International Journal of Electrical and Computer Engineering (IJECE)
Vol. 7, No. 5, October 2017, pp. 2911~2918
ISSN: 2088-8708, DOI: 10.11591/ijece.v7i5.pp2911-2918

Optimized High-Utility Itemsets Mining for Effective Association Mining Paper

K. Rajendra Prasad
Department of Computer Science and Engineering, Institute of Aeronautical Engineering, Dundigal, Hyderabad, India

Article Info

Article history:

Received Jan 14, 2017
Revised Jun 7, 2017
Accepted Sep 11, 2017

Keywords:

Association rule mining
FHM
Frequent itemsets
High-utility itemsets
HUIM

ABSTRACT

Association rule mining is intently used for determining the frequent itemsets of transactional database; however, it is needed to consider the utility of itemsets in market behavioral applications. Apriori or FP-growth methods generate the association rules without utility factor of items. High-utility itemset mining (HUIM) is a well-known method that effectively determines the itemsets based on high-utility value and the resulting itemsets are known as high-utility itemsets. Fastest high-utility mining method (FHM) is an enhanced version of HUIM. FHM reduces the number of join operations during itemsets generation, so it is faster than HUIM. For large datasets, both methods are very expenisve. Proposed method addressed this issue by building pruning based utility co-occurrence structure (PEUCS) for elimatination of low-profit itemsets, thus, obviously it process only optimal number of high-utility itemsets, so it is called as optimal FHM (OFHM). Experimental results show that OFHM takes less computational runtime, therefore it is more efficient when compared to other existing methods for benchmarked large datasets.

Corresponding Author:

K Rajendra Prasad
Department of Computer Science and Engineering
Institute of Aeronautical Engineering
Dundigal, Hyderabad-500043, India
Email: krprgm@gmail.com

1. INTRODUCTION

Association rule mining methods [1] are used for discovering rules and items that are of frequent and user interested items. Existing association mining methods [2-3] use the support-confidence framework [4] in the discovery of user-interested rules. However, this framework is not sufficient for measuring the utility of item sets. In finding the utility of item sets [5], the traditional support-confidence framework is enhanced for measuring the semantic relations among the items which takes the semantic measure of the rule i. e the importance of the item is considered in the rule.

Frequent item set mining (FIM) [6] is one of the most important data mining task and it is popular in wide range of real life applications. The FIM discovers frequent itemsets using either Apriori or FP-growth [7] from a given transaction database, so frequently itemsets are appeared in results of transactions. Apriori and FP-growth methods generated the frequent itemsets without considering the profit of itemsets. It is emerging that; we can also consider the importance of frequent itemsets in terms of either a profit or utility. High Utility itemsets refers to a set of frequent items with high utility. High Utility itemsets mining (HUIM) [8] methods are playing a vital role in producing the set of high utility frequent item sets [9].

Association rule mining system is one of the popular methods for discovering of knowledge discovery about finding the relationships among the items. Aim of traditional association rule mining (or Apriori) is to discover the frequent itemsets, which defines the itemsets of each transaction in the transactional database. One of the limitation of this mining system is not concerned the other factors of itemsets such as profit, quantity, and cost etc., however, utility based mining system overcome these difficulties [10-11]. Thus, utility mining is performed based on the sold items quantity (this known as internal utility value) and profit of items (external utility). Support-confidence framework is used for measuring the strength of the rules but it may not sufficient in utility-based mining system. The proposed utility-based mining system uses the utility-confidence framework.

