Study on High Utility Itemset Mining

Nilovena.K.V[1], Anu.K.S[2]

[1, 2]KMCT College of Engineering, Calicut University, Kerala, India

Abstract: ***Data mining is the process of mining new non trivial and potentially valuable information from large data basis. Data mining has been used in the analysis of customer transaction in retail research where it is termed as market basket analysis. Earlier data mining methods concentrated more on the correlation between the items that occurs more frequent in the transaction. In frequent itemset mining they do not consider the utility or importance of an item. The limitations of frequent items at mining led to a emerging area called utility mining. In utility items at mining the usefulness or profit of an item is considered. The term utility means the importance or profit of an item in a transaction. The main objective of high utility items at mining is to find the item set having utility values above the given threshold. In this paper we present a literature study on various mining algorithms.***

Keywords: Data mining, Frequent itemset mining, Candidate set, Utility mining.

1. Introduction

Data mining is the process of mining non-trivial, formerly extraordinary, previously unknown and potentially valuable information from large databases. It is also concerned with analysis of large amount of data to discover interesting regularities or relationships which in turn leads to better understanding. Thus data mining refers to extracting or mining knowledge from large amounts of data. Data mining activities uses combination of techniques from database technologies, artificial intelligence, machine learning etc and the application areas which include this are includes bioinformatics, genetics, medicine, clinical research, education, retail and marketing research.

Frequent itemset mining [3],[13],[18],[19] is a fundamental research topic in data mining. Frequent itemsets are the itemsets that appear frequently in the transactions. The goal of frequent itemset mining is to identify all the itemsets in a transaction dataset which occurs frequently. In Data Mining the task of finding frequent pattern from large databases is very useful in many applications over the past few years. This task is computationally more expensive, especially when a large number of patterns exist and the large number of patterns which are mined during the various approaches makes the user very difficult to identify the patterns which are very interesting for the user. Data mining has been considerably used in the analysis of customer transaction in retail research. One of its popular application is market basket analysis, which refers to the discovery of itemsets that are frequently purchased by customers.

In the real world each item in the supermarket has a different importance/price and single customer will be interested in buying multiple copies of same item. Therefore, finding only traditional frequent patterns in a database cannot fulfill the requirement of finding the most valuable itemsets that contribute the most to the total profit in a retail business. So we go for utility mining.
In utility mining[2] ,[4] ,[7]each item has a unit weight and can appear more than once in a transaction. The term utility refers to the importance or the usefulness of the appearance of the itemset in the transaction quantified in terms of profit, sales or any other user preference. A transaction database consists of two measures such as internal utility and external utility. Quantity of a product present in a particular transaction is called the internal utility and the profit value of a product in each transaction is called external utility. The utility of itemset is defined as the product of external utility and internal utility.

Mining high utility itemsets from databases refers to finding the itemsets with high profits. High utility itemsets mining has become one of the most interesting data mining tasks with broad applications and it identifies itemsets whose utility satisfies a given threshold. An itemset is called a high utility itemset if its utility is not less than a user specified minimum utility threshold value; else that itemset is treated as a low utility itemset.

2. Literature Survey

R. Agrawal et al in [3] proposed Apriori algorithm and it is used to obtain frequent itemsets from the database The first pass of the algorithm it simply counts occurrences of each item to determine the large 1-itemsets. First it generates the candidate sequences and then it chooses the itemsets having support count greater than the minimum support count from the candidate ones. The second step involves generating association rules from frequent itemsets.Thus algorithm generate (k+1) candidate itemsets from length k frequent itemset. Algorithm terminate when no frequent or candidate set can be generated.Apriori is a classic algorithm for frequent itemset mining and association rule learning over transactional databases. After identifying the large itemsets, only those itemsets are allowed which have the support greater than the minimum support allowed. Apriori Algorithm generates lot of candidate item sets and scans database every time. When a new transaction is added to the database then it should rescan the entire database again.

J. Han et al in [8] proposed frequent pattern tree (FP-tree) structure, an extended prefix tree structure for storing crucial information about frequent patterns, compressed and develop an efficient FP-tree based mining method is Frequent pattern tree structure. Pattern fragment growth mines the complete set of frequent patterns using the FP-growth. It constructs a highly compact FP-tree, which is usually substantially smaller than the original database, by which costly database scans are saved in the subsequent mining processes. It applies a pattern growth method which avoids costly candidate generation.

