International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue-01
January 2018

A Study on Top-K High Utility Itemsets Mining

Ameena Aiman & Dr. Raafiya Gulmeher
[1]*M. Tech Student KBNCE, Kalaburagi*
[2]*CSEDept., KBNCE, Kalaburagi*

Abstract:

High utility itemsets (HUIs) mining is an emerging topic in data mining, which refers to discovering all itemsets having a utility meeting a user-specified minimum utility threshold min_util. However, setting min_util appropriately is a difficult problem for users. Finding an appropriate minimum utility threshold by trial and error is a tedious process for users. If min_util is set too low, too many HUIs will be generated, which may cause the mining process to be very inefficient. On the other hand, if min_util is set too high, it is likely that no HUIs will be found. In this paper, we study top-k high utility itemset mining, where k is the desired number of HUIs to be mined.

Keywords

Data mining, high-utility itemset, pattern growth.

1. INTRODUCTION

Today, the information has become superabundant in all sectors such as, from business transactions and scientific data, to satellite pictures, text reports and military intelligence. Though it brings huge new benefits but, it also creates big headache. So, it is very important to handle the huge data sets in an effective manner. Hence, Data mining is the way that helps in discovering relevant patterns from large data sets. In simple words, it is the process of analyzing data from different perspectives and summarizing it into useful information. The overall goal of the data mining process is to extract information from a data set and transform it into an understandable structure for further use. Aside from the raw analysis step, it involves database and data management aspects, data pre-processing, model and inference considerations, interestingness metrics, complexity considerations, post-processing of discovered structures, visualization, and online updating. Technically, the goal is the extraction of patterns and knowledge from large amount of data, not the extraction of data itself. Data mining is primarily used today by companies with a strong consumer focus - retail, real estate, financial transactions, banking, telecommunications and marketing organizations. It enables these companies to determine relationships among "internal" factors such as price, product positioning, or staff skills, and "external" factors such as economic indicators, competition, and customer demographics. And, it enables them to determine the impact on sales, customer satisfaction, and corporate profits. Finally, it enables them to "drill down" into summary information to view detail transactional data. Discovering useful patterns hidden in a database plays an essential role in several data mining tasks, such as high utility pattern mining. Mining high utility itemsets from a transactional database refers to the discovery of itemsets with high utility like profits. The following description brings the importance of utility mining which overcomes the drawbacks of frequent itemset and weighted association rule mining. Frequent patterns [6] are itemsets, which appear in a data set with frequency no less than a user-specified threshold. For example, a set of items, such as milk and bread that appear frequently together in a transaction data set is a frequent itemsets. Finding frequent patterns plays an essential role in mining associations, correlations, and many other interesting relationships among data. Frequent pattern mining helps in data indexing, classification, clustering, and other data mining tasks such as mining association rules [2]. The applications of this kind of mining are in the field of telecommunications, text analysis and census analysis. The interestingness metric used to express the frequency of itemsets is in terms of support value of the itemsets. The Support value of an itemset is the percentage of transactions that contain the itemset.
Given a finite set of items I = {i1, i2, …. im} each item ip ($1 \le p \le m$) has a unit profit pr (ip) An itemset X is a set of k distinct items {i1, i2, …. ik}, where k is the length of X. An itemset with length k is called a k-itemset. A transaction database D = {T1, T2,…, Tn} contains a set of transactions, and each transaction Td has a unique identifier d, called TID. Each item ip in transaction Td is associated with a quantity q (ip, Td) that is, the purchased quantity of ip in Td.

Definition 1: Utility of an item ip in a transaction Td is the product of unit profit and the purchased quantity. It is denoted as u (ip, Td) and defined as pr (ip, Td) ×q (ip, Td)

Definition 2: Utility of an itemset X in Td is the sum of utilities of the transaction containing X. It is denoted as U (X, Td) and defined as ∑ipƐX∧X⊆ Td u (ip, Td)

Definition 3: Utility of an itemset X in D is the sum of utilities of all the transactions containing X in the database D. It is denoted as u(X) and ∑X⊆Td∧TdƐD u (X, Td).

Definition 4: An itemset is called a high utility itemset if its utility is no less than a user-specified minimum utility threshold or else it is low-utility itemset.

Definition 5: Transaction utility of a transaction Td is the sum of utilities of all the items in the transaction Td. It is denoted as TU (Td) and defined as u (Td, Td)

Definition 6: Transaction-weighted utility 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 ∑X⊆Td∧TdƐD TU (Td)

Definition 7: An itemset X is called a high-transaction weighted utility itemset (HTWUI) if TWU(X) is no less than minimum utility threshold.

