International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

Mining Method for Long Pattern from Database

Madhu Nashipudimath[1], Monali Deshmukh[2]

[1]Assistant Professor PIIT, New Panvel, Navi Mumbai, India

[2]ME Student PIIT, New Panvel, Navi Mumbai, India

Abstract: ***Mining high utility item sets from a transactional database refers to the discovery of item sets with high utility like profits. Although a number of relevant algorithms have been proposed in recent years, they incur the problem of producing a large number of candidate item sets for high utility item sets. Such a large number of candidate item sets degrades the mining performance in terms of execution time and space requirement. The situation may become worse when the database contains lots of long transactions or long high utility item sets. In this paper, we propose two algorithms, namely UP-Growth (Utility Pattern Growth) and UP-Growth+, for mining high utility item sets with a set of effective strategies for pruning candidate item sets. The information of high utility item sets is maintained in a tree-based data structure named UP-Tree (Utility Pattern Tree) such that candidate item sets can be generated efficiently with only two scans of database. The performance of UP-Growth and UP Growth+ is compared with the state-of-the-art algorithms on many types of both real and synthetic datasets. Experimental results show that the proposed algorithms, especially UP-Growth+, not only reduce the number of candidates effectively but also outperform other algorithms substantially in terms of runtime, especially when databases contain lots of long transactions.***

Keywords: Candidate pruning, frequent itemset, high utility itemset, utility mining, data mining.

1. Introduction

Data mining is the process of revealing nontrivial, previously unknown and potentially useful information from large databases. Discovering useful patterns hidden in a database plays an essential role in several data mining tasks, such as frequent pattern mining, weighted frequent pattern mining, and high utility pattern mining. Among them, frequent pattern mining is a fundamental research topic that has been applied to different kinds of databases, such as transactional databases [1] streaming databases and time series databases [9], [12], and various application domains, such as bioinformatics [8], [11], Web click-stream analysis [7] and mobile environments [15].Nevertheless, relative importance of each item is not considered in frequent pattern mining. To address this problem, weighted association rule mining was proposed [4]. In this framework, weights of items, such as unit profits of items in transaction databases, are considered. With this concept, even if some items appear infrequently, they might still be found if they have high weights. However, in this framework, the quantities of items are not considered yet. Therefore, it cannot satisfy the requirements of users who are interested in discovering the itemsets with high sales profits, since the profits are composed of unit profits, i.e., weights, and purchased quantities. In view of this, utility mining emerges as an important topic in data mining field. 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. In view of this, utility mining emerges as an important topic in data mining field. 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 externalutility, 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. Mining high utility itemsets from databases is an important task has a wide range of applications such as website click stream analysis [16] business promotion in chain hypermarkets, cross marketing in retail stores [3], [10] online e-commerce management, mobile commerce environment planning [24], and even finding important patterns in biomedical applications [5].However, mining high utility itemsets from databases is not an easy task since downward closure property [1] infrequent itemset mining does not hold. In other words, pruning search space for high utility itemset mining is difficult because a superset of a low-utility itemset may be a high utility itemset. A naive method to address this problem is to enumerate all itemsets from databases by the principle of exhaustion. Obviously, this method suffers from the problems of a large search space, especially when databases contain lots of long transactions or a low minimum utility threshold is set. Hence, how to effectively prune the search space and efficiently capture all high utility itemsets with no miss is a crucial challenge in utility mining Major contributions are summarized as follows:

1. Two algorithms, named utility pattern growth (UP Growth)and UP-Growth+, 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 itemset scan be generated from UP-Tree efficiently with only two scans of original databases.
2) Several strategies are proposed for facilitating the mining processes of UP-Growth and UP-Growth+ by maintaining only essential information in UP-Tree. By these strategies, overestimated 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.

3) Different types of both real and synthetic data sets are used in a series of experiments to compare the performance of the proposed algorithms with the state-of-the-art utility mining algorithms. Experimental results show that UP-Growth and UP-Growth+outperform other algorithms substantially in terms of execution time, especially when databases contain lots of long transactions or low minimum utility thresholds are set.

2. Existing System

The traditional association rule mining (ARM) 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. Though, frequency of occurrence may not express the semantics of applications, because the user's interest may be related to other factors, such as cost, profit, or aesthetic value. .For example, a sales manager may not be interested in frequent item sets that do not generate significant profit. The frequent item set mining approach may not satisfy a sales manager's goal. The support measure reflects the statistical correlation of items, but it does not reflect their semantic significance. In other words, statistical correlation may not measure how useful an item set is in accordance with a user's preferences (i.e., profit). The profit of an item set depends not only on the support of the item set, but also on the prices of the items in that item set.

In existing method Potential high utility itemsets (PHUIs) are found first, and then an additional database scan is performed for identifying their utilities. However, existing methods often generate a huge set of PHUIs and their mining performance is degraded consequently. This situation may become worse when databases contain many long transactions or low thresholds are set. The huge number of PHUIs forms a challenging problem to the mining performance since the more PHUIs the algorithm generates, the higher processing time it consumes.