Utility mining system is an extension approach of FIM and it defines the resulting itemsets based on local transaction utility and external utility [12-13]. Key limitation of FIM is that it assumes that importance or profit of each item is same or simply ignored the profit (or utility of item). However, this assumption does not work in real life applications. This problem is addressed in HUIM method. HUIM method discovers the frequent itemsets with high utility. Another advanced method, FHM improves the utility mining process with respect to speed parameter. It is memory efficient, because FHM uses Estimated Utility Co-Occurrence Structure (EUCS) [14] for speed up the process of high-utility itemset mining. A pruning strategy [15] is used in FHM, which reduces the searching space, so that it is six times faster than HUIM [16]. Limitation of FHM is takes the more computational time of EUCS for large datasets. This paper explores the optimized EUCS and it is proposed for discovering of efficient and fastest high-utility itemsets mining method. Contributions of this proposed work are summarized as follows:

a. Determine the internal and external utility of each and every itemset in the transactional database
b. Mining the High Utility Frequent Itemsets
c. Determining the high utility itemsets based rules that are interest to stakeholders
d. Demonstrate and show the efficiency of proposed utility-based association system in the experimental study using real datasets

Remaining sections of the paper is organized as follows: Section 2 presents the Background study of the work, Section 3 describes the proposed methodology, Section 4 discusses the experimental study, and Section 5 presents the conclusion and future scope.

2. BACKGROUND STUDY

Several methods are proposed for HUIM, some of methods are already discussed in the paper, which are PB [17], Two-Phase [18], BAHUI [19], UP-growth [20], UP-growth+ [21]. Two-phase model is too efficient, because it suffers from extracting of huge amount of candidates and repeated scans of database. HUI-Miner [21] is proposed to the purpose of extracting high-utility itemsets using a single phase. Therefore, it had better approach for mining high-utility itemsets.

The HUIM is one of popular approach for discovering of high-utility itemsets by Transaction-Weighted-Downward closure model [22] and it uses two key phases. Two-phase, IHUP [23] and UP-growth algorithms are used two phases. In a first phase, these algorithms compute transaction weighted

(TWU) of generated candidate high-utility itemsets. In second phase, these algorithms find the utility of obtained candidates by scanning of database. HUIM algorithm filters the low-utility itemsets and discovers only high-utility itemsets by setting minimum threshold utility value. FHM [24] constructs the EUCS [25] as per sequence of illustrated steps of Figure 1. This figure shows the input data in terms of transaction database and utility table. Utility table describes the profit of each item of transactional database. For example, in transaction T1, number of items of I1 is 1 (it is defined in Figure 1a as I1:1), thus utility becomes number of items is multiplied by profit of item, as per the computation, utility=1 x 4=4. In Figure 1c, Transaction utility (TU) is computed by Equation (1).

$$\mathrm{TU(Ti)} = \sum_{x \in \mathrm{Ti}} \mathrm{frequency}(x) \times \mathrm{profit}(x) \qquad (1)$$

Transaction weighted utility is computed by Equation (2) in Figure 1d, EUCS by Equation (3) in Figure 1e.

$$\mathrm{TWU(item)} = \sum_{\mathrm{item} \in \mathrm{Ti}} \mathrm{TU(Ti)} \qquad (2)$$

$$\mathrm{EUCS}(a,b) = \sum_{\{a,b\} \subseteq \mathrm{Ti}} \mathrm{TU(Ti)} \qquad (3)$$