In FP tree method, scan the transaction DB for the first time, find frequent items (single item patterns) and order them into a list L in frequency descending order. In the second database scan, construct FP-tree by putting each frequency ordered transaction onto it. Then construct conditional pattern base for each item in the header table and conditional FP-tree from each conditional pattern base. Recursively mine conditional FP-trees and grow frequent patterns obtained so far. If the conditional FP-tree contains a single path, simply enumerate all the patterns. FP-growth is not able to find high utility itemsets.

Y.Liu, W-K.Liao, A.Choudhary [9] proposed a two phase algorithm which was developed to find high utility itemsets, using the download closure property of apriori. The algorithms have defined the transaction weighted utilization while maintaining the download closure property. In this paper they defined two database scans. In the first database scan, the algorithm finds all the one-element transaction-weighted utilization itemsets and its results form the basis for two element transaction weighted utilization itemsets. In the second database scan, the algorithm finds all the two element transaction-weighted utilization itemsets and it results in three element transaction weighted utilization itemsets. The drawback of this algorithm is that it suffers from level wise candidate generation and test methodology.

Although two-phase algorithm reduces search space by using TWDC property, it still generates too many candidates to obtain HTWUIs and requires multiple database scans. To overcome this problem, Li et al. [5] proposed an isolated items discarding strategy (IIDS) to reduce the number of candidates. Y-C. Li, J-S. Yeh and C-C. Chang proposed an isolated item discarding strategy (IIDS). In this paper, they discovered high utility itemsets and also reduced the number of candidates in every database scan. They retrieved efficient high utility itemsets using the mining algorithm called FUM[8] and DCG+[9]. In this technique they showed a better performance than all the previous high utility pattern mining technique. However, their algorithms still suffer with the problem of level wise generation and test problem of apriori and it require multiple database scans.

Ahmed CF et al [10] developed HUC-Prune. In the existing high utility pattern mining it generate a level wise candidate generation and test methodology to maintain the candidate pattern and they need several database scans which is directly dependent on the candidate length. To overcome this, they proposed a novel tree based candidate pruning technique called HUC-tree, (high utility candidate tree) which captures the important utility information of transaction database. HUC-Prune is entirely independent of high utility candidate pattern and it requires three database scans to calculate the result for utility pattern. The drawback of this approach is that it is very difficult to maintain the algorithm for larger database scan regions.

To efficiently generate HTWUIs in phase I and avoid scanning database too many times, Ahmed et al. [4] proposed a tree-based algorithm, named IHUP. A treebased structure called IHUP-Tree is used to maintain the information about itemsets and their utilities. Each node of an IHUP-Tree consists of an item name, a TWU value and a support count. IHUP algorithm has three steps: 1) construction of IHUP-Tree, 2) generation of HTWUIs, and 3) identification of high utility itemsets. This framework may produce too many HTWUIs in phase I since the overestimated utility calculated by TWU is too large. Such a large number of HTWUIs will degrade the mining performance in phase I substantially in terms of execution time and memory consumption. Moreover, the number of HTWUIs in phase I also affects the performance of phase II since the more HTWUIs the algorithm generates in phase I, the more execution time for identifying high utility itemsets it requires in phase II. As stated above, the number of generated HTWUIs is a critical issue for the performance of algorithms.

Cheng-Wei Wu [2] presented a novel algorithm with a compact data structure for efficiently discovering high utility itemsets from transactional databases. The UP-Growth is one of the efficient algorithms to generate high utility itemsets depending on construction of a global UP-Tree. In phase I, the framework of UP-Tree follows three steps: (i). Construction of UP-Tree. (ii). Generate PHUIs from UP-Tree. (iii). Identify high utility itemsets using PHUI The construction of global UP-Tree is follows,

1) Discarding global unpromising items (i.e., DGU strategy) is to eliminate the low utility items and their utilities from the transaction utilities.
2) Discarding global node utilities (i.e., DGN strategy) during global UP-Tree construction. By DGN strategy, node utilities which are nearer to UP-Tree root node are effectively reduced. The PHUI is similar to TWU, which compute all itemsets utility with the help of estimated utility. Finally, identify high utility itemsets.

