International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 04 Issue: 07 | July -2017 www.irjet.net p-ISSN: 2395-0072

An Efficient and Scalable UP-Growth Algorithm with Optimized Threshold (min_util) for Mining High Utility Item sets from Transactional Database.

Mr. Sunil H. Sangale[1], Prof Dr. D.V.Patil[2], Prof. R.C. Samant[3]

1 PG Student, Dept. of Computer Engg. R.H. Sapat College, Pune University, Nashik , Maharashtra, India
2 Head Of Dept. of Computer Engg. R.H. Sapat College, Pune University, Nashik , Maharashtra, India
3 Asst. Professor, Dept. of Computer Engg. R.H. Sapat College, Pune University, Nashik , Maharashtra, India

---

Abstract - *High utility itemsets mining from a big transactional database is an emerging concept in data mining which refers to the discovery of knowledge like high utility itemsets (profits) with user-specified minimum utility threshold min_util. Since a number of relevant algorithms have been proposed in past years, they fall into the problem of producing a large number of candidate itemsets for high utility itemsets. Though, setting min_util properly is a difficult problem for users. Generally discourse, finding a suitable minimum utility threshold by trial and error is a tedious process for users. If min_util is set very small value, then very large set of High Utility Itemsets will be generated, which may cause the mining process to be very inefficient. On the further case, if min_util is set very large, it is expected that no High Utility Itemsets will be found. Such a huge number of candidate itemsets decrease the mining performance in terms of time and space complexity. In this paper, we discourse the above issues by proposing a new framework for high utility itemset mining, with desired number of HUIs to be mined. Here we have done a structural comparison of the two algorithms with discussions on their advantages and limitations. Experiential evaluations on both real and synthetic datasets show that the performance of the proposed algorithms is close to that of the optimal case of state-of-the-art utility mining algorithms. This template, modified in MS Word 2007 and saved as a "Word 97-2003 Document (Size 10 & Italic , cambria font)*

Key Words: Candidate pruning, frequent itemset, high utility itemset, utility mining, data mining.

1. INTRODUCTION

Frequent item set mining (FIM) is a fundamental research concept in data mining. The traditional FIM may yield a large numbers of frequent but low-value item sets and may lose the information on valuable item sets having low selling frequencies. Hence, it cannot satisfy the requirement of users who desire to discover item sets with high profits. Even, the association rule mining algorithm named apriori is used to find the candidate itemsets and then derive the frequent itemsets based on the minimum support value. The apriori used join and prune mechanism to find the itemsets. To address the issues of frequent mining, utility mining came into existence. In utility mining, each item is associated with a unit profit and the quantity of that item. An item set is called high utility item set (HUI) if its utility is no less than a user specified minimum utility threshold min_util. Efficient mining the high utility itemsets in databases is not an easy task because the downward closure property used in FIM does not hold for the utility of item sets. In other words, pruning search space for HUI mining is difficult because a superset of a low utility item set can be high utility. To tackle this problem, the concept of transaction weighted utilization (TWU) model was introduced. In this model, an item set is called high transaction-weighted utilization item set (HTWUI) if its TWU is no less than min_util, where the TWU of an item set represents an upper bound on its utility.

Depending on the threshold value, the search space can be very small or very large. Besides, the choice of the threshold greatly influences the performance of the algorithms. If the threshold is set too low, many high utility itemsets are generated and it is difficult for the users to comprehend the results. A huge search space makes mining algorithms incompetent or even run out of memory, because the more HUIs the algorithms generate, the more resources they consume. On the contrary, if the threshold is set too high, no HUI will be found. To find a proper value for the min_util threshold, users need to try different thresholds by estimating and re-executing the algorithms over and over until being satisfied. In this paper, we discourse all of the above challenges by proposing a novel framework for high utility item set mining, with the desired number of HUIs to be mined. This technique is proposed for mining the complete set of top HUIs in databases without the need to specify the min_util threshold. This strategy is concerned with any kind of one phase algorithm which have item set with their utility.

2. LITERATURE SURVEY

R. Agrawal et al in [2] has proposed Apriori algorithm, it is used to find frequent itemsets from the database. In miming the association rules we have the problem to generate all association rules that have support and confidence greater than the user specified minimum threshold respectively.

The first pass of the algorithm simply sums item existences to decide the large 1-itemsets.