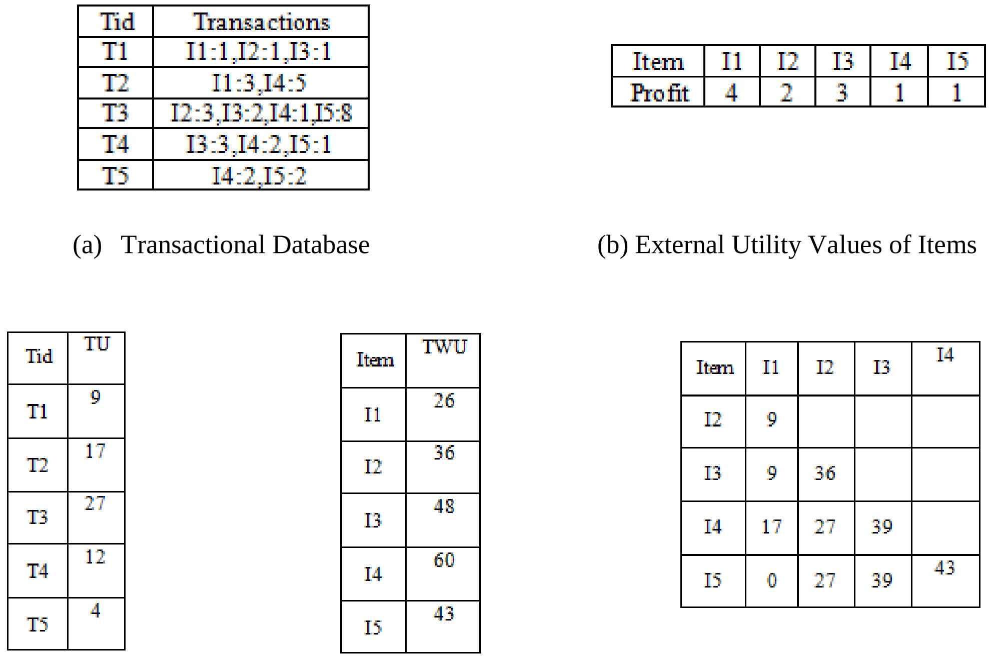
The algorithm of FHM [24] is as follows

Algorithm 1: FHM

Input: D, a transactional database,
Minutil, a user specified threshold

Output : set of high-utility itemsets

1. Scan database 'D' and compute the TWU of single items using Equation (2)
2. Find high-utility itemsets using condition TWU(i) > Minutil, here 'i' refers to high-utility item
3. Build the EUCS structure using Equation (3)
4. Derive high-utility itemsets using Search procedure [24].



Tid	Transactions
T1	I1:1,I2:1,I3:1
T2	I1:3,I4:5
T3	I2:3,I3:2,I4:1,I5:8
T4	I3:3,I4:2,I5:1
T5	I4:2,I5:2

Item	I1	I2	I3	I4	I5
Profit	4	2	3	1	1

(a) Transactional Database (b) External Utility Values of Items

Tid	TU
T1	9
T2	17
T3	27
T4	12
T5	4

Item	TWU
I1	26
I2	36
I3	48
I4	60
I5	43

Item	I1	I2	I3	I4
I2	9			
I3	9	36		
I4	17	27	39	
I5	0	27	39	43

(c)Transaction Utility (d) Transaction Weighted Utility (e) EUCS

Figure 1. Illustration of EUCS process for transactional database

Search procedure in step 4 of algorithm 1 starts from single items and it describes recursively the search space of itemsets by appending single items and it prunes the space based on following property 1. In the implementation of present FHM system, building EUCS is a speedy process, because it is noted that few items that co-occur in the EUCS, so it uses less space in a memory. Limited number of pairs is co-occurred from a transactional database. Another key importance of FHM is that it builds the EUCS after deleting low-utility itemsets in a search procedure of step 4 in algorithm 1.

Property 1 [24] ((sum of iutils and rutils). Let X is an itemset. Let the extensions of X be the itemsets that can be obtained by appending an item y to X such that $y \succ i$ for all item i in X. If the sum of iutil and rutil values in the utility-list of x is less than minutil, all extensions of X and their transitive extensions are low-utility itemsets.

More optimized EUCS construction steps are required for improving of FHM and proposed OFHM (Optimized EUCS based FHM) is presented in following sub-section.

3. PROPOSED WORK

The key novelty is to establish a mechanism, which is called as pruning based EUCS (PEUCS). It eliminates an expensive join operation in FHM and allows to elimination of low-utility extension without

defining of utility list. Suppose, no tuple satisfying the minimum utility and then ignore the current utility item and its supersets and these itemsets need not be explored. This proposed method optimizes the process of FHM by ignoring or pruning of low-utility itemsets.
Algorithm 2: OFHM

