PROFICIENT SCHEME FOR EXTRACTING TOP-K GREAT VALUE ITEM SETS

SAYEEDA SAJEEDUNNISA
M. Tech Dept of CSE
Assistant Professor at VIF College of engineering and technology
shaziyasyed09@gmail.com

ABSTRACT

High utility item sets (HUIs) withdrawal is a developing topic in data mining, which refers to discovering all item sets having a utility meeting a user-specified minimum utility threshold min util. However, setting min until appropriately is a difficult problem for users. Generally speaking, finding an appropriate minimum utility threshold by trial and error is a tedious process for users. If minutia is set too low, too many HUIs will be generated, which may cause the mining process to be very in efficient. On the other hand, if min until is set too high, it is likely that no HUIs will be found. In this paper, we deal with the above issues via providing a new framework for top-ok high utility item set mining, in which k is the desired quantity of HUIs to be mined. Two varieties of efficient algorithms named TKU (mining Top-K Utility item sets) and TKO (mining Top-K utility item sets in one section) are proposed for mining such item sets without the want to set minutia. We provide a structural evaluation of the two algorithms with discussions on their benefits and barriers. Empirical reviews on each real and artificial datasets display that the overall performance of the proposed algorithms is near that of the greatest case of modern day utility mining algorithms.

Keywords:—*Utility mining, high utility item set mining, top-k pattern mining, top-k high utility item set mining.*

I. INTRODUCTION:

Frequent item set mining (FIM) is a fundamental research topic in statistics mining. However, the conventional FIM may find out a massive quantity of frequent but low-value itemsets and lose the information on valuable item sets having low promoting frequencies. Hence, it can't satisfy the requirement of users who choice to find out itemsets with excessive utilities inclusive of excessive profits. To address those problems, application mining emerges as an important subject matter in statistics mining and has obtained giant attention in latest years. In application mining, each item is associated with an application (e.g. Unit profit) and a prevalence count in each transaction (e.g. Amount). The application of an item setrepresents its significance,

which may be measured in terms of weight, price, amount or other information depending on the consumer specification. An item set is called high utility item set (HUI) if its software is no much less than a person-specified minimum software threshold minutia. HUI mining is critical to many programs consisting of streaming evaluation marketplaceanalysis cellularcomputing and biomedicine [1]. However, efficiently mining HUIs in databases is not an easy assignment because the downward closure assets used in FIM does now not maintain for the software of item sets. In different words, pruning seek space for HUI mining is difficult because a superset of a low software item set may be high software. To address this hassle, the concept of transaction weighted usage (TWU) version become introduced to facilitate the overall performance of the mining challenge. In this model, an item set is known as high transaction-weighted usage item set (HTWUI) if its TWU is not any much less than minutia, wherein the TWU of an item set represents a higher bound on its software. Therefore, a HUI have to be a HTWUI and all the HUIs have to be blanketed inside the entire set of HTWUIs. A classical TWU model-primarily based algorithm consists of stages. In the first segment, referred to as phase I, the complete set of HTWUIs are determined. In the second one section, referred to as phase II, all HUIs are acquired by way of calculating the precise utilities of HTWUIs with one database scan. Although many studies had been dedicated to HUI mining, it's miles difficult for customers to pick the best minimum software threshold in practice. Depending on the threshold, the output size can be very small or very big. Besides, the selection of the edge greatly influences the overall performance of the algorithms. If the edge is set too low, too many HUIs could be presented to the users and it is difficult for the customers to realize the consequences. A huge number of HUIs additionally reasons the mining algorithms to become inefficient or maybe run out of reminiscence, because the more HUIs the algorithms generate, the greater resources they eat. On the opposite, if the threshold is about too high, no HUI will be determined. To find the perfect cost for the minutia threshold, customers want to strive extraordinary thresholds via guessing and re-executing the algorithms over and over till being satisfied with the results. This process is both inconvenient and time-ingesting. To precisely manipulate the