Even the numbers of candidates in Phase 1 are efficiently reduced by DGU and DGN strategies. But during the construction of the local UP-Tree (Phase-2) they cannot be applied. For discarding utilities of low utility items from path utilities of the paths DLU strategy should be used instead of it and for discarding item utilities of descendant nodes during the local UP-Tree construction DLN strategy should be used. Even though the algorithm is facing still some performance issues in Phase-2.

In FIM,to reduce the computational cost of the mining task and present fewer but more important patterns to users, many studies focused on developing concise representations, such as free sets [20], non-derivable sets [21], odds ratio patterns [25], disjunctive closed itemsets [26], maximalitemsets [16] and closed itemsets [11],[18]. These representations successfully reduce the number ofitemsets found, but they are developed for FIM instead of HUI mining.

So Cheng-Wei Wu, Philippe Fournier-Viger, Philip S. Yu, Fellow, IEEE, Vincent S. Tseng proposed a condensed and meaningful representation of HUIs named Closed+ High Utility Itemsets(CHUIs)[1], [6] which integrates the concept of closed itemset into high utility itemset mining.Three effcient algorithms named AprioriHC (Apriori-based algorithm for mining High utility Closed+ itemset), AprioriHC-D (AprioriHC algorithm with Discarding unpromising and isolated items) and CHUD (Closed+ High Utility itemset Discovery)is used to find this representation. The AprioriHC and AprioriHC-D algorithms employs breadth first search to find CHUIs and inherits some nice properties from the well-known Apriori algorithm [3]. The CHUD algorithm includes three novel strategies named REG, RML and DCM that greatly enhance its performance.

The Close algorithm[18] is presented by Nicolas Pasquier, Yves Bastide, Rafik Taouiland Lotff Lakhal for an effcient mining of association rules using closed itemsets lattices. Close is also another Apriori-like algorithm which is directly mines frequent closed itemsets. At the start of thisalgorithm is to use bottom-up search technique to identify the smallest frequent itemset that determines a closed itemset. After finding the frequent k-itemsets, Close compares the support of each itemset set with its subsets at the previous level. When the support of an itemsets matches the support of any of its subsets, the itemsets is pruned. After that in Close algorithm is to calculate the closure of all the itemsets which is found at the very first step.The CLOSET algorithm[22] is a very efficient but highly complex technique.

The CLOSET algorithm was designed to extract frequent closed itemsets from large databases. It reduces both the computational and cognitive cost in association rule analysis, by limiting the results to just frequent closed itemsets. These algorithms start with scanning for frequent items. The algorithmthat divides the frequent items by finding just the frequent closed itemsets. The CLOSET technique continues by recursively mining the subsets of the frequent item closed sets. The algorithm then effectively creates conditional databases of the frequent closed-items separately from the initial transactional database. The actual mechanics of this process can become little complex.

The DCI Closed[12]a fast and memory effcient algorithm to mine frequent closed itemsets ispresented by Claudio Lucchese, Salvatore Orlando, Raffaele Perego. One of the main problem is occurs at the time of mining the frequent closed itemsets is the duplication. In this algorithm ageneral technique is used for find out and removes the duplicate closed itemsets, without the need of storing the whole closed itemsets in main memory. This is one of the very important features of this algorithm their approach can be exploited with substantial performance benefits by any algorithm that adopts a vertical representation of the dataset. This algorithm contains mainly three functions CLOSED SET, PRE SET, POST SET. From CLOSED SET new closed set, new generators and corresponding closed sets can be building. While the composition of POST SET guarantees that the various generators will be produced according to the lexicographic order.

Based on literature survey,it has been found that many of the previous systems considers thefrequency of the itemsets. Many studies [2],[4],[14],[15],[17],[23],[24] has been proposed for mining HUIs, but they often present a large number of high utility itemsets to users. A very large number of high utility itemsets makes it diffcult for the users to comprehend the results. It may also cause the algorithms to become ineffcient in terms of time and memory requirement, or even run out of memory. It is widely recognized that the more high utility itemsets the algorithms generate, the more processing they consume. The performance of the mining task decreases greatly for low minimum utility thresholds or when dealing with dense databases[12],[16],[18],[19]. The proposed system considers the mining of closed high utility itemsets. The two effective strategies DGU and IIDS reduce the number of candidates significantly.