Input : D-transactional database, minutil- minimum utility threshold utility
Output : generating high-utility itemsets

a. Scan the transactional database with utility values'D' for finding transaction utility 'TU' of transactions and Transaction Weighted Utilization (TWU) of items
b. Find high utility 1- itemsets, which satify the condition of TWU(item) ≥ minutil
c. Build the EUCS structure for co-items using TU
d. Prune the low-utility itemsets from the EUCS, that is, pair (x, y) value in EUCS is lessthan minutil, then 2-itemset (x, y) is low-utility itemset and this 2-itemset and its supersets values are eliminated. So, that number of resulting high-utility itemsets is optimized.
e. Step 4 is applied recursively for the extensions of y when (x, y) is high-utility itemset for generating of high-utility itemsets.

The OFHM procedure takes, as input is 'D' (transactional database) and 'minutil' (threshold for minimum utility). In Step 1, it scans the entire transactional database for finding transactional utility (TU) of all transactions and TWU of individual items. Step 2 determines the high utility 1-itemsets, which results are satisfied with value of 'minutil'; Utility of co-items with size of 2 i.e 2-itemsets are formed in EUCS structure with utility in Step 3. Step 4 performs the optimizations of EUCS; i.e low-utility 2-itemsets are eliminated or pruned in further steps. Thus, size of data of high-utility itemsets is optimized and automatically Step 5 performs speedup the process of determining of next level high-utility itemsets (i.e 3-itemsets, 4-itemsets….).

High utility itemset considers the profit and quantity of itemsets of transactional database. The problem of high utility mining is attempted in the proposed method. This proposed algorithm can efficiently generate the high-utility itemsets and it effectively applied the pruning steps for reduction of unpromising items by setting minimum threshold value. Discovering the promising high-utility itemsets is addressed by proposed OFHM method. It optimizes the size of promised high utility itemsets. The proposed algorithm needs only a single phase instead of two-phases. It computes the TU of transactions and TWU of individual items in the same phase. In real life applications, actual products have the information in terms of transaction with frequency and profit (or utility). In peculiar supermarket applications, few of products may occur with very low frequency, however, it may with high-profit. In such practical cases, this proposed method effectively works for performing high-utility mining. Traditional methods may not consider the utility of itemsets during mining process and some other methods are considered the utility value, and they required multiple scans of database. It is too expensive in large data computation of utility mining.

High utility itemset mining is a much more difficult problem than frequent itemset mining. Therefore, algorithms for high-utility itemset mining are generally slower than frequent itemset mining algorithms. The proposed methodology is faster than traditional HUIM algorithm for discovering frequent itemsets. This proposed framework is useful in real world applications such as e-commerce business retails applications, web recommended systems. In which, either profit or number of times a user visited page is considered as utility. It then performs utility mining based on this utility value. One of tradition method, Apriori is used for pruning the candidate item search space, but that cannot applicable for high-utility itemsets, so, OFHM is proposed for addressing this problem effectively when compared with other utility mining methods. It optimizes the searching space, which enhances the performance of utility mining task. It uses the some kind of data structure, named as utility-list; this is defined for high utility itemset. Suppose the utility of item is not satisfied with minimum utility threshold value, and the subsequent utility list may be undefined and the corresponding list is ignored, hence it retrieves the utility-list of high-utility itemsets and ignores the utility-list of other itemsets. It is the best improvement step in OFHM than othet utility mining methods. It uses the strategy of pruning to the purpose of ignoring several join operations in order to eliminate a low-utility extension of several unsatisfied items using EUCS. The EUCS can be denoted as a hash map table and the relevant example is illustrated in the previous section for analyzing the problem of utility mining. This hash map is used in OFHM for achieving of memory efficiency, since EUCS is sparse in nature. Construction of utility-list from OFHM is taken very less time. Therefore, space and time requirements are optimized in OFHM. Extensive study of experimental results of real world datasets is discussed in the following section.