output size and discover the item sets with the best utilities without putting the thresholds, a promising solution is to redefine the mission of mining HUIs as mining top-ok high software item sets (top-okay HUIs). The concept is to permit the users specify k, i.e., the variety of favored item sets, in preference to specifying the minimum software threshold. Setting okay is extra intuitive than setting the edge because ok represents the quantity of item sets that the customers need to find while deciding on the threshold depends mainly on database characteristics, that are regularly unknown to users. Using a parameter k rather than the minutia threshold is very proper for lots applications. For example, to analyze client buy behavior, pinnacle-ok HUI mining serves as a promising solution for users who preference to recognize "What are the top-ok sets of merchandise (i.e., item sets) that contribute the highest profits to the employer?" and "How to efficiently find these item sets without placing the minutia threshold?". Although top-k HUI mining is vital to many programs, developing efficient algorithms for mining such styles isn't a clean assignment. It poses 4 essential demanding situations as discussed below. First, the utility of item sets is neither monotone nor ant monotone [2]. In different phrases, the utility of an item set may be equal to, higher or lower than that of its supersets and subsets. Therefore, many techniques developed in pinnacle-k frequent pattern mining that rely on anti-monotonicity to prune the quest space cannot be directly carried out to pinnacle-k excessive application item set mining. The second venture is the way to include the idea of pinnacle-k sample mining with the TWU model. Although the TWU model is broadly utilized in software mining, it is difficult to adapt this version to top-okay HUI mining due to the fact the exact utilities of item sets are unknown in section I. When a HTWUI is generated in segment I, we cannot guarantee that its application is better than different HTWUIs and that it's far a top-okay HUI before performing section II. To guarantee that each one the pinnacle-okay HUIs may be captured within the set of HTWUIs, a naive technique is to run the set of rules with minutia ¼ 0. However, this approach might also face the trouble of a very big seek area. The third task is that the minutia threshold isn't given earlier in pinnacle-ok HUI mining. In conventional HUI mining, the hunt space may be efficiently pruned by using the algorithms by way of using a given

minutia threshold. However, within the situation of top-okay HUI mining, no minutia threshold is furnished earlier. Therefore, the minimal software threshold is initially set to zero and the designed algorithm has to step by step enhance the brink to prune the search space. Such a threshold is an internal parameter of the designed algorithm and is called the border minimum software threshold minutiaBorder in this paper. It isn't the same as the outside parameter minutia this is given by means of users in advance. If an set of rules can't boost the minutiaBorder threshold effectively and efficiently, it'd produce too many intermediate low utility item sets for the duration of the mining process, which may degrade its performance in terms of execution time and reminiscence usage. Thus the project is to layout effective techniques that could increase the minutia threshold as excessive as viable and as quick as possible, and in addition lessen as.

1. BACKGROUD WORK:

This phase introduces associated works approximately pinnacle-ok excessive utility item set mining, together with high software item set mining, pinnacle-ok common sample mining and pinnacle-k excessive utility item set mining.

High Utility Item set Mining In latest years, high utility item set mining has obtained plenty of attention and many efficient algorithms have been

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

Proposed, including Two-Phase [13], IHUP, IIDS, UP Growth, d2HUP and HUI-Miner. These algorithms may be usually classified into two kinds: twophase and one-segment algorithms. The important characteristic of two-section algorithms is that they include two phases. In the first phase, they generate a set of applicants that are capacity excessive application itemsets. In the second one phase, they calculate the precise software of each candidate found in the first phase to discover high utility itemsets. Two-Phase, IHUP, IIDS and UP-Growth are -segment based algorithms. UPGrowth is one of the present day -segment algorithms, which includes four powerful techniques DGU, DGN, DLU and DLN for pruning

candidates inside the first phase. One the opposite, the principle characteristic of one-section algorithms is they discover high utility itemsets using only one segment and convey no candidates. D2HUP and HUI-Miner are one-section algorithms. D2HUP transforms a horizontal database into a tree-based totally structure referred to as CAUL [15] and adopts a sample-increase approach to without delay find out high application itemsets in databases. HUI-Miner considers a database of vertical layout and transforms it into application-lists [3]. The software-list shape utilized in HUI-Miner permits immediately computing the utility of generated itemsets in essential reminiscence without scanning the authentic database. Although the above studies may additionally perform nicely in a few packages, they are not advanced for pinnacle-k high application itemset mining and still be afflicted by the diffused problem of setting suitable thresholds

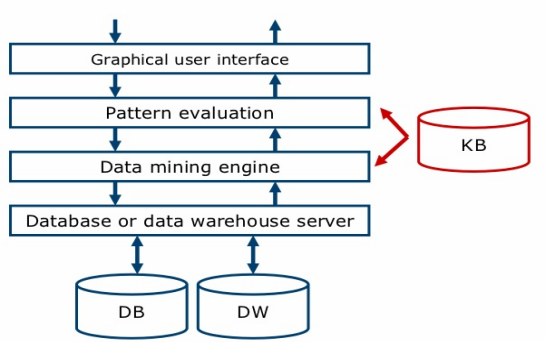
Top-k Pattern Mining:

Many research have been proposed to mine different kinds of top-okay patterns, consisting of pinnacle-k frequent item sets pinnacle-okay common closed itemsets top-ok closed sequential patterns pinnacle-okay affiliation policies top-ok sequential rules , top-ok correlation patterns and pinnacle-ok cosine similarity exciting pairs . What distinguishes every pinnacle-okay sample mining algorithm is the form of styles determined, as well as the records structures and seek techniques which are employed. For instance, a few algorithms use a rule expansion strategy for finding patterns, at the same time as others depend on a pattern-boom search the use of systems which include FP-Tree [4]. The desire of statistics structures and search strategy affect the efficiency of a pinnacle-okay pattern mining algorithm in terms of each memory and execution time. However, the above algorithms discover top-okay styles according to standard measures in preference to the software degree. As a outcome, they'll miss styles yielding excessive application.