2. LITERATURE SURVEY

The paper [2] proposed Apriori algorithm, is used to obtain frequent itemsets from the database. This algorithm simply counts item occurrences to further determine the large one itemsets which requires scanning of database each time for each item. Next, the database scan is performed to count the support of candidate's itemsets. Then the association rules are generated from frequent itemsets. After identifying the large itemsets, only those itemsets which have the support greater than the minimum support are allowed. Drawback: It generates a lot of candidate item sets and scans database every time and when a new transaction is added to the database then it should rescan the entire database again.
The paper [3] proposed an efficient FP-tree based mining method that builds frequent pattern tree (FP-tree) structure, for storing crucial information about frequent patterns into compressed 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. Drawback: FP-Growth Consumes more memory and performs badly with long pattern data sets. Thus, it is not able to find high utility itemsets.
The paper [4] proposed Two-Phase algorithm to efficiently prune down the number of candidates and can precisely obtain the complete set of high utility itemsets. In Phase I, High transaction-weighted utilization itemsets (HTWUIs) are identified. The size of candidate set is reduced by only considering the supersets of high transaction-weighted utilization itemsets. In Phase II, one database scan is performed to filter out the high transaction-weighted utilization itemsets that are indeed low utility itemsets. Drawback: It generates numerous candidates to obtain HTWUIs and requires multiple database scans. Traditional methods of association rule mining consider the appearance of an item in a transaction, whether or not it is purchased, as a binary variable.
However, customers may purchase more than one of the same item, and the unit cost may vary among items. Hence, developing an efficient algorithm is crucial for utility mining.
To overcome this problem, the paper [5] proposed isolated items discarding strategy (IIDS) to reduce the number of candidates. By pruning isolated items during level wise search, the number of candidate itemsets for HTWUIs in phase one can be reduced. Drawback: This algorithm still scans database for several times and uses a candidate generation-and-test scheme to find high utility itemsets and thus cannot improved performance.
The paper [1] proposed a tree-based algorithm, named Incremental High Utility Pattern (IHUP). A tree based structure called IHUP-Tree is used to maintain the information about itemsets and their utilities. Drawback: Though it achieves a better performance than IIDS and Two-Phase, it still produces several HTWUIs. Since the overestimated utility calculated by TWU is too large.

3. RELATED WORK

The traditional FIM (Frequent itemset mining) may discover a large amount of frequent but low-value itemsets and lose the information on valuable itemsets having low selling frequencies. Hence, it cannot satisfy the requirement of users who desire to discover itemsets with high utilities such as high profits.

To address these issues, utility mining emerges as an important topic in data mining and has received extensive attention in recent years. In utility mining, each item is associated with a utility (e.g. unit profit) and an occurrence count in each transaction (e.g. quantity).

The utility of an itemset represents its importance, which can be measured in terms of weight, value, quantity or other information depending on the user specification. An itemset is called high utility itemset (HUI) if its utility is no less than a user-specified minimum utility threshold min_util.

In recent years, high utility itemset mining has received lots of attention and many efficient algorithms have been proposed, such as Two-Phase, IHUP, IIDS, UPGrowth, d2HUP and HUI-Miner. These algorithms can be generally categorized into two types: two-phase and one-phase algorithms.

TKO (mining Top-K utility itemsets in One phase):

The TKO Base algorithm takes as input the parameter k and a transactional database D in horizontal format. But if a database has already been transformed into vertical format such as initial utility-lists, TKOBase can directly use it for mining top-k HUIs. TKO Base initially sets the min_util Border threshold to 0 and initializes a min-heap structure TopK-CI-List for maintaining the current top-k HUIs during the search. The algorithm then scans D twice to build the initial utility-lists F-ULs. Then, TKOBase explores the search space of top-k HUI using a procedure that we name TopK-HUI-Search. It is the combination of a novel strategy named RUC (Raising threshold by Utility of Candidates) with the HUI-Miner search procedure. During the search, TKOBase updates the list of current top-k HUIs in TopK-CI-List and gradually raises the min_utilBorder threshold by the information of TopK-CI-List. When the algorithm terminates, the TopK-CI-List captures the complete set of top-k HUIs in the database.

TKU algorithm:

The TKU algorithm adopts a compact tree-based structure named UP-Tree to maintain the information of transactions and utilities of itemsets. TKU inherits useful properties from the TWU model and consists of two phases.In phase I, potential top-k high utility itemsets (PKHUIs) are generated. In phase II, top-k HUIs are identified from the set of PKHUIs discovered in phase I.

4.CONCLUSION

In this paper, we have studied about top-k high utility itemsets mining, where k is the desired number of high utility itemsets to be mined. TKU is the first two-phase algorithm for mining top-k high utility itemsets, which incorporates five strategies PE, NU, MD, MC and SE to effectively raise the border minimum utility thresholds and further prune the search space. On the other hand, TKO is the first one-phase algorithm developed for top-k HUI mining, which integrates the novel strategies RUC, RUZ and EPB to greatly improve its performance. Empirical evaluations on different types of real and synthetic datasets show that the algorithms have good scalability on large datasets and the performance of the algorithms is close to the optimal case of the state-of-the art two-phase and one-phase utility mining algorithms.

5. REFERENCES

[1] C. Ahmed, S. Tanbeer, B. Jeong, and Y. Lee, "Efficient tree structures for high-utility pattern mining in incremental databases," IEEE Trans. Knowl. Data Eng., vol. 21, no. 12, pp. 1708–1721, Dec. 2009.
[2] K. Chuang, J. Huang, and M. Chen, "Mining top-k frequent patterns in the presence of the memory constraint," VLDB J., vol. 17, pp. 1321–1344, 2008.
[3] P. Fournier-Viger and V. S. Tseng, "Mining top-k sequential rules," in Proc. Int. Conf. Adv. Data Mining Appl., 2011, pp. 180–194.
[4] P. Fournier-Viger, C. Wu, and V. S. Tseng, "Mining top-k association rules," in Proc. Int. Conf. Can. Conf. Adv. Artif. Intell., 2012, pp. 61–73.
[5] P. Fournier-Viger, C. Wu, and V. S. Tseng, "Novel concise representations of high utility itemsets using generator patterns," in Proc. Int. Conf. Adv. Data Mining Appl. Lecture Notes Comput. Sci., 2014, vol. 8933, pp. 30–43.
[6] G. Lan, T. Hong, V. S. Tseng, and S. Wang, "Applying the maximum utility measure in high utility sequential pattern mining," Expert Syst. Appl., vol. 41, no. 11, pp. 5071–5081, 2014.