Enterprise based approach to Mining Frequent Utility Itemsets from Transactional Database

B.Rajasekhara Reddy, M.V.Jaganatha Reddy

Abstract— Data mining can be used extensively in the enterprise based applications with business intelligence characteristics to provide a deeper kind of analysis while meeting strict requirements for administration management and security. Business intelligence is information about a company's past performance that is used to help predict the company's future performance. ARM is a well-known technique in the data mining field, is used to identify frequently occurring patterns of item sets. Although, frequency of occurrence may reflects the statistical correlation between items, and it does not reflect the semantic significance of the items because the user's interest may be related to other factors, such as cost and profit. Utility based itemset mining approach is used to overcome this limitation. This approach identifies itemsets with high utility like high profits. A specialized form of high utility itemset mining is utility-frequent itemset mining which is for considering the business yield and demand or rate of occurrence of the items while mining a retail business transaction database. Such a data mining process will help in mining different types of itemsets of varying business utility and demand such as HUHF, HULF,LUHF and LULF itemsets. Which would significantly helps in inventory control and sales promotion. These itemsets are generated using FUM and FUFM of algorithms which also capable of identifying the active customers of each such type of itemset mined and ranking them based on their total or lifetime business value which would be extremely helpful in improving CRM processes.

Index Terms— Data Mining, Utility and Frequency Based Item set Mining, Customer Relationship Management, High Utility High Frequent Itemsets, High Utility Low Frequent Itemsets, Low Utility High Frequent Itemsets, Low Utility Low Frequent Itemsets

1 Introduction

DATA MINING processes is extensively used on extracting knowledge from large databases irrespective of its significance to the user or the business contexts. The large number of patterns mined during knowledge discovery makes it difficult for the user to understand and identify the patterns that are interesting to user. The need for considering the economic utility of the itemsets in the data mining process has great significance in the business. The quantitative measures like the support measure used in the traditional Association Rules Mining [1,2], which is used to identify frequently occurring patterns of itemsets, reflects only the statistical correlation of items. It does not reflect their semantic significance. Such measures reflecting the statistical correlation may not measure how useful an itemset is in accordance with a user's preferences (i.e., profit). The profit of an itemset depends not only on the support of the itemset, but also on the prices of the items in that develop a utility based itemset mining approach [7], to enable the user to conveniently express his or her preferences centred around the economic usefulness of itemsets as utility values and then find itemsets with utility values higher than a minimum threshold utility value as set by the user.

The paradigm increased the need for considering the business yield (utility) and demand of the items (frequency) while mining a retail business transaction database. Such a data mining process will help in mining different types of itemsets of varying business utility and demand. Data mining has the potential to aid the companies in their quest to become more customers' oriented. It plays a critical role in the overall CRM process[5], which includes interaction with the data mart or warehouse in one direction, and interaction with campaign management software in the other direction. Today the trend is to integrate the data mining and campaign management process in order to gain a competitive advantage .Keeping this in mind, we here present a set of algorithms for mining all types of utility and frequency based itemsets from a retail business transaction database which would significantly aid in inventory control and sales promotion. These set of algorithms are also capable of identifying the active customers of each such type of itemset mined and rank them based on their total business value which would be extremely helpful in improving Customer Relationship Management processes like campaign management and customer segmentation[5]. Utility based data mining[1] is a new research area entranced in all types of utility factors like profit, significance, subjective interestingness, etc., which add economic and business utility to existing data mining processes and techniques. A research area within utility based data mining known as high utility itemset mining is aimed at finding itemsets that interpose high utility. The proposed set of algorithms is built using a combination of efficient Fast Utility Mining (FUM) [3] algorithm and Fast Frequent Utility Mining (FUFM) [3] algorithm. The Fast Utility Mining FUM algorithm employed for finding all high