4. EXPERIMENTAL STUDY

Substantial experiments were conducted on real life datasets; these are chainstore, foodmart, mashroom utility datasets. The datasets are collected from FIMI repository [26].Table 1 shows the characterstics of the datasets. The implemented OFHM method adopts the pruning strategy for ignoring of unwanted low-utility itemsets; it act as improved strategy, whcih reduce the rescan of low-utility itemsets for large datasets. It is significable improvement. Performance of exisiting utility mining methods, such as HUIM [23], FHM [24], GHUI-Miner [27] are compared with OFHM in this experimental study. Optimization of high-utility items is performed in OFHM. Existimg and proposed methods are executed in Java on Eclipse neo platform with JDK 1.6. These experiments are conducted on Windows 7 platform with hardware of core i3 1.7Ghz processor with 32GB RAM.

Table 1. Characterstics of the Datasets

Dataset	Number of Transactions	Average Length	Number of Items	Type
Chainstore Utility	340,183	33.8	468	Dense
Chess	3,196	37.0	75	Dense
Foodmart	227	17.88	1559	Sparse
Mushroom	8,124	23.0	119	Dense

Each transaction is composed with collection of several items, for example, chainstore utility dataset have used 468 items for transactions. Each transaction consists of combination of items among 468 items. During utility itemset generation, it is derived that very large number of itemsets are occurred in a fashion of utility based 1-itemsets, 2-itemsets,….. so on. Existing HUIM, GUI Miner, FHM are experimented on large datasets of Table 1 with different threshold utility values or minimum utility values for generation of high-utility frequent itemsets based association rules. It is noted that FHM is taken less computational time compared to HUIM, and GUI Miner methods. Proposed OFHM prunes the low-utility itemsets by construnting PEUCS, therefore, OFHM obtains optimal number of utility itemsets for generation of high utility basesd association rules. Computational rumtime is compared for HUIM, GUI Miner, FHM, and proposed OFHM and it is demonstrated in following figures (Figure 2 to Figure 5). It is noted that proposed OFHM is faster and more fit for generation of high-utility based association rules.

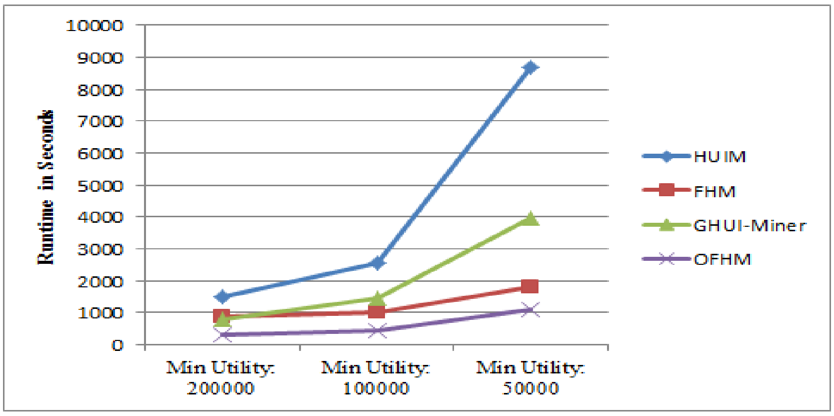


Figure 2. Computational runtime comparison for chainstore utility dataset

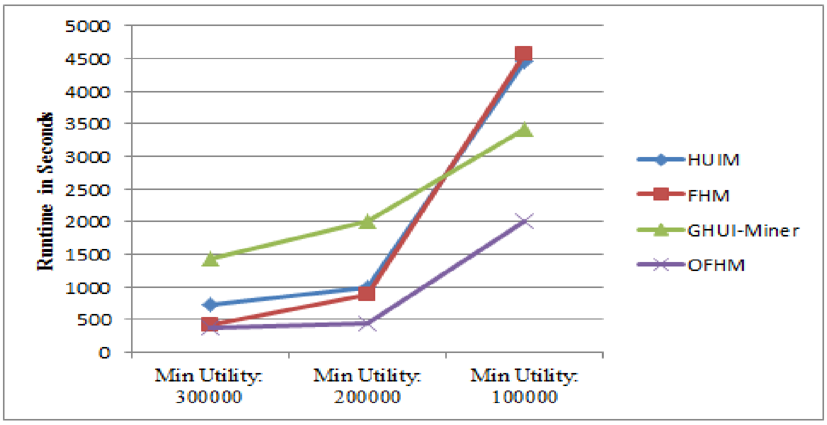


Figure 3. Computational runtime comparison for chess utility dataset

The performance of methods are compared based on the computation run time. Figure 2 indicates that our proposed method OFHM is perfomed as better and achives faster association mining results when tested with chainstore utility dataset.

