SIMULATION AND ANALYSIS OF EFFICIENT ALGORITHMS FOR MINING TOP-K HIGH UTILITY ITEMSETS

Surbhi Choudhary¹, Devendra Nagal², Swati Sharma³
¹PhD Research Scholar, Dept. of Computer Applications, JNU Jodhpur
²,³Faculty, Dept. of Electrical Engineering, JNU Jodhpur

Abstract: High utility sequential pattern mining is an emerging topic in the data mining community. Compared to the classic frequent sequence mining, the utility framework provides more informative and actionable knowledge since the utility of a sequence indicates business value and impact. However, the introduction of “utility” makes the problem fundamentally different from the frequency-based pattern mining framework and brings about dramatic challenges. Although the existing high utility sequential pattern mining algorithms can discover all the patterns satisfying a given minimum utility, it is often difficult for users to set a proper minimum utility. A too small value may produce thousands of patterns, whereas a too big one may lead to no findings. In this paper, we propose a novel framework called top-k high utility sequential pattern mining to tackle this critical problem. Accordingly, an efficient algorithm, Top-k high Utility Sequence (TUS for short) mining, is designed to identify top-k high utility sequential patterns without minimum utility. In addition, three effective features are introduced to handle the efficiency problem, including two strategies for raising the threshold and one pruning for filtering unpromising items. Our experiments are conducted on both synthetic and real datasets.

I. INTRODUCTION

Frequent sequential pattern mining [1], as one of the fundamental research topics in data mining, discovers frequent subsequences in sequence databases. It is very useful for handling order-based business problems, and has been successfully adopted to various domains and applications such as complex behavior analysis [1] and gene sequence analysis [2], [3], [4]. In the frequency-based framework, for typical sequence analysis, the downward closure property (also known as Apriori property) [1] plays a fundamental role in identifying frequent sequential patterns. However, taking the frequency to measure pattern interestingness may be insufficient for selecting actionable sequences associated with expected quality and business impact, because the patterns identified under the frequency (support) framework do not disclose the business value and impact. To solve the above problems, the concept of utility is introduced into sequential pattern mining to select sequences of high utility by considering the quality and value (such as profit) of items. This leads to an emerging area, high utility pattern mining [1], [2], [3], [8], [9] and high utility sequential pattern mining, which selects interesting patterns/sequential patterns based on minimum utility rather than minimum support. The utility-based patterns are proven to be more informative and actionable for decision-making than the frequency-based ones [2]. For instance, in [4], [5], the authors discuss the extraction of profitable behaviors from the mobile commerce environment. [4] Proposes methods to mine high utility sequences from web logs by assigning each page an impact/significance. In [6], a USpan algorithm is built for utility-based sequential pattern mining satisfying a predefined minimum utility. Although algorithms such as USpan can obtain high utility sequences based on a given minimum utility, it is very difficult for users to specify an appropriate minimum utility threshold and to directly obtain the most valuable patterns. This is because the complexity of utility-based sequence databases (which may be different from the classic itemsets), determining multiple factors including the distribution of the items and utilities, the density of the database, lengths of these sequences, and so on. Consequently, it is not surprising that, with a same minimum utility threshold, some datasets may produce millions of patterns while others may contribute nothing. The challenge here is that it may not be doable to tune the threshold to capture the expected number of patterns. This is because the sensitivity of the threshold makes it hard to tune for a variety of databases. It may be much costly and time consuming to achieve the proper threshold for the desired patterns. In fact, the classic frequency/support based pattern mining also faces the same challenge. Accordingly, the concept of extracting top-k patterns has been proposed in [2], [4], [7], [8] to select the patterns with the highest frequency. In the top-k frequent pattern mining, instead of letting users specify a threshold, the top-k pattern selection algorithm allows a user to set the number of top-k high frequency patterns to be discovered. This makes it much easier and more intuitive and practical than determining a minimum support; also the determination of k by a user is more straightforward than considering data characteristics, which are often invisible to users, for choosing a proper threshold. The easiness for users to determine k does not indicate the simplicity of developing an efficient algorithm for selecting top-k high utility sequential patterns. In the utility framework, TKU is the only method for mining top-k high utility itemsets, to the best of our knowledge. Nowork is reported on mining top-k high utility sequences. There is significant difference between top-k utility itemset mining and top-k utility sequence mining in which the order between itemsets is considered. In fact, the problem of top-k high utility sequence mining is much more challenging than mining top-k high utility itemsets. First, as with high utility itemset mining, the downward closure property does not hold in the utility-based sequence mining. This means that the existing top-k frequent sequential pattern mining algorithms [7] cannot be directly applied. Second,
compared to top-k high utility itemset mining [8], utility-based sequence analysis faces the critical combinational explosion and computational complexity caused by sequencing between itemsets. This means that the techniques in [9] cannot be directly transferred to top-k high utility sequential pattern mining either. Third, since the minimum utility is not given in advance, the algorithm essentially starts the searching from 0 minimum supports. This not only incurs very high computational costs, but also the challenge of how to raise the minimum threshold without missing any top-k high utility sequences.

II. SIMULATION RESULTS
Comparison with UPGrowth and UPGrowth+ Algorithm

In this project, two algorithms utility pattern growth (UP-Growth) and UP-Growth+, for mining high utility itemsets are used to compare. Performance of UP-Growth and UP-Growth+ become more efficient since database contain long transactions and generate fewer number of candidates than FP-Growth.

Two important problems are always in consideration first is how minimize number no of candidates and another is how to remove space and time complexity. Also, choosing an appropriate minimum utility threshold is a difficult task for application users: if the threshold is high, there might be no HUI; if the threshold is low, there might result too many HUIs, and the mining performance might be severely affected, even leading to memory overflow. It would also be a time-consuming task if one tries to determine the threshold value through various testing calculations.