- *B. Rajasehar Reddy is currently pursuing masters degree program in computer science & engineering in MITS-madanapalli,India. He has one research paper in national conference. His ares of interests include algorithms, information retrieval and data mining .*
  PH:+91-9160758088. E-mail: sehar.brsr@mail.com
- *M.V.Jaganatha Reddy is a Assoc.Professor in CSE dept at MITS-Madanapalli,India. He completed B.E in Electronics and Communication Engineering and M.Tech in Computer Science and engineering.He presented and Published eight papers in international conferences and five papers in international Journals.*
- *E-mail: mvjagannathreddy@yahoo.co.in*

utility itemsets which satisfies the given utility constraint threshold, is faster and simpler than the original UMining algorithm which is based on utility upper bound property.

1.1 Types of interestingness measures

Basically interestingness measures are classified into three types

i. Objective measures
ii. Subjective Measures
iii. Semantic measures

i. Objective measures

Objective measure is based only on the raw data. No knowledge about the user or application is required. Most objective measures are based on the information theory or probability statistics.

ii. Subjective measures

Subjective measure is based on both the data and the user of these data. Defining a subjective measure requires access to the users domain or background knowledge about the data is required. This access can be obtained by interacting with the user during the data mining process or by explicitly representing the user's knowledge or expectations. A Semantic measure considers the semantics and explanations of the patterns. because semantic measures involve domain knowledge from the user, some researchers consider them a special type of subjective measure.

iii. Semantic measures

Subjective measures is the preferred method to integrate the Utility-based measures, where the relevant semantics are the utilities of the patterns in the domain, are the most common type of semantic measure within the data mining algorithm. To use a utility-based approach, the user must specify additional knowledge about the domain. Unlike subjective measures, where the domain knowledge is about the data itself and which is usually represented in the format similar to the discovered pattern. The domain knowledge required for semantic measures does not relate to the user's knowledge or expectations concerning the data. Instead, it represents a utility function that reflects the user's goals. In the preferred method, the utility measures are used to prune uninteresting patterns during data mining process so as to narrow the search space and thus improve mining efficiency.

1.2 Customer Relationship management

The maximization of the life time business value of the entire customer base is the primary objective of Customer Relationship Management. The key to attain this objective is to understand the behavior of the customer. Clear understanding of customer behavior required for the customer segmentation and compensation management.

2 Overview of The Existing System

The traditional Association Rule Mining is used to identify frequently occurring patterns of item sets. ARM model treats all the items in the database equally by only considering if an item is present in a transaction or not. However, frequency of occurrence may not express the semantics of applications, because the user's interest may be related to other factors, [3]such as cost, profit, or aesthetic value. These limitations motivated us to develop a utility based itemset mining approach, which allows a user to conveniently express his or her perspectives concerning the usefulness of itemsets as utility values and then find itemsets with utility values higher than a threshold. In this section describes existing algorithms, techniques used for utility item set mining. Some of the algorithms are shown below.

2.1 Umining algorithm

UMining algorithm is used for mining all high utility itemsets using pruning strategy.The framework of the UMining algorithm is shown below. The functions called by the UMining algorithm are Scan, CalculateAndStore, Discover, Generate, and Prune. The **Scan** function finds the set of all items in the transaction database T. The **CalculateAndStor*e*** function accesses transaction database T to calculate the actual utility value of each k-itemset in C_k. It is assumed that each itemset S in C_k has associated with it a u field, denoted u(S), for storing its utility value. The **Discover** function selects all high utility itemsets in candidate set C_k. The **Generate** function generates all possible candidate k-itemsets from the (k-1) itemsets in C_{k-1}. The **Prune** function calculates the utility upper bound of each itemset in C_k based on the utility values of the candidate itemsets in C_{k-1}, and then re[4]moves any itemset with a utility upper bound less than minutil from C_k. The UMining algorithm follows the basic framework of the Apriori algorithm, but there are significant differences in three sub functions (Prune, CalculateAndStore, and Generate functions).

```
Algorithm Umining((T,f.minUtil,k)
Input: Transaction Database T
Utilty function f
Utility value threshold minutil
Maximum size of itemset K
Output: A set of high utilty itemsets H
{
    I= Scan(T);
    $C_1$=I;
    k=1;
    $C_k$=CalculateAndStore($C_{k,,}$ T, f);
    H=Discover($C_{k,}$ minUtil);
    while (| $C_k$ |> 0 and k<= k)
    {
      k=k+1;
      $C_{k=}$Generate($C_{k-1}$ ,I)
      $C_{k=}$Prune($C_k$,$C_{k-1}$ ,minUtil)
      $C_{k=}$CalculateAndStore($C_k$,T,fl)
      H = H U Discover($C_k$,minUtil)
    }
    return HUI;
}
```