J. Han et al in [6] proposed frequent pattern tree (FP-tree) structure, an extended prefix tree structure for storing central 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.

W. Wang et al in [7] proposed weighted association rule (WAR). In WAR(weighted association rule), they discover first frequent itemsets and the weighted association rules for each frequent itemset are generated. In WAR, they have used a twofold approach. First it generates frequent itemsets; here they ignore the weight associated with each item in the transaction. In second for each frequent itemset the WAR finds that meet the support, confidence.

Liu et al in [8] proposes a Two-phase algorithm for finding high utility itemsets. The utility mining is to identify high utility itemsets that initiative a large lot of the total utility. Utility mining is to find all the itemsets whose utility values are beyond a user specified threshold. Two-Phase algorithm, it efficiently trims down the number of candidates and finds the complete set of high utility itemsets. Filter the overestimated itemsets. Two-phase requires fewer database scans, less memory space and less computational cost.

Li et al in [9] suggested two efficient one pass algorithms MHUI-BIT and MHUI-TID for mining high utility itemsets from data streams within a transaction sensitive sliding window. To improve the efficiency of mining high utility itemsets two effective representations of an extended lexicographical tree-based summary data structure and itemset information were developed.

V.S. Tseng et al in [13] proposes a novel method THUI (Temporal High Utility Itemsets)-Mine for mining temporal high utility itemset mining. The temporal high utility itemsets are effectively identified by the novel contribution of THUI-Mine by generating fewer temporal high transaction weighted utilization 2-itemsets such that the time of the execution will be compact significantly in mining all high utility itemsets in data streams.

J. Hu et al in [12] defines an algorithm for frequent item set mining, that identify high utility item combinations. The objective of the algorithm is different from the frequent item mining procedures and old association rule.

Erwin et al in [10] observed that the conventional candidate-generate-and-test approach for identifying high utility itemsets is not suitable for dense date sets. The high utility itemsets are finds using the pattern growth approach is the innovative algorithm called CTU-Mine.

Shankar [11] proposed a novel algorithm Fast Utility Mining (FUM) which finds all high utility itemsets within the given utility constraint threshold.

Cheng-Wei Wu et al in [15] suggested an innovative algorithm with a compressed data structure for efficiently discovering high utility itemsets from transactional databases. Depending on the construction of a global UP-tree the high utility itemsets are generated using UP-Growth which is one of the efficient algorithms. In phase-I three steps are followed by framework of UP-tree as: (i). UP-Tree construction, (ii). Generation of PHUIs from the UP-Tree and (iii). The high utility itemsets should be identified using PHUI.

3. PROBLEM DEFINITION

In the literature survey we have considered the different proposed methods for high utility mining from large datasets. But the frequency of item set is not sufficient to reflect the actual utility of an item set. The Proposed system will required dataset and minimum utility threshold which is enter by administrator as an input for finding the High utility itemsets. But most of time administrator is confuse to enter threshold value. When the threshold value is too less then system will generate too large set of HUI and when the threshold value is too large, it will generate too less or expected no item sets will found. So, to solve this problem the optimized minimum utility value or threshold value will generate by system.

To analyze utility of items or itemsets will be computed by using UP-Tree as data structure and optimized threshold value with the efficient algorithm UP-Growth+.

4. PROPOSED SYSTEM

The proposed high utility mining system will be a conceptual model built for large transactional database. Mining high utility itemsets from transactional databases refers to finding the itemsets with high profit. The high utility itemset means that if its utility is no less than a user-stated minimum utility threshold, else it is called a low-utility itemset. Administrator can enter the Specified Minimum Utility Threshold value. When administrator wants to find the utility of items with its profit the administrator gives the minimum threshold and dataset to the system. The system will compute the operation on given input and generate the high utility itemsets.