3. Proposed System

Problem statement : Given a transactional database D and a user specified minimum utility threshold minimumutility, the problem of mining high utility itemsets from D is to find the complete set of the itemsets whose utilities are larger than or equal to minimum utility.

To address this issue, we propose two novel algorithms as well as a compact data structure for effciently discovering high utility itemsets from transactional databases. Major contributions of this work are summarized as follows:

1.Two algorithms, named Apriori algorithm and utility pattern growth (UPGrowth) , and a compact tree structure, called utility pattern tree (UP-Tree), for discovering high utility itemsets and maintaining important information related to utility patterns within databases are proposed. High-utility itemsets can be generated from UP-Tree effciently with only two scans of original databases.

2. Several strategies are proposed for facilitating the mining processes of UP-Growth and Apriori algorithm by maintaining only essential information in UP-Tree. By these strategies, over- estimated utilities of candidates can be well reduced by discarding utilities of the items that cannot be high utility or are not involved in the search space. The proposed strategies can not only decrease the overestimated utilities of PHUIs but also greatly reduce the number of candidates.

Mining high utility itemsets from databases refers to finding the itemsets with high profits. Here, the meaning of itemset utility is interestingness, importance, or profitability of an item to users. Utility of items in a transaction database consists of two aspects: 1) the importance of distinct items, which is called external utility, and 2) the importance of items in transactions, which is called internal utility. Utility of an itemset is defined as the product of its external utility and its internal utility. An itemset is called a high utility itemset if its utility is no less than a user-specified minimum utility threshold; otherwise, it is called a low-utility itemset. An itemset $X \subseteq L$ is a closed itemset iff there exists no itemset $Y \subseteq L$ such that (1) $X \subseteq Y$ and (2) support count (X) = support count (Y). Otherwise, X is non-closed itemset.

In the first scan, TU of each transaction is computed. At the same time, TWU of each single item is also accumulated. By TWDC property, an item and its supersets are unpromising to be high utility itemsets if its TWU is less than the minimum utility threshold. Such an item is called an unpromising item. An item is called a promising item if its TWU value is greater than the minimum utility. Otherwise it is called an unpromising item.During the second scan of database, transactions are inserted into a UP-Tree. When a transaction is retrieved, the unpromising items should be removed from the transaction and their utilities should also be eliminated from the transactions TU. Only the supersets of promising items are possible to be high utility itemsets. Thisconcept forms the first strategy called DGU. Discarding global unpromising items and their actual utilities from transactions and transaction utilities of the database. The unpromising items play no role in closed high utility itemsets mining. Thus, when utilities of itemsets are being estimated, utilities of unpromising items can be regarded as irrelevant and be discarded.

New TU after pruning unpromising items is called reorganized transaction utility (RTU). By reorganizing the transactions, not only less information is needed to be recorded in UP-Tree, but also smaller overestimated utilities for itemsets are generated. Since the utilities of unpromising items are excluded, RTU must be no larger than TWU. Therefore, the number of PHUIs with DGU must be no more than that of HTWUIs generated with TWU [4], [9]. DGU is quite effective especially when transactions contain lots of unpromising items, such as those in sparse data sets.

The algorithm AprioriHC is regarded as a baseline algorithm in this work and AprioriHC-D is an improved version of AprioriHC. Initially, a variable k is set to 1. Each item having a TWU no less than minimum utility is added to the set of 1-HTWUIs Ck. Then the algorithm proceeds recursively to generate itemsets having a length greater than k. During the k-th iteration, the set of k-HTWUIs Lk is used to generate (k+1)- candidates Ck+1 by using the Apriori-gen function [3]. It includes an effective strategies to reduce the number of PCHUIs generated in Phase I that areinspired by the IIDS algorithms [5]. An item is isolated if it appears in only few transactions. Isolated items play no role in CHUIs. Thus, when utilities of k-itemsets are being estimated, utilitiesof isolated items can be regarded as irrelevant, and therefore can be discarded. Therefore, isolated items of level k can be removed from each transaction . After removing the isolated items k itemsetis generated. From k itemset non closed items are removed.