Figure. 1. Pseudo code of the UMining algorithm

2.2 FUM algorithm

FUM algorithm generates high utility itemsets using Combination Generator. It is simpler and executes faster than Umining algorithm, when more itemsets are identified as high utility itemsets.

Algorithm FUM(DB, T, minUtil)
Task: Discovery of High Utility Itemsets
Input: Database DB {Set of Transactions}
Transaction T ∈ DB
Utility value threshold minUtil
Output: High Utility Itemsets HUI

```
Compute the utility value ∀ single itemset
For each T ∈ DB
{
  if T ∉ S  {where S ⊆ DB | S = [0 .. T-1]}
  {
    Candidateset = CombinationGenerator(T)
    For each C ∈ CandidateSet
    {
      if ( C ∉ H ) ∧ (U(C,T) ≥ minUtil)
      H.add (C);
    }
  }
}
return (HUI);
```

Figure. 2. Pseudo code of the FUM algorithm

Combination Generator(T)

Generate all possible combinations of itemset ∈ T

The miss rate of FUM algorithm is nil when compared to that of the UMining algorithm. In these ways, in a domain driven data mining application, FUM algorithm fits itself perfectly by mining all the High Utility Itemsets(HUI) with maximum speed, accuracy, reliability and cost effectiveness

2.3 Problem identification

Utility mining aims at identifying the itemsets with high utilities. The practical usefulness of the high utility itemset mining is limited by the significance of the discovered item sets. A huge number item sets that are not interesting to the user are often generated. For example, there may be thousands of combinations of products that occur in 1% of the transactions. [4] If too many uninteresting frequent item sets are found, the user is forced to do additional work to select the item sets that are indeed interesting. That is when clients(companies) seek for the combination of products which can generate different values of utility and frequency the association rule mining (ARM) or the utility mining will not achieve such task. ARM mines frequent itemsets without knowing the producing profit. On the other hand, the utility mining seeks high profit items but no guarantee the frequency.

3 Solution to the Proposed Problem

3.1 Design Of EFUFM

Umining and FUM algorithms generates high utility item sets, FUFM algorithm generates high utility and high frequent item sets. FUFM algorithm finds High Utility High frequency (HUHF) items. From the discussions with our clients, we learnt and understood their knowledge requirements. Using HUHF itemsets generated by the FUFM algorithm and the High Utility Itemsets (HUI) generated using our FUM algorithm, we have generated three new itemsets namely High utility and low frequency itemsets (HULF), Low utility and high frequency itemsets (LUHF) and Low utility and low frequency itemsets (LULF). Based on our experiments, we have proposed system architecture for the generation of different kinds of utility frequent itemsets which is self explanatory as in Figure. 3. Customer Relationship Management is incorporated into the system by generating a list of customers who are frequent buyers of these four different kinds of item sets.