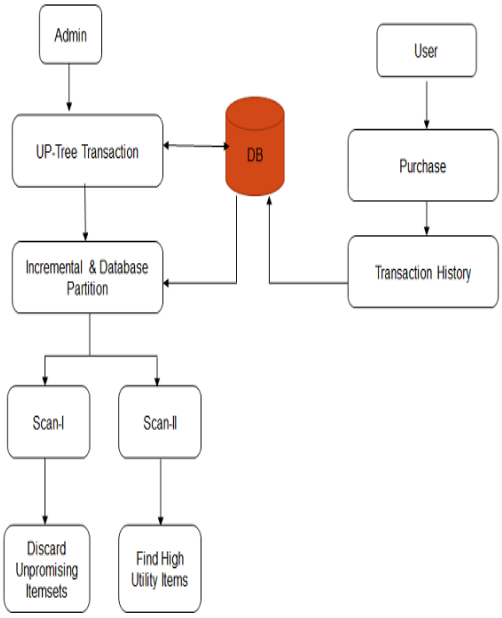

Fig. 1. System Architecture

5. IMPLEMENTATION DETAILS

A. Mathematical model

S : I,D,X,M,N,K
Where;
I = (i 1; i2; :::im): items,
X = (i1; i2; :::ik): itemset
D = (T 1; T 2; :::; T n) : Transaction Database .
k = number of distinct items.
M=number of finite set of items
N =number of Transaction
F =F1, F2, F3, F4, F5, F6, F7

Function F1:
Utility of an item ip in a transaction Td is denoted as u(ip , Td).
u(ip , Td)=pr(ip)*q(ip , Td)
Where,
pr(ip)=unit profit,
q=quantity of item in transactions

Function F2:
Utility of an itemsets X in Td is denoted as u(X, Td)
u(X, Td)= ∑ip€X∧X⊆Td u(ip,Td)

Function F3:
Utility of an itemset X in Td is denoted as u(X, Td)
u X, Td = ∑X⊆Td∧Td∈DX, u(X,Td)

Function F4:
High Utility itemsets. Itemset is called High utility itemset if its utility is no less than a user specified minimum utility threshold which is denoted as min_util. Otherwise, it's called low utility itemset.

Function F5:
Transaction utility of a transaction Td denoted as TU(Td).
TU(Td)=u(Td, Td)

Function F6:
Transaction-Weighted utility of an itemset X is sum of the transaction containing X, which is denoted as TWU(X)
TWU(X) = TU (X⊆Td∧Td∈D,Td)

Function F7:
An itemset X is called a high-transaction weighted utility itemset (HTWUI) if TWU(X) is no less than min_util.

B. Algorithm

Input:
- UP-tree *T*X.
- A header table Hx.
- An itemset X.
- Minimum utility threshold min_util,

Output:
- All PHUIs in *T*X

Steps:
1) For each entry ik in Hx do
2)Trace each node related to ik via ik h link and accumulate ik. nu to nusum(ik);
3) If nusum(ik)≥ min_util, do
4) Generate a PHUI Y = X ∪ik;
5) Set pu(ik) as estimated utility Y;
6) Construct Y-CPB;
7) Put local promising item in Y-CPB into HY
8) Apply DPU to reduce path utilities of the paths;
9) Apply Insert_ Reorganized_ Path mnu to insert into *T*Y with DPN;
10) If *T*Y ≠ null then call enhanced UP-Growth+ ()
11) End if
12) End for

6. Results & Discussion

This section describes the experimental environment and the performance of the proposed algorithm with different parameters compared to the IHUP algorithm. This algorithm is implemented in java language. The software tool used is Eclipse IDE 8.0.

6.1 Experimental Environment

In order to show the performance of proposed Efficient UP-Growth+ algorithm, we compare with the PHUI algorithm. Here we assume a simple transaction database with few items and transactions.

6.2 Results

Figure.2 shows the Execution Time Result of utility mining for minimum utility and specified number of items using Up-growth and Optimized Threshold Up-growth algorithms.

After analyzing below both charts, the Optimized Threshold Up-growth Algorithm is more Efficient and scalable in terms of Execution Time while finding high utility Items from transactions.