UP tree is generated. In an UP-Tree, each node N consists of N.name, N.count, N.nu, N.parent, and a set of child nodes. N.name is the nodes item name. N.count is the nodes support count. N.nu is the nodes node utility. N.parent records the parent node of N. Each transactions are added into the UP tree. The support count and the node utilities are added with each item in the UP tree.The node sum of each item is calculated and items with smaller node sum are removed from the header table.In the header table, each entry records an item name and TWU value. Conditional pattern base is generated for every items with greater node sum. The common method for generating patterns in tree-based algorithms contains three steps: 1) Generate conditional pattern bases by tracing the paths in the original tree; 2) construct conditional trees (also called local trees) by the information in conditional pattern bases; and 3) mine patterns from the conditional trees. However, strategies DGU cannot be applied into conditional UP-Trees since actual utilities of items in different transactions are not maintained in a global UP-Tree.

The strategy DLU is applied. Then path utility is calculated. Paths with smaller path utilities are removed. Local tree is generated with reorganized path and the strategy DLN is applied by subtracting the transaction utility from each item in the path. PHUIs from UP tree is generated. And the Apriori algorithm is applied to the generated PHUIs and k itemset is generated. Non-closeditems are removed and refined result of closed + high utility itemsets is retrieved.

4. Future Scope

The integration of closed itemset mining and high utility itemset mining is only considered now.However, there are many other compact representations [18], [19], [20] that have not yet beencombined with high utility itemset mining. Is it possible to develop other compact representationsof high utility itemsets inspired by our work to reduce the number of redundant high utility patterns? High on shelf utility itemsets[22] is also considered in future. Although closed+ high utility itemset mining is essential to many research topics and industrial applications, it is still a novel and challenging problem. Other related research issues are worthwhile to be explored in the future.

5. Conclusion

Algorithms for mining closed high utility itemsets from transactional databases have been proposed. The concept of closed itemset is incorporated with high utility itemset mining to develop a representation named closed + high utility itemset.Two efficient algorithms named AprioriHC and AprioriHC-D are used. AprioriHC-D is an improved version of AprioriHC.And a data structure named UP tree is used for maintaining the information of high utility itemsets. PHUIs can be generated from UP tree with only few database scans. Several strategies are applied to decrease the overestimated utility and to enhance the performance of utility mining.Results show that the strategies adopted considerably improved performance by reducing both the search space and the number of candidates. This algorithm outperform the state of art algorithms substantially, especially when database contain lots of transactions, a low minimum utility threshold and support count is used.

References

[1] Cheng-Wei Wu, Philippe Fournier-Viger, Philip S. Yu, Fellow, IEEE, Vincent S. Tseng, Efficient Algorithms for Mining the Concise and Lossless Representation of

Closed+ High Utilit Itemsets", 2014.2345377, IEEE Transactions on Knowledge and Data Engineering.

[2] V. S. Tseng, C.-W. Wu, B.-E. Shie, and P. S. Yu, \UP-Growth: An E_cient Algorithm for High Utility Itemset Mining", in Proc. of the ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining, pp. 253262, 2010.

[3] 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.

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

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

[6] C.-W Wu, P. Fournier-Viger, P. S. Yu. and V. S. Tseng, \Efficient Mining of a Concise and Lossless Representation of High Utility Itemsets", in Proc. of the IEEE Int'l Conf. on DataMining, pp. 824-833, 2011.

[7] R. Chan, Q. Yang, and Y. Shen, \Mining High Utility Itemsets", in Proc. of the IEEE Int'l Conf. on Data Mining, pp. 19-26, 2003.

[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] Y. Liu, W. Liao, and A. Choudhary, \A Fast High Utility Itemsets Mining Algorithm", in Proc. of the Utility-Based Data Mining Workshop, pp. 90-99, 2005.

[10] Chowdhury Farhan Ahmed, Syed Khairuzzaman Tanbeer, Byeong-Soo Jeong, and Young- Koo Lee, \An Efficient Candidate Pruning Technique for High Utility Pattern Mining"in Proc. of the Utility-Based Data Mining Workshop, pp. 90-99, 2005.

[11] J. Wang, J. Han, and J. Pei, \Closet+: Searching for the Best Strategies for Mining Frequent Closed Itemsets", in Proc. of the ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining, pp. 236245, 2003.