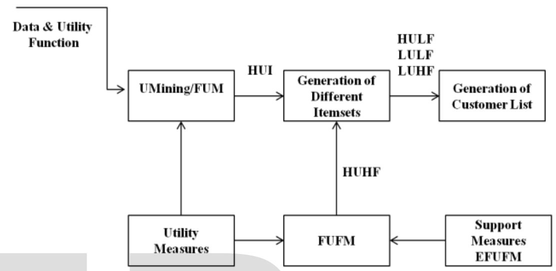


Figure. 3. Proposed system architecture design of EFUFM

3.2 Generation of Different Utility Frequent itemsets

3.2.1 Mining of the HUHF itemsets

We made our efforts to incorporate support consideration in the FUM algorithm in order to mine the HUHF itemsets from a transaction database. In the Fast Utility-Frequent Mining(FUFM) Algorithm, the High Utility Itemsets (HUI) are mined using the FUM algorithm initially. Then for each HUI mined, corresponding support is calculated and checked with the minimum support threshold (minSup). Those HUI with support greater than the minSup are added to the HUHF itemset list.

Fast Utility-Frequent Mining(FUFM) Algorithm:

FUFM finds all utility-frequent itemsets within the given utility and support constraints threshold. Utility-frequent itemsets are a special form of high utility itemsets. For a given utility threshold μ each itemset S is associated with a set of transactions defined as $_{S,\mu} = \{T \mid S \subseteq T \wedge u(S, T) \geq \mu \wedge T \in DB \}$ On the basis of this set of transactions an extended support measure can be identified as support(s,μ)= |Ts,μ/DB|

Itemset S is utility-frequent if for a given utility threshold μ and support threshold s the extended support measure support(S, μ) is greater or equal to s.The support measure is always greater or equal to the extended support measure. Proof is trivial because when computing extended support we count only those transaction containing given itemset S that also gives minimum utility on S, but when computing "ordinary" support we count all transactions containing S. The practical consequence of this statement is that frequent itemset mining algorithms can be used to mine utility-frequent itemsets.

Algorithm FUFM(DB, minUtil, minSup**)**
Task: Discovery of Utility Frequent Itemsets
Input: Database DB,Constraints minUtil and minSup
Output: High Utility High Frequent itemsets

```
[1]  L = 1
[2]  Find the set of candidates of length L with support >= minSup
[3]  Compute extended support for all candidates and Output utility frequent itemsets
[4]  L += 1
[5]  Use the frequent itemset mining algorithm to obtain new set of frequent candidates of length L from the old set of frequent candidates
[6]  Stop if the new set is empty otherwise go to step[3]
```

Figure. 4. Pseudo code of the FUFM(HUHF itemset set) Mining Algorithm

3.2.2 Mining of the HULF itemsets

HULF Itemset Mining Algorithm

This algorithm is designed to generate high utility and low frequent itemsets. It follows the basic frame work of FUM and FUFM algorithms. The first phase is to generate high utility itemsets using FUM algorithm. In the second phase high utility high frequent itemsets are generated using FUFM. Then using set difference function high utility low frequent itemsets are generated from HU and HUHF.

Algorithm HULF Itemset Mining( )
Task: Discovery of High Utility and Low Frequent itemsets
Input: Database DB Constraints minUtil and minSup
Output: High Utility and Low Frequent Itemsets (HULFI)

```
[1]  Compute High utility item sets HU using FUM algorithm.  didates Compute High utility and high frequent item sets HUHF using FUFM algorithm.
[2]  HULF = HU \ HUHF /* Set Difference */
[3]  Return (HULF).
```

Figure. 5. Pseudo code of the HULF Mining Algorithm

3.2.3 Mining of the LUHF itemsets

LUHF Itemset Mining Algorithm:

This algorithm is designed to generate Low utility and high frequent itemsets. It follows the basic frame work of FUFM algorithm.