2. THE TKU ALGORITHM:

In this phase, we propose an efficient algorithm named TKU (mining Top-okay Utility itemsets) for discovering pinnacle-k HUIs without specifying min_util. We first present its simple version named TKUBase and then describe the TKU set of rules, which includes several novel techniques [5].

3.1 The Baseline Approach TKUBase The baseline approach TKUBase is an extension of UPGrowth [25], a tree-based set of rules for mining HUIs. TKUBase adopts the UP-Tree structure of UP-Growth to preserve the facts of transactions and pinnacle-ok HUIs. TKUBase is finished in three steps: (1) building the UP-Tree, (2) generating capacity pinnacle-ok excessive software itemsets (PKHUIs) from the UP-Tree, and (3) identifying pinnacle-okay HUIs from the set of PKHUIs.



3.1.1 UP-Tree Structure

Then, we briefly introduce the UP-Tree shape. For extra details about it, readers are noted. Each node N of a UP-Tree has five entries: N.Name is the object name of N; N. Count number is the aid rely of N; N.Nu is the node application of N; N.Figure suggests the determine node of N; N.Hlink is a node link which may additionally point to a node having the same object call as N.Call. The Header desk is a shape hired to facilitate the traversal of the UP-Tree. A header table access includes an object call, an expected software fee, and a hyperlink [6]. The link factors to the first node within the UP-Tree having the same item call because the entry. The nodes whose object names are the equal may be traversed efficiently by using following the links in header table and the node links in the UP-Tree.

Performance Comparison of the REPT, TKU, and TKO Algorithms

In this section, we evaluate the performance of the proposed algorithms TKU and TKO against REPT and two stateof-the-art HUI mining algorithms UP-Growth and HUI-Miner. Here, HUI-Miner(Opt) and UP-Growth (Opt) respectively represents HUI-Miner and UP-Growth tuned with the optimal thresholds. Besides, REPT with varied N ¼ y (i.e., the parameter for the RSD strategy) is denoted as REPT (N¼y).

3. CONCLUSION:

In this Section, we have studied the problem of top-okay excessive utility itemsets mining, where is the preferred variety of excessive software item sets to be mined. Two efficient algorithms TKU (mining Top-K Utility item sets) andTKO

(mining Top-K utility itemsets in one section) are proposed for mining such itemsets without placing minimum software thresholds. TKU is the first two-section algorithm for mining top-k high software item sets, which includes five techniques PE, NU, MD, MC and SE to successfully raise the border minimum software thresholds and further prune the quest area. On the alternative hand, TKO is the first one-segment algorithm advanced for top-k HUI mining, which integrates the unconventional techniques RUC, RUZ and EPB to substantially enhance its overall performance. Empirical evaluations on one-of-a-kind forms of real and artificial datasets show that the proposed algorithms have accurate scalability on large datasets and the performance of the proposed algorithms is close to the optimum case of the nation-of-the art two-phase and one-segment application mining algorithms.

4. REFERENCES:

[1] R. Chan, Q. Yang, and Y. Sheen, "Mining high-utility item sets," in Proc. IEEE Int. Conf. Data Mining, 2003, pp. 19–26.

[2] P. Fournier-Vigor and V. S. Tseng, "Mining top-k sequential rules," in Proc. Int. Conf. Adv. Data Mining Appl., 2011, pp. 180–194.

[3] P.Fournier-Vigor,C.Wu,andV.S.Tseng, "Miningtop-kassociation rules, pinprick.Int.Conf.Can.Conf.Adv.Artif.Intell" 2012,pp.61–73.

[4] S. Krishnamoorthy, "Pruning strategies for mining high utility itemsets," Expert Syst. Appl., vol. 42, no. 5, pp. 2371–2381, 2015.

[5] C. Lin, T. Hong, G. Lan, J. Wong, and W. Lin, "Efficient updating of discovered high-utility itemsets for transaction deletion in dynamic databases," Adv. Eng. Informat., vol. 29, no. 1, pp. 16–27, 2015.

[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.