[12] C. Lucchese, S. Orlando and R. Perego, \Fast and Memory E_cient Mining of Frequent Closed Itemsets", IEEE Transactions on Knowledge and Data Engineering, Vol. 18, Issue 1, pp. 21-36, 2006.

[13] Y.-C. Li, J.-S. Yeh, C.-C. Chang, Efficient algorithms for mining share-frequent itemsets", in: Proceedings 11th World Congress of Intl.Fuzzy Systems Association, Beijing, China, July 2005, pp. 534539.

[14] Y.-C. Li, J.-S. Yeh, C.-C. Chang, \Direct candidates generation: a novel algorithm for discovering complete share-frequent itemsets", in: L. Wang, Y. Jin (Eds.), 2nd Intl. Conf. on FuzzySystems and Knowledge Discovery (FSKD 2005), Lecture Notes in Arti_cial Intelligence, vol.3614, Springer-Verlag, Berlin, 2005, pp. 551560.

[15] C.-W. Lin, T.-P. Hong, W.-H. Lu, \An E_ective Tree Structure for Mining High Utility Itemsets,"Expert Systems with Applications, Vol. 38 Issue 6, pp. 7419-7424, 2011.

[16] K. Gouda and M. J. Zaki, \E_ciently Mining Maximal Frequent Itemsets", in Proc. of the IEEE Int'l Conf. on Data Mining, pp. 163-170, 2001.

[17] C.-W. Lin, T.-P. Hong, W.-H. Lu, \An Effective Tree Structure for Mining High Utility Itemsets", Expert Systems with Applications, Vol. 38 Issue 6, pp. 7419-7424, 2011.

[18] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal,\E_cient Mining of Association Rules using Closed Itemset lattice," Journal of Information Systems, Vol 24, Issue 1, pp. 2546, 1999.

[19] M. J. Zaki and C. J. Hsiao, \Efficient Algorithms for Mining Closed Itemsets and Their Lattice Structure", IEEE Transactions on Knowledge and Data Engineering, Vol. 17, Issue 4, pp. 462478, 2005.

[20] J. -F. Boulicaut, A. Bykowski, and C. Rigotti, \Free-sets: A Condensed Representation of Boolean Data for the Approximation of Frequency Queries", Data Mining and Knowledge Discovery, Vol. 7, Issue 1, pp. 522.

[21] T. Calders and B. Goethals, \Mining All Non-derivable Frequent Itemsets", in Proc. of the Intl Conf. on European Conference on Principles of Data Mining and Knowledge Discovery, pp. 74-85, 2002.

[22] Vincent S.Teng, Tzung-Pei Hong, Guo Cheng Lan, \A Three Scan Mining Algorithm for High On Shelf Utility Itemsets". Journal of Information Systems, Vol 24, Issue 1, pp. 2546, 1999.

[23] C.-W. Wu, B.-E. Shie, V. S. Tseng and P. S. Yu, \Mining Top-k High Utility Itemsets", in Proc. of the ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining, pp. 7886, 2012.

[24] B.-E. Shie, H.-F. Hsiao, V. S. Tseng and P. S. Yu, \Mining High Utility Mobile Sequential Patterns in Mobile Commerce Environments", in Proc. of the Intl. Conf. on Database Systems for Advanced Applications and Lecture Notes in Computer Science (LNCS), Vol. 6587/2011, pp. 224-238, 2011.

[25] H. Li, J. Li, L. Wong, M. Feng, Y. Tan, \Relative risk and odds ratio: a data mining perspective", in Proc. of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, pp. 368-377, 2005.

[26] T. Hamrouni, S. Yahia, E. M. Nguifo,\Sweeping the Disjunctive Search Space Towards Mining New Exact Concise Representations of Frequent Itemsets", Data Knowledge Engineering, Vol. 68, Issue 10, pp. 1091-1111, 2009

Author Profile

Nilovena received the B.Tech degrees in Information Technology Engineering from KMCT College of Engineering in 2012 and pursuing M.Tech degree in Computer Engineering from Calicut University. Her research interests include data mining and its application.

Anu received the Diploma in Computer science Engineering from Govt polytechnic Meppadi in 2004. AMIE in computer science in 2007 and M.Tech degree in Computer Engineering from NMAMIT Nitte Manglore in 2010. She is working as Assistant Professor, in KMCT College of Engineering.Her research interests include data mining and its application.