Algorithm LUHF itemset Mining( )
Task: Discovery of Low Utility and High Frequent Itemsets
Input: Database DB Constraints minUtil and minSup
Output: Low Utility and High Frequent Itemsets (LUHF)

```
[1]  L = 1
[2]  Find the set of candidates of length L with support >= minSup
[3]  Compute extended support where TS,μ = {T|S ⊆ T ∧ u(S, T) < μ ∧T ∈ DB } for all candidates and output low utility high frequent itemsets
[4]  L+=1
[5]  Use the frequent itemset mining algorithm to obtain new set of frequent candidates of length L from the old set of frequent candidates.
[6]  Stop if the new set is empty otherwise go to [3]
[7]  Return (LUHF)
```

Figure. 6. Pseudo code of the LUHF Mining Algorithm

3.2.4 Mining of the LULF itemsets

LULF Itemset Mining Algorithm

This algorithm is designed to generate Low utility and low frequent itemsets. There are two phases in this algorithm. In the first phase using exhaustive search low utility itemsets are determined. In the second phase, using set difference function low utility low frequent itemsets are generated from LU and LUHF. Thus we generate Low utility and low frequent itemsets.

Algorithm LULF Itemset Mining( )
Task: Discovery of Low Utility and Low Frequent Itemsets
Input: Database DB Constraints minUtil and minSup
Output: Low Utility and Low Frequent Itemsets (LULF)

```
[1]  Compute the utility value ∀ single itemset
[2]  For each T ∈ DB
[3]  begin
[4]  if T ∉ S  {where S ⊆ DB | S = [0 .. T-1]}
[5]  begin
[6]    Candidateset =   CombinationGenerator (T)
[7]        For each C ∈ CandidateSet
[8]          begin
[9]            if ( C ∉ H ) ^ U(C,T) < minutil )
[10]           LU.add (C);
[11]         end
[12]     end
[13] end
[14] LULF=LU-LUHF /* set difference*/
[15] return (LULF)
```

Figure. 7. Pseudo code of the HULF Mining Algorithm

3.3 Ranked Customer List Generation for different kinds of itemsets

The method of generating the ranked customer list for each type of itemsets is similar method. The customer who buys the maximum number of items of a particular type of itemset will have the highest customer value with regard to that type of itemset and hence will be ranked at the top in the respective category.

3.3.1 HUHFI customer list generation Algorithm

This algorithm is designed to generate and rank customers buying Low utility and high frequent itemsets. **Algorithm** HUHFI CustomerListGeneration(HUHFI)

Task: Generation of HUHFI customer list.
Input: Database DB, Constraints minUtil and minSup
Output: Ranked list of customers who buy HUHF items

```
[1]  Compute High utility and high frequent (HUHF) itemsets using FUFM algorithm
[2]  For each I ∈ HUHF itemset, scan the database DB to find the customers who buy that itemset
[3]  Increment the count value associated with the customer who is a buyer of I.
[4]  Stop if the HUHF is empty else Go to [2]
[5]  List the HUHF customers in descending order of the count value associated with each customer
[6]  return (list of HUHF customers)
```

Figure. 8. Pseudo code of the HUHFI customer list generation algorithm

3.3.2 HULFI customer list generation Algorithm

This algorithm is designed to generate and rank customers buying High utility and Low frequent itemsets.

Algorithm HULFI CustomerListGeneration(HULFI)
Task: Generation of HULFI customer list.
Input: Database DB, Constraints minUtil and minSup
Output: Ranked list of customers who buy HULF items

```
[1]  Compute High utility and high frequent (HULF) itemsets using FUFM algorithm
[2]  For each I ∈ HULF itemset, scan the database DB to find the customers who buy that itemset
[3]  Increment the count value associated with the customer who is a buyer of I.
[4]  Stop if the HULF is empty else Go to [2]
[5]  List the HULF customers in descending order of the count value associated with each customer
[6]  return (list of HULF customers)
```

Figure. 9. Pseudo code of the LULFI customer list generation algorithm

3.3.3 LUHFI customer list generation Algorithm

Algorithm LUHFI CustomerListGeneration(LUHFI)
Task: Generation of LUHFI customer list.
Input: Database DB, Constraints minUtil and minSup
Output: Ranked list of customers who buy LUHF items

```
[1]  Compute High utility and high frequent (LUHF) itemsets using FUFM algorithm
[2]  For each I ∈ LUHF itemset, scan the database DB to find the customers who buy that itemset
[3]  Increment the count value associated with the customer who is a buyer of I.
[4]  Stop if the LUHF is empty else Go to [2]
[5]  List the LUHF customers in descending order of the count value associated with each customer
[6]  return (list of LUHF customers)
```