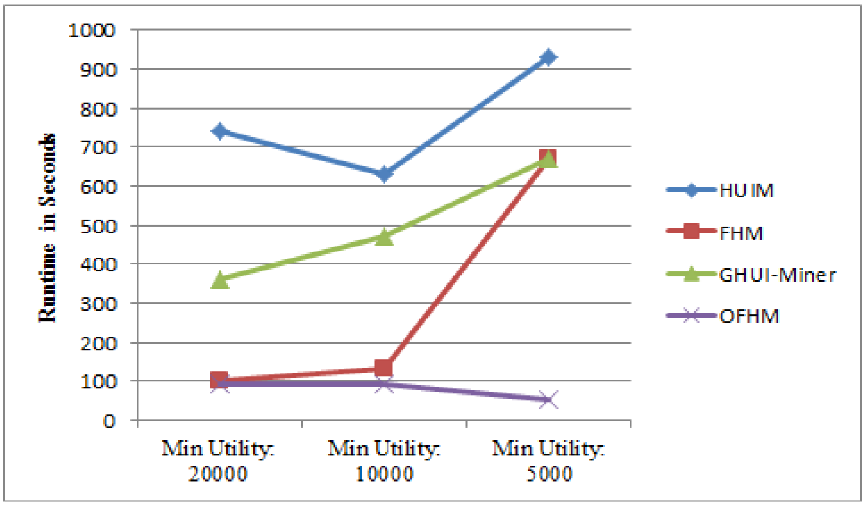


Figure 4. Computational runtime comparison for foodmart utility dataset

Choosing of minimum utility depends on size and charactestics of datasets. Minimum utility value is varied for every dataset. In Figure 4, the methods are compared at minimum utility values of {20000, 10000, and 5000} at foodmart dataset, while in chess dataset we have taken minimum utility values are {300000, 200000, 100000} and it is shown in Figure 3. In Figure 5, computational times of existing and proposed methods are compared with minimum utility values of {550000, 500000, 400000}.

The GHUI miner also makes the faster computation of utility mining results than FHM and HUIM, however, our proposed OFHM is prunes the unnecessary low-utility itemsets, thus, this method is recommended as best for high-utility itemset mining. It is proved that is uses a very less memory than other methods. There it is also memory efficient.

