis.ece.northwestern.edu/projects/DMS/MineBench.html

[16] T. M. Quang, S. Oyanagi, and K. Yamazaki, ExMiner: An Efficient Algorithm for Mining Top-K Frequent Patterns, ADMA 2006, LNAI 4093, pp. 436 – 447, 2006.

[17] L. Shen, H. Shen, P. Pritchard and R. Topor, Finding the N Largest Itemsets, in Proc. Int'l Conf. on Data Mining, pp. 211-222, 1998.

[18] B.-E. Shie, V. S. Tseng, and P. S. Yu. Online Mining of Temporal Maximal Utility Itemsets from Data Streams. In Proc. of the 25th Annual ACM Symposium on Applied Computing (ACM SAC 2010), 2010.

[19] V. S. Tseng, C.-W. Wu, B.-E. Shie, and P. S. Yu. UP-Growth: an efficient algorithm for high utility itemset mining. In Proc. of Int'l Conf. on ACM SIGKDD, pp. 253–262, 2010.

[20] V. S. Tseng, C. J. Chu, and T. Liang. Efficient mining of temporal high-utility itemsets from data streams. In ACM KDD Workshop on Utility-Based Data Mining Workshop, 2006.

[21] B. Vo, H. Nguyen, T. B. Ho, and B. Le. Parallel Method for Mining High-utility Itemsets from Vertically Partitioned Distributed Databases. In KES 2009, Part I, LNAI 5711, pp. 251-260, 2009.

[22] J. Wang and J. Han, TFP: An Efficient Algorithm for Mining Top-K Frequent Closed Itemsets, IEEE Transactions on Knowledge and Data Engineering, Vol. 17, No. 5, pp. 652-664, May 2005.

[23] H. Yao, H. J. Hamilton, L. Geng, A unified framework for utility-based measures for mining itemsets. In Proc. of ACM SIGKDD 2nd Workshop on Utility-Based Data Mining, pp. 28-37, 2006.

[24] J.-S. Yeh, C.-Y. Chang and Y.-T. Wang. Efficient Algorithms for Incremental Utility Mining. In Proc. of the 2nd Int'l Conf. on Ubiquitous information management and communication, pp. 212-217, 2008.

[25] S.-J. Yen and Y.-S. Lee. Mining High-utility Quantitative Association Rules. In Proc. of 9th Int'l Conf. on Data Warehousing and Knowledge Discovery(DaWaK'2007), Lecture Notes in Computer Science (LNCS) 4654, pp. 283-292, 2007.

[26] S. Kannimuthu , Dr. K. Premalatha iFUM - Improved Fast Utility Mining International Journal of Computer Applications (0975 – 8887) Volume 27– No.11, August 2011.

[27] Yao, H., Hamilton, H.J., Buzz, C.J.: A Foundational Approach to Mining Itemset Utilities from Databases. In: 4th SIAM International Conference on Data Mining. Florida USA (2004)