Figure. 10. Pseudo code of the LUHFI customer list generation algorithm

3.3.4. LULFI customer list generation Algorithm

This algorithm is designed to generate and rank customers buying Low utility and Low frequent itemsets.

Algorithm LULFI CustomerListGeneration(LULFI)
Task: Generation of LULFI customer list.
Input: Database DB, Constraints minUtil and minSup
Output: Ranked list of customers who buy LULF items

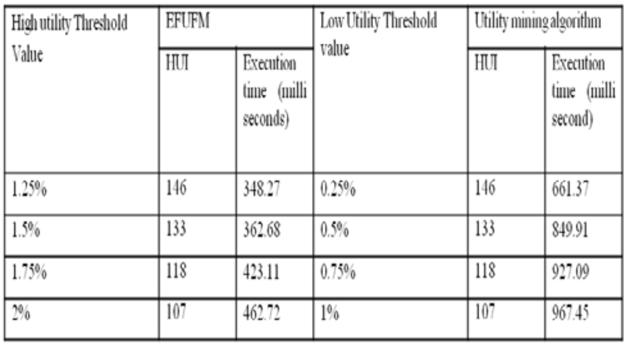
```
[1]  Compute High utility and high frequent (LULF) itemsets using FUFM algorithm
[2]  For each I ∈ LULF itemset, scan the database DB to find the customers who buy that itemset
[3]  Increment the count value associated with the customer who is a buyer of I.
[4]  Stop if the LULF is empty else Go to [2]
[5]  List the LULF customers in descending order of the count value associated with each customer
[6]  return (list of LULF customers)
```

Figure.11. Pseudo code of the LULFI customer list generation algorithm

3 Results And Discussion

The UMining and FUM algorithms are for mining all high utility item sets with low threshold value. The proposed method gives a new perspective in analyzing the item sets and this method scan the database with high threshold value. It is faster than the existing algorithm. The item sets that are both high frequent and high utility can be obtained using this method. From the basic framework of these algorithms the different kinds of item sets namely high utility high frequent, high utility low frequent, low utility high frequent and low utility low frequent item sets are generated. Then Customer Relationship Management is incorporated into the system by tracking the customers who are frequent buyers of the different kinds of item sets.

Table 1: Performance of Umining verses EFUFM algorithm



High utility Threshold Value	EFUFM		Low Utility Threshold value	Utility mining algorithm	
	HUI	Execution time (milli seconds)		HUI	Execution time (milli second)
1.25%	146	348.27	0.25%	146	661.37
1.5%	133	362.68	0.5%	133	849.91
1.75%	118	423.11	0.75%	118	927.09
2%	107	462.72	1%	107	967.45

following figures shows the execution time of Umining algorithm for Low threshold Value, Execution time of EFUFM algorithm for High Threshold Value.