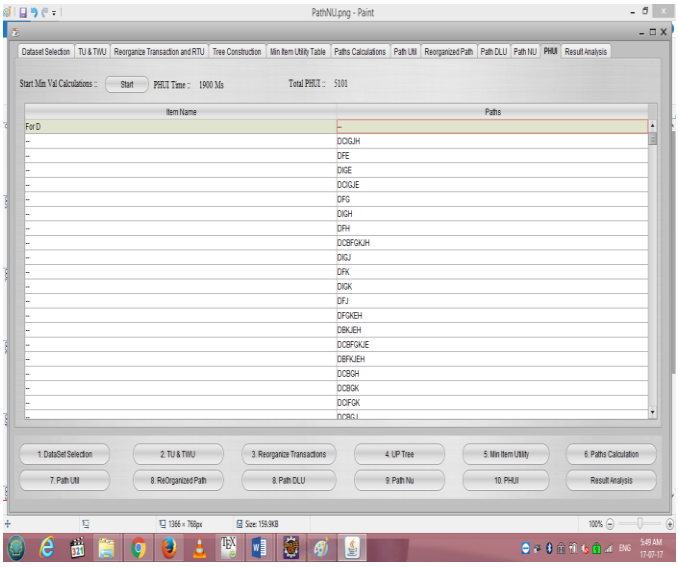

Fig. 2. Snapshot of the PHUI

Fig. 3. Phase I. Comparison of No. of PHUI

Fig. 4. Phase II. Time Analysis

Here Fig 3. Phase I shows the comparison graph between UP-Growth with Optimized UP-Growth algorithm. No of PHUI generated by optimized algorithm is less and optimum. And second Fig.4 shows time analysis comparison with UP-Growth with Optimized UP-Growth algorithm. Optimized UP-growth algorithm takes less time to generate PHUI.

CONCLUSION

The main problems with the existing methods are the generation a huge set of itemsets and scanning of the original database several times. Hence, the proposed algorithm ensures that it generates efficient itemsets with only two scans. From the above experimental results, we can conclude that the proposed algorithm can efficiently find the high utility itemsets. Since the algorithm generates optimum candidate items or itemsets, it takes less time to find the high utility itemsets. Thus, pruning the itemsets very well at early stages saves the time as well as space.

ACKNOWLEDGEMENT

I am thankful to Prof. **Dr. Prof D.V.Patil** and **Mrs. R. C. Samant** for their kind support and valuable guidance. They help me time to time to improve the quality of project work in all aspects. I thank to my colleagues and friends who guide me directly and indirectly to complete the paper work. I would also thankful to all the staff members of Computer Engineering , who gives their valuable guidance to me. Specially, I am thankful to my family members for their support and co-operation during this Project work.

REFERENCES

[1] R. Agrawal, T. Imielinski and A. Swami, 1993,"Mining association rules between sets of items in large databases", in Proceedings of the ACMSIGMOD

International Conference on Management of data, pp 207-216.

[2] R. Agrawal and R. Srikant, 1994, "Fast Algorithms for Mining Association Rules", in Proceedings of the 20th International Conference Very Large Databases, pp. 487-499.

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

[4] M. Add, L. Wu, Y. Fang, "Rare Item set Mining", Sixth International conference on Machine Learning and Applications, 2007, pp 73-80.

[5] R. Chan, Q. Yang, Y. D. Shen, "Mining High utility Item sets", In Proc. of the 3rd IEEE Intel.Conf. On Data Mining (ICDM), 2003.

[6] J. Han, J. Pei, 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.

[7] W. Wang, J. Yang and P. Yu, "Efficient mining of weighted association rules (WAR)," in Proc. of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2000), pp. 270-274, 2000.

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

[9] H. F. Li, H. Y. Huang, Y. C. Chen, Y. J. Liu and 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.

[10] A. Erwin, R. P. Gopalan and N. R. Achuthan, "Efficient mining of high utility itemsets from large datasets," in Proc. of PAKDD 2008, LNAI 5012, pp. 554-561.

[11] S.Shankar, T.P.Purusothoman, S. Jayanthi,N.Babu, A fast agorithm for mining high utility itemsets , in :Proceedings of IEEE International Advance Computing Conference (IACC 2009), Patiala, India, pp.1459-1464.

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

[13] V. S. Tseng, C. J. Chu and T. Liang, "Efficient Mining of Temporal High Utility Itemsets from Data streams," in Proc. of ACM KDD Workshop on Utility-Based Data Mining Workshop (UBDM'06), USA, Aug., 2006.

[14] V. S. Tseng, C.-W. Wu, B.-E. Shie and P. S. Yu, "UP-Growth: An Efficient Algorithm for High Utility Itemsets Mining," in Proc. of the 16th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining (KDD 2010), pp. 253-262, 2010.

[15] Vincent S. Tseng, Bai-En Shie, Cheng-Wei Wu, and Philip S. Yu, Fellow, IEEE, "Efficient Algorithms for Mining High Utility Itemsets from Transactional Databases", IEEE Transactions on Knowledge and Data Engg., VOL. 25, NO.8, AUGUST 2013.

[16] Frequent Itemset Mining Implementations Repository, http://fimi.cs.helsinki.fi/, 2012.