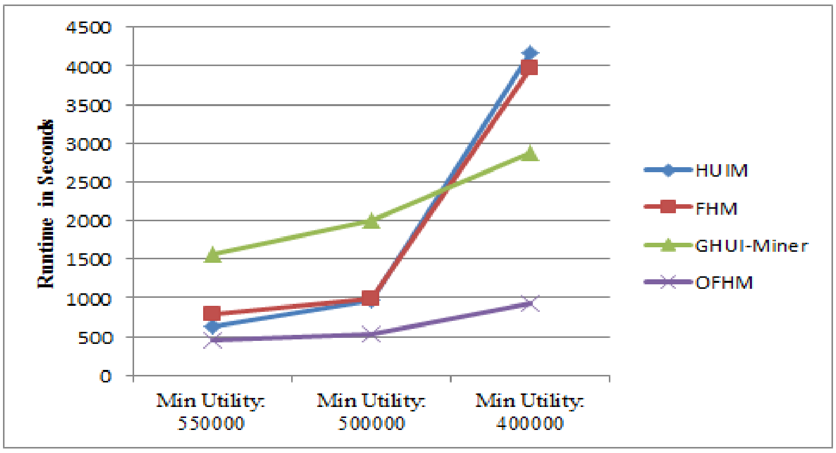


Figure. 5. Performance comparison for mushroom utility dataset

In order for evaluating, if mining results is useful to the retail store, we compare the high utility itemsets with the frequent itemsets mined by traditional association rule mining methods. For example, a kind of bagged fresh vegetable is a frequent item (the support is over 4%), however, its contribution the total profit is less than 0.35%. A combination of two kinds of canned vegetable is also a good example, which occurs in more than 1% of the transactions, but contributes less than 0.25% of the overall profit. Therefore, utility mining can help the marketing professionals in this retail store makes better decision, such as highlight their highly profitable items or itemsets and reduce the inventory cost for frequent but less profitable items or itemsets.

It is observerd that threshold value i.e. minimum utility value is different for each dataset, since the frequency and utility of itemsets are varied for datasets. Therefore, in the experimental, differenet threshold values are given for different datasets for measuring the computation time of utility mining methods. From the investigation of experimental results of utility mining methods, it is noted that the OFHM derives the high-utility itemsets in a faster way than other methods.

5. CONCLUSION AND FUTURE SCOPE

This paper is majorly focused on association mining methods for effective generation of high-utility itemsets. Traditional association methods discover frequent itemsets without considering either a profit or utility of itemsets. Recent advances presented utility based mining methods and these methods generate both low and high-utility itemsets. It is required that prune the low-utility itemsets for faster mining results. The proposed OFHM addressed the pruning problem for elimination of low-utility itemsets; therefore, it generates high-utility itemsets effectively in a shorter time when compared to other methods. Future scope of the proposed work is that to extend the propose work for big datasets and enhance the utility mining methods for performing of distributed processing for better data analytics about high-utility itemsets.

REFERENCES

[1] Sushil Kumar Verma, R.S. Thakur, "Fuzzy Association Rule Mining Based Model to Predict Students' Performance," International Journal of Electrical and Computer Engineering, Vol. 7, No.4, 2017
[2] Madhu G, Nagachandrika G, "A New Paradigm for Development of Data Imputation Approach for Missing Value Estimation," International Journal of Electrical and Computer Engineering, pp: 3222- 3228, Vol. 6, No.6, 2016
[3] Goethals, "Survey on Frequent Pattern Mining," manuscript, 2003
[4] P. Fournier-Viger, C. Wu, V. S. Tseng, —Mining Top-K Association Rules,‖ in Proc. of Int'l Conf. on Canadian conference on Advances in Artificial Intelligence, pp. 61–73, 2012
[5] H. Ryang, U Yun and K. Ryu, —Discovering High Utility Itemsets with Multiple Minimum Supports,‖ Intelligent Data Analysis, Vol. 18(6), pp. 1027-1047, 2014
[6] J. Han, J. Pei, and Y. Yin, "Mining Frequent Patterns without Candidate Generation," Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 1-12, May 2000
[7] Savasere, E. Omiecinski, and S. B. Navathe, "An Efficient Algorithm for Mining Association Rules in Large Databases," in Proc. 21st Int. Conf. Very Large Databases, 1995, pp. 432–444.
[8] M. Liu and J. Qu, —Mining High Utility Itemsets without Candidate Generation,‖ in Proc. of ACM Int'l Conf. on Information and Knowledge Management, pp. 55-64, 2012

[9] Lin, T. Hong, G. Lan, J. Wong and W. Lin, ―Efficient Updating of Discovered High-utility Itemsets for Transaction Deletion in Dynamic Databases,‖ Advanced Engineering Informatics, Vol. 29(1), pp. 16-27, 2015.

[10] G. Pyun and U. Yun, ―Mining Top-K Frequent Patterns with Combination Reducing Techniques, ―Applied Intelligence, Vol. 41(1), pp. 76-98, 2014.