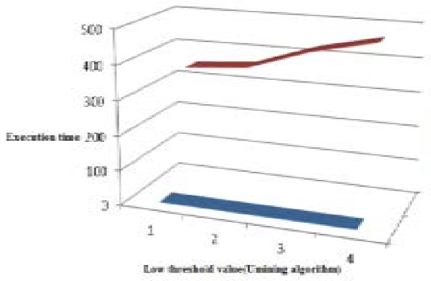


Figure12. Umining Algorithm Performance Comparison for Low Threshold Value

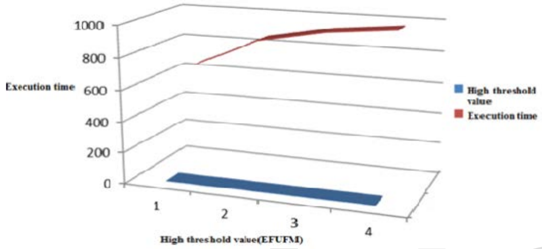


Figure.13 EFUFM Algorithm Performance Comparison for High Threshold Value

4 Importance of the Proposed Approach

The generation of different types of itemsets based on their business utility and rate of recurrence in the transactions made by the customer can have greate importance in the enterprise applications for inventory control and sales promotion. Identifying the corresponding customers will greatly benefit in customer segmentation, campaign management and Customer Relationship Management. Inventory control is a function of materials management, and the objective of inventry control is to keep the total cost associated with the system to be a possibly minimum [4]. Out of thousands of items held in an inventory of a typical organization,only a small percentage of them deserve management's closest attention and tightest control [4].The High Utility and High Frequency Itemsets like say Basumati Rice can be stocked to a greater extent in order to meet the greaterdemand. Stringent safety measures can be followed in storing such items. This will gurenty definite and increased profits.The Low Utility and Low Frequency Items are also of importance to the sales and inventory management. From the view of inventory control, LULF items can be stocked less. When there is an excess availability of LULF items, they can be given as premium gifts for the active customers who purchase the HUHF items. Result of thus, the sales of HUHF items can also be increased.Moreover the organization will be able to provide premium service to the highly valuable active customers thereby strengthening the customer loyalty and such customer retention.There is a possibility that customers who buy HUHF itemsets may become responders for LULF itemsets if they feel that specific LULF items received as gifts are really worth useful to them.Such customers will start using the LULF items. Thus, its demand will increase automatically thereby increasing the business value. This also could benefit the business positively. Certain High Utility and Low Frequency Itemsets can be of similar value and at times of greater value than the HUHF items. For example gems like diamonds, platinum etc may not be bought by the customers frequently. But such items have enormous business value. Identifying active customers of such HULF items using the proposed approach will be simple and efficient in segmenting the most valuable customers and designing a customized campaign management programme for such customers [5].Advantage of LUHF itemset generation is that, it will increase therevenue of the business substantially. LUHF items are the mostcommonly purchased items.

5 Conclusion

The UMining and FUM algorithms are for mining all high utility item sets with low threshold value. The proposed method gives a new perspective in analyzing the item sets and this method scan the database with high threshold value[7]. It is faster than the existing algorithm. From the basic framework of these algorithms the different kinds of item sets namely HUHF,HULF,LULF and LUHF item sets are generated. Then Customer Relationship Management is incorporated into the system by tracking the customers who are frequent buyers of the different kinds of item sets.The enterprise business also discussed.

6 References

[1] Vincent S. Tseng, Bai-En Shie, Cheng-Wei Wu, and Philip S. Yu, Fellow, " Efficient Algorithms for Mining High Utility Itemsets from Transactional Databases"IEEE transactions on knowledge and data engineering 2012.

[2] Agrawal R, Srikant R, 'Fast algorithms for mining association rules', Proceedings of 20th International Conference on Very Large Databases, Santiago, Chile, pp.487–499, 1994.

[3] S. Shankar, T.P.Purusothoman, S.Jayanthi and N.Babu, "A Fast Algorithm for Mining High Utility Item sets", Proceedings of IEEE International Advance Computing Conference (IACC 2009), Patiala, India, pages : 1459 – 1464

[4] Pradip Kumar Bala, "Mining Association Rules for Selective Inventory Control", Journal of the Academy of Business and Economics, Feb 2008 .

[5] Richard Boire, "Data Mining for Customer Loyalty", Direct Marketing, March 2009.

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

[7] H.Yao, H. J. Hamilton, and C. J. Butz, "A Foundational Approach to Mining Itemset Utilities from Databases", Proceedings of the Third SIAM International Conference on Data Mining, Orlando, Florida, pp. 482-486, 2004.