To address this issue, Wu [10] proposes top-k algorithm, mining the top k itemsets with the highest utility values without presetting the minimum threshold.

Materials and Methods

Frequent itemset mining has been studied extensively in data mining. Recent studies have shown that it is more desirable to mine closed itemsets than the complete set of frequent itemsets. Efficient methods for mining closed itemsets, such as CLOSET, CHARM, and CLOSET+, have been developed. However, these methods all require a user-specified support threshold. Hidber presented Carma, an algorithm for online association rule mining, in which, a user can change the support threshold any time during the first scan of the data set (in other words, Carma still needs the user to specify the final support threshold), but its performance is worse than Apriori in general. In comparison with Carma, our algorithm does not need users to provide any minimum support and, in most cases, runs faster than two efficient algorithms, CHARM and CLOSET+ (running at the best tuned min_support thresholds), which, in turn, outperform Apriori substantially. Recently, there are proposals on association rule mining without support requirement, which are aimed at discovering confident rules instead of significant rules. As a result, they only use the confidence threshold to prune rules of small confidence. Our motivation is different because our algorithm still targets at mining significant rules, but we do not need an user to specify any min_support threshold.

The problem of mining top-k frequent itemsets has attracted the attention of some researchers recently. Fu et al.[12] studied mining N most interesting itemsets for every length l, which is different from our work in several aspects:

- They mine all the itemsets instead of only the closed ones, and mining closed itemsets is not only more desirable but also more challenging;
- They do not have minimum length constraints—since it mines itemsets at all the lengths, some heuristics developed here cannot be applied, and
- Their philosophy and methodology of FP-tree modification are also different from ours.

To the best of our knowledge, this is the first study on mining top-k frequent closed itemsets with length
constraint, therefore, we only compare our method with two well-known efficient closed itemset mining algorithms. From the user-interaction point of view, since our performance study shows that there is no real need to specify min_l if one wants to mine frequent closed itemsets of any length, and there is no crucial need to specify k for top-k mining as long as k is a default number that fits user’s expectation or application requirements, this method gives the user the minimal burden to specify mining parameters, representing a step toward parameter-free frequent-pattern mining.

There are extensive studies on mining frequent itemsets from many different angles, such as constraint-based mining [6], [4], [19], mining generalized and quantitative rules [2], [13], and mining correlation rules. Our study on mining top-k frequent itemsets is orthogonal to these studies. Since their mining and optimization frameworks are based on a predefined min_sup support threshold, the techniques developed in this study can be extended to the scope of these studies to improve their corresponding algorithms for mining top-k frequent itemsets. We also expect that the basic principles developed here can be applied to recently developed new frequent itemset mining algorithms, such as when the requirement is changed to mining top-k frequent itemsets. Finally, it is expected that the philosophy developed here will influence the mining of top-k frequent structured patterns, where a structured pattern may contain sequences, trees, lattices, and graphs.

There are various methods for mining high utility itemsets. Mining high utility itemsets has four main methods used for finding frequent itemsets. These leave wide rooms for exploration as future work. Since their mining and optimization frameworks are based on a predefined min_sup support threshold, the techniques developed in this study can be extended to the scope of these studies to improve their corresponding algorithms for mining top-k frequent itemsets. We also expect that the basic principles developed here can be applied to recently developed new frequent itemset mining algorithms, such as when the requirement is changed to mining top-k frequent itemsets. Finally, it is expected that the philosophy developed here will influence the mining of top-k frequent structured patterns, where a structured pattern may contain sequences, trees, lattices, and graphs.

Data Structure

Data Structure is nothing but organizing the data so that we can use that data efficiently. Mining high utility itemsets requires a technique to improve their corresponding algorithms for mining top-k frequent itemsets. We also expect that the basic principles developed here can be applied to recently developed new frequent itemset mining algorithms, such as when the requirement is changed to mining top-k frequent itemsets. Finally, it is expected that the philosophy developed here will influence the mining of top-k frequent structured patterns, where a structured pattern may contain sequences, trees, lattices, and graphs.

UP-Growth Mining Method

In the first step we get the global UP tree that is mining UP-Tree by FP-Growth. Which can be used for generating PHUIs will generate so many candidates in order to avoid that UP-Growth method is used with two techniques mainly: First one is discarding unpromising items during constructing a local UP-Tree and second is discarding local node utilities.

An Improved Mining Method: UP-Growth+

As compared with UP-Growth FP-Growth gives the better performance. FP growth is used to find the frequent itemsets. FP-Growth uses DLU and DLN to decrease the overhead utilities of itemsets. However, the overestimated utilities can be closer to their actual utilities by eliminating the estimated utilities that are closer to actual utilities of unpromising items and descendant nodes. In this section, we propose an improved method, named UP-Growth+, for reducing overestimated utilities more effectively. In UP-Growth, minimum item utility table is used to reduce the overestimated utilities. In UP-Growth+, minimal node utilities in each path are used to make the estimated pruning values closer to real utility values of the pruned items in database.

Efficiently Identify High Utility Itemsets

After finding all PHUIs, the third step is to identify high utility itemsets and their utilities from the set of PHUIs by scanning original database once [3], [11]. However, in previous studies, two problems in this phase occur: 1) number of HTWUIs is too large; and 2) scanning original database is very time consuming. In our framework, overestimated utilities of PHUIs are smaller than or equal to TWUs of HTWUIs since they are reduced by the proposed strategies. Thus, the number of PHUIs is much smaller than that of HTWUIs. Therefore, in phase II, our method is much more efficient than the previous methods. Moreover, although our methods generate fewer candidates.

REFERENCES