An Example Database

Table 1.1

TID	Transaction	TU
T1	(A,1) (C,1) (D,1)	8
T2	(A,1) (C,1) (D,1)	27
T3	(A,1) (B,2) (C,1) (D,6) (E,1) (F,5)	30
T4	(B,4) (C,3) (D,3) (E,1)	20
T5	(B,2) (C,2) (E,1) (G,2)	11

Table 1.2: Profit Table

Item	A	B	C	D	E	F	G	H
Profit	5	2	1	2	3	5	1	1

Given a finite set of items I = {i1, i2… im}. Each item ip (1 ≤ p ≤ m) has a unit profit p (ip).An itemset X is a set of k distinct items {i1, i2… ik}, where $i_j \in I$, 1≤ j ≤ k, and k is the length of X. An itemset with length k is called k-itemset. A transaction database D = {T1, T2… Tn} contains a set of transactions, and each transaction Td (1 ≤ d ≤ n) has an unique identifier d, called TID.

Each item ip in the transaction Td is associated with a quantity q (ip, Td), that is, the purchased number of ip in Td.

Definition 1: The utility of an item ip in the transaction Td is denoted as u(ip, Td) and defined as p(ip) × q(ip, Td).

For example, in Table 1, u({A}, T1) = 5 × 1 = 5.

Definition 2: The utility of an itemset X in Td is denoted as u(X, T_d) and defined as

$$\sum_{i_p \in X \wedge X \subseteq T_d} u(i_p, T_d)$$

For example, u({AC}, T1) = u({A}, T1) + u({C}, T1) = 5 + 1 = 6.

Definition 3: The utility of an itemset X in D is denoted as

$$\sum_{X \subseteq T_d \wedge T_d \in D} u(X, T_d)$$

u(X) and defined as

For example, u({AD}) = u({AD}, T1) + u({AD}, T3) = 7 + 17 = 24.

Definition 4: An itemset is called a high utility itemset if its utility is no less than a user-specified minimum utility threshold which is denoted as min_util. Otherwise; it is called a low utility itemset.

3. Problem Statement

Given a transaction database D and a user-specified minimum utility threshold min_util, mining high utility itemsets from the transaction database is equivalent to discover from D all itemsets whose utilities are no less than min_util. After addressing the problem definition of utility mining, we introduce the transaction-weighted downward closure

Definition 5: The transaction utility of a transaction Td is denoted as TU (Td) and defined as u(Td, Td). For example, TU (T1) = u ({ACD}, T1) = 8.

Definition 6: The transaction-weighted utilization of an itemset X is the sum of the transaction utilities of all the transactions containing X, which is denoted as TWU(X) and defined as

$$\sum_{X \subseteq T_d \wedge T_d \in D} TU(T_d)$$

For example, TWU ({AD}) = TU (T1) + TU (T3) = 8 + 30 = 38. If TWU(X) is no less than the minimum utility threshold, X is called a high transaction-weighted utilization itemset (abbreviated as HTWUI).

Definition 7: The transaction-weighted downward closure, which is abbreviated as TWDC, is stated as follows. For any itemset X, if X is not a HTWUI, any superset of X is a low utility itemset. By this definition, the downward closure property can be maintained by using transaction-weighted utilization. For example, in Table 1, any superset of {AD} is a low utility itemset since TWU ({AD}) < min_util.

4. Proposed System

The goal of utility mining is to discover all the high utility itemsets whose utility values are beyond a user specified threshold in a transaction database. The state-of-the-art algorithms are for mining all high utility item sets. FUFM and FUMF algorithms use both the statistical and the utility measures. UP-Growth and UP-Growth+ this gives a new perspective in analyzing the item sets. 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 are generated. Then Customer Relationship Management (CRM) is incorporated into the system by tracking the customers who are frequent buyers of the different kinds of item sets.

4.1 UP-Growth Algorithm

The UP-Growth is one of the efficient algorithms to generate high utility itemsets depending on construction of a global.

(Phase 1) The framework of UP-Tree follows three steps:

1) Construction of UP-Tree
2) Generate PHUIs from UP-Tree.
3) 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 (not less than min_sup) from PHUIs values. The global UP-Tree contains many sub paths. Each path is considered from bottom node of header table. This path is named as conditional pattern base (CPB).

4.2 UP-Growth + Algorithm

Although DGU and DGN strategies are efficiently reduce the number of candidates in Phase 1(i.e., global UP-Tree). But they cannot be applied during the construction of the local UP-Tree (Phase-2). Instead use, DLU strategy (Discarding local unpromising items) to discarding utilities of low utility items from path utilities of the paths and DLN strategy (Discarding local node utilities) to discarding item utilities of descendant nodes during the local UP-Tree construction. Even though, still the algorithm facing some performance issues in phase-2. To overcome this, maximum transaction weight utilizations (MTWU) are computed from all the items and considering multiple of min_sup as a user specified threshold value as shown in algorithm. By this modification, performance will increase compare with existing UP-Tree construction also improves the performance of UP-growth algorithm. An improved utility pattern growth is abbreviated as IUPG.

