An Efficient Approach for Mining Potential High Utility Itemsets from Dubious Database

C. Niveditha and Mr. R. Karthikeyan

Received 28 January 2017; Accepted 22 May 2017; Available online 28 May 2017

ABSTRACT

Mining High utility itemsets from precise databases and from uncertain databases is one of the major issues in market basket analysis. Comparing with precise databases, uncertain databases incur the problem in generating more number of potential itemsets which varies from time to time along with their probability value. The term potential high utility itemsets refers to the mining results based on user interest like profit, quantity. Mining potential itemsets greater than minimum threshold value and their minimum probability value in uncertain databases degrades in mining results and efficiency. The aim is to improve mining results and their speed of extracting useful information. The objective of streaming in mining is to extract useful information with less memory consumption. In an existing system, the high utility itemsets are extracted using PHUI-UP(Potential High Utility Itemsets Upper Bound) and PHUI-List(potential high utility itemsets PU-list-based mining algorithm) without using streaming concept. Even though, it extracts the high utility itemsets it degrades in memory consumption and finding important itemsets. In proposed system, the continuous interval of batches (transactions) is reduced to discrete intervals. The items in the transactions are splitted based on the base interval using ExactU2Stream using sliding window technique and for mining potential high utility itemsets an algorithm named PHUI-UP and PHUI-List is used. By using this algorithm, PHUI are mined without generating candidate itemsets and use probability utility list structure (PU-List) to improve scalability of mining important frequent itemsets. The user defined minimum threshold values are changed according to user convenient to mine the promising itemsets from uncertain databases effectively.

KEYWORDS: high utility item sets, data streams, utility mining, probability utility list structure, minimum threshold value, uncertain database.

INTRODUCTION

Data Mining manages the way towards extricating the required data from any sort of database. It decreases manual pursuit done by human. It is helpful from multiple points of view. It gives better answer for some ongoing issues which includes with basic leadership [1].

Objectives and necessities from business perspective change the data into Data mining issue definition brings about preliminary course of action. The data understanding stage starts with a fundamental data assembling and proceeds with activities to get settled with the data, to recognize data quality issues, to discover first bits of learning into the data, or to distinguish captivating subsets to casing hypotheses for disguised information [1][5].The information readiness stage of data availability organize covers all activities to build up the last informational index (data that will be supported into the showing tool(s)) from the basic rough data. Generation of the model is for the paying little mind to the likelihood that the inspiration driving the model is to fabricate learning of the data, the learning got ought to be created and shown in a way that the client can be used.
Frequent pattern mining in market basket analysis such as online shopping and kind of internet based applications work.

The Rule calculations are Apriori, Frequent Pattern Growth. Apriori creates more number of applicant itemset. To beat this, continuous example development appeared. Visit design development comes about just two sweeps of database which additionally brings about creating hopeful itemsets yet a great deal less when contrasted with Apriori.[2]

Utility mining appears of mining incessant example in view of client intrigue. Data mining is an interdisciplinary subfield of Personal Computer Science. It is the computational method of discovering cases in significant data sets including procedures at the meeting of fake awareness, machine learning, estimations, and database frameworks. The target of mining is to notice information from data set and changes it into a reasonable structure for further utilize [1,2,3]. Beside the essential stride, it incorporates database and data from administrator points of view, data pre-handling, model and derivation thoughts, charming quality estimations, capriciousness examinations, post-get ready of discovered structures, recognition, and web based upgrading. Information mining is the examination wander of the "learning divulgence in databases" process, or Knowledge Discovery Process (KDD) [4,5].

Data Stream Mining is the way toward extricating learning structures from ceaseless, quick information records. A Data stream is a requested arrangement of examples that in numerous utilizations of information stream mining can be perused just once or a little number of times utilizing constrained processing and capacity abilities. Cases of information streams incorporate PC organize activity, telephone discussions, ATM exchanges, web quests, and sensor information. Information stream mining can be viewed as a subfield of information mining, machine learning, and learning disclosure. In numerous information stream mining applications, the objective is to anticipate the class or estimation of new occasions in the information stream gives us some learning about the class enrollment or estimations of past cases in the information stream. Machine learning systems can be utilized to take in this forecast errand from marked cases in a computerized form. Frequently, ideas from the field of incremental taking in, a speculation of Incremental heuristic inquiry are connected to adapt to basic changes, on-line learning and constant requests. In numerous applications, particularly working inside non-stationary situations, the appropriation hides the occasions or the standards basic their naming may change after some time, i.e. the objective of the expectation, the class to be anticipated or the objective incentive to be anticipated, may change after some time [6].

Related Work:

Jerry Chun-Wei Lin et al (2016) illustrated about High-utility item set mining (HUIM) is a useful set of techniques for discovering patterns in transaction databases, which considers both quantity and profit of items[18]. However, most algorithms for mining high-utility item sets (HUIs) assume that the information stored in databases is precise, i.e., that there is no uncertainty. But in many real-life applications, an item or item set is not only present or absent in transactions but is also associated with an existence probability.

This is especially the case for data collected experimentally or using noisy sensors. In the past, many algorithms were respectively proposed to effectively mine frequent item sets in uncertain databases.