[11] P. Tzvetkov, X. Yan and J. Han, ―TSP: Mining Top-K Closed Sequential Patterns,‖ Knowledge and Information System, Vol. 7(4), pp. 438-457, 2005.

[12] Wu, B. 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. 78–86, 2012.

[13] J. Yin, Z. Zheng, L. Cao, Y. Song and W. Wei, ―Mining Top-K High Utility Sequen-tial Patterns,‖ *in Proc. of IEEE Int'l Conf. on Data Mining*, pp. 1259-1264, 2013.

[14] H. Yao and H. J. Hamilton, “Mining Itemset Utilities from Transaction Databases,” Data Knowl. Eng., vol. 59, no. 3, pp. 603–626, 2006

[15] H. Yao, H. J. Hamilton, and C. J. Butz, “A Foundational Approach to Mining Itemset Utilities from Databases,” in Proc. SIAM Int. Conf. Data Mining, 2004, pp. 482–486.

[16] H. Yao, H.J. Hamilton,” Mining Itemset Utilities from Transaction Databases,” Data Knowl. Eng. 59 (2006) 603–626

[17] Lan, G. C., et al. “ An Efficient Projection Based Indexing Approach for Mining High Utility Itemsets,” Knowledge and Information System, Vol.38, Issue.1, pp. 85-107, 2014

[18] Ying liu et al. “ A Two-phase Algorithm for Fast Discovery of High-utility Itemsets,” LNAI, PAKDD 2005, Springer, pp. 689-695, 2005

[19] W. song et al., “BAHUI: Fast and Memory Efficient Mining of High Utility Itemsets Based on Bitmap.” Int. journal of data warehousing and mining. Vol. 10, Issue.1, pp.1-15, 2014

[20] S. Dawar and V. Goyal, “UP-Hist tree: An Efficient Data Structure for Mining High Utility Patterns from Transaction Databases,” in Proc. 19th Int. Database Eng. Appl. Symp., 2015, pp. 56–61.

[21] M.Venkatesh and M. Krishnamurthi, “Mining Association Rules for High-utility Itemsets Using up-growth+ Algorithm from Transactional Databases,” Int. Journal of Computer Engineering and Technology, Vol.5 Issue.3, pp.164-173

[22] U. Yun, H. Ryang, and K. H. Ryu, “High Utility Itemset Mining with Techniques for Reducing Overestimated Utilities and Pruning Candidates,” Expert Syst. Appl., vol. 41, no. 8, pp. 3861–3878, 2014

[23] Jerry Chun-Wei Lin et al. “An Incremental High-Utility Mining Algorithm with Transaction Insertion.” ScientificWorld Journal, Vol 2015, pp.1-15 2015.

[24] Y. Liu, W. Liao, and A. Choudhary, “A Fast High Utility Itemsets Mining Algorithm,” in Proc. Utility-Based Data Mining Workshop SIGKDD, 2005, pp. 253–262.

[25] Chun-wei Lin et al., “ Efficient Updating of Discovered High-utility Itemsets for Transaction Deletion in Dynamic Databases,” Advanced Engineering Informatics, Vol.29, pp.16-27, 2015

[26] http://fimi.ua.ac.be/

[27] Phillippe Fournier-Viger et al., “ FHM: Faster High-utility Itemset Mining Using Estimated Utility Co-occurrence Pruning,” LNAI

BIOGRAPHY OF AUTHOR

Dr. K. Rajendra Prasad Graduated in B.Tech (CSE) from Jawaharlal Nehru Technological University, Hyderabad in 1999. He received Masters Degree in M.Tech (CSE) from Visvesvaraya Technological University, Belgaum, in 2004. He received Ph.D in Computer Science & Engineering from JNTUA, Ananthapur, in 2015. Presently, he is working as Professor and Head of CSE Dept., Institute of Aeronautical Engineering, Hyderabad. He has more than 30 Publications in various International Journals and Conferences. He is a life member of CSI, and member of IEEE. His research interests are data mining &data warehousing, and databases.