4.3 Algorithm

Input: Transaction database D, user specified threshold.
Output: high utility itemsets.
Begin
1. Scan database of transactions Td&D
2. Determine transaction utility of Td in D and TWU of itemset (X)
3. Compute min_sup (MTWU * user specified threshold)
4. If (TWU(X) ≤ min_sup) then Remove Items from transaction database
5. Else insert into header table H and to keep the items in the descending order.
6. Repeat step 4 & 5 until end of the D.
7. Insert Td into global UP-Tree
8. Apply DGU and DGN strategies on global UP- tree
9. Re-construct the UP-Tree
10. **For** each item ai in H do
11. Generate a PHUI Y= X U ai
12. Estimate utility of Y is set as ai's utility value in H
13. Put local promising items in Y-CPB into H
14. Apply strategy DLU to reduce path utilities of the paths
15. Apply strategy DLN and insert paths into Td
16. If Td ≠ null then call for loop

End for
End

5. Conclusion

We have proposed efficient algorithms named UP-Growth and UP-Growth+ for mining high utility itemsets from transaction databases. A data structure named UP-Tree was proposed for maintaining the information of high utility itemsets. PHUIs can be efficiently generated from UP-Tree with only two database scans. Moreover, we developed several strategies to decrease overestimated utility and enhance the performance of utility mining. In the experiments, both real and synthetic data sets were used to perform a thorough performance evaluation. Results show that the strategies considerably improved performance by reducing both the search space and the number of candidates. Moreover, the proposed algorithms, especially UP-Growth+, outperform the state of the art algorithms substantially especially when databases contain lots of long transactions or a low minimum utility threshold is used.

References

[1] Vincent S. Tseng, Bai-En Shie, Cheng-Wei Wu, and Philip S. Yu, Fellow, IEEE:2013 Efficient Algorithms for Mining High UtilityItemsets from Transactional Databases
[2] Vincent. S. Tseng, C. W. Wu, B. E. Shie, and P. S. Yu.: UP-Growth: An Efficient Algorithm for High Utility Itemset Mining. In Proc. of ACM-KDD, Washington, DC, USA, pp. 253-262, July 25–28, 2010.
[3] 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.

[4] 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," Proc. IEEE Eighth Int'l Conf. on Data Mining, pp. 881- 886, 2008.
[5] Y.-C. Li, J.-S.Yeh, and C.-C. Chang, "Isolated Items Discarding Strategy for Discovering High Utility Itemsets," Data and Knowledge Eng., vol. 64, no. 1, pp. 198-217, Jan. 2008.
[6] Y. C. Li, J. S. Yeh, and C. C. Chang.: Isolated items discarding strategy for discovering high utility itemsets. In Data & Knowledge Engineering, Vol. 64, Issue 1, pp. 198-217, Jan., 2008.
[7] S. J. Yen and Y. S. Lee.: Mining high utility quantitative association rules. In Proc. of 9th Int'l Conf. on Data Warehousing and Knowledge Discovery, Lecture Notes in Computer Science 4654, pp. 283-292, Sep., 2007.
[8] S.C. Lee, J. Paik, J. Ok, I. Song, and U.M. Kim, "Efficient Mining of User Behaviors by Temporal Mobile Access Patterns," Int'l J. Computer Science Security, vol. 7, no. 2, pp. 285-291, 2007.
[9] H. Yao, H. J. Hamilton, and L. Geng.: A unified framework for utility-based measures for mining itemsets. In Proc. of ACM SIGKDD 2nd Workshop on Utility-Based Data Mining, pp. 28-37, USA, Aug., 2006.
[10] Y. Liu, W. Liao, and A. Choudhary.: A fast high utility itemsets mining algorithm. In Proc. of the Utility-Based Data Mining Workshop, 2005
[11] C.H. Lin, D.Y. Chiu, Y.H. Wu, and A.L.P. Chen, "Mining Frequent Itemsets from Data Streams with a Time-Sensitive Sliding Window," Proc. SIAM Int'l Conf. Data Mining (SDM '05), 2005.
[12] R. Chan, Q. Yang, and Y. Shen.: Mining high utility itemsets. In Proc. of Third IEEE Int'l Conf. on Data Mining, pp. 19-26, 2003.

Author Profile

Prof. Madhu M Nashipudimath has Bachelors and Master's Degree in the field of Computer science and Engineering. She has keen interest in the area of data mining, software Engg, fuzzy systems, neural networks, and has published several papers in National and International conferences and journals. Her interest is also extended to data storage and information retrieval. Prof. Madhu has more than fifteen years of experience in teaching undergraduate as well as postgraduate students. She has attended several workshops and faculty development programs. She has 20 Research Papers to her credit.

Prof. Monali Deshmukh, Bachelor of Engineering, Student, PIIT, Panvel Navi Mumbai. Her area of Interest includes Data Mining.