But mining HUIs in an uncertain database has not yet been proposed, although uncertainty is commonly seen in real-world applications. In this paper, a novel framework, named potential high-utility item set mining (PHUIM) in uncertain databases, is proposed to efficiently discover not only the item sets with high utilities but also the item sets with high existence probabilities in an uncertain database based on the tuple uncertainty model [18]. The PHUI-UP algorithm (potential high-utility item sets upper-bound-based mining algorithm) is first presented to mine potential high-utility item sets (PHUIs) using a level-wise search. Since PHUI-UP adopts a generate-and-test approach to mine PHUIs, it suffers from the problem of repeatedly scanning the database. To address this issue, a second algorithm named PHUI-List (potential high-utility item sets PU-list-based mining algorithm) is also proposed. It directly mines PHUIs without generating candidates, thanks to a novel probability-utility-list (PU-list) structure, thus greatly improving the scalability of PHUI mining. Substantial experiments were conducted on both real-life and synthetic datasets to assess the performance of the two designed algorithms in terms of runtime, number of patterns, memory consumption, and scalability.

Liang Wang et al (2016) proposed how the data handled in emerging applications like location-based services, sensor monitoring systems, and data integration, are often inexact in nature.[19] In this paper, the important problem of extracting frequent item sets from a large uncertain database, interpreted under the Possible World Semantics. This issue is technically challenging, since an uncertain database contains an exponential number of possible worlds. By observing that the mining process can be modeled as a Poisson binomial distribution, developed an approximate algorithm, which can efficiently and accurately discover frequent item sets in a large uncertain database. The important issue of maintaining the mining result for a database that is evolving (e.g., by inserting a tuple). Specifically, proposed incremental mining algorithms, which enable probabilistic frequent itemset results to be, refreshed [19]. Reduces the need of re-executing the whole mining algorithm on the new database, which is often more expensive and unnecessary. Existing
algorithm that extracts exact item sets, as well as approximate algorithm, can support incremental mining. All approaches support both tuple and attribute uncertainty, which are two common uncertain database models. Extensive evaluations on real and synthetic datasets are validated.

Jerry Chun-Wei Lin et al (2015) have showed that High-utility pattern mining (HUPM) is an emerging topic in recent years instead of association-rule mining to discover more interesting and useful information for decision making. Many algorithms have been developed to find high-utility patterns (HUPs) from quantitative databases without considering timestamp of patterns, especially in recent intervals [17]. A pattern may not be a HUP in an entire database but may be a HUP in recent intervals. In this paper, a new concept namely up-to-date high-utility pattern (UDHUP) is designed. It considers not only utility measure but also timestamp factor to discover the recent HUPs. The UDHUP-apriori is first proposed to mine UDHUPs in a level-wise way. Since UDHUP-apriori uses Apriori-like approach to recursively derive UDHUPs, a second UDHUP-list algorithm is then presented to efficiently discover UDHUPs based on the developed UDU-list structures and a pruning strategy without candidate generation, thus speeding up the mining process. A flexible minimum-length strategy with two specific lifetimes is also designed to find more efficient UDHUPs based on a users’ specification. Experiments are conducted to evaluate the performance of the proposed two algorithms in terms of execution time, memory consumption, and number of generated UDHUPs in several real-world and synthetic datasets.

Morteza Zihayat et al (2014) illustrated that online high utility item set mining over data streams has been studied recently [20]. However, the existing methods are not designed for producing top-k patterns. Since there could be a large number of high utility patterns, finding only top-k patterns is more attractive than producing all the patterns whose utility is above a threshold. A challenge with finding top-k high utility item sets over data streams is that it is not easy for users to determine a proper minimum utility threshold in order for the method to work efficiently. In this paper, a new method (named T-HUDS) for finding top-k high utility patterns over sliding windows of a data stream is proposed. The method is based on a compressed tree structure, called HUDS-tree that can be used to efficiently find potential top-k high utility item sets over sliding windows. T-HUDS uses a new utility estimation model to more effectively prune the search space. Several strategies for initializing and dynamically adjusting the minimum utility threshold are used [20]. No top-k high utility item set is missed by the proposed method. Experimental results on real and synthetic datasets show that strategies and new utility estimation model work very effectively and that T-HUDS outperforms two state-of-the-art high utility item set algorithms substantially in terms of execution time and memory storage.

P. Fournier-viger et al (2014) describes high utility item set mining is a challenging task in frequent pattern mining, which has wide applications. The state-of-the-art algorithm is HUI-Miner [19]. It adopts a vertical representation and performs a depth-first search to discover patterns and calculate their utility without performing costly database scans. Although, this approach is effective, mining high-utility item sets remains computationally expensive because HUI-Miner has to perform a costly join operation for each pattern that is generated by its search procedure. In this paper, issue is addressed by proposing a novel strategy based on the analysis of item co-occurrences to reduce the number of join operations that need to be performed.

V.S. Tseng et al (2013) proposed two algorithms, namely utility pattern growth (UP-Growth) and UP-Growth+, for mining high utility item sets with a set of effective strategies for pruning candidate item sets [22]. The information of high utility item sets is maintained in a tree-based data structure named utility pattern tree (UP-Tree) such that candidate item sets can be generated efficiently with only two scans of database. Two novel algorithms as well as a compact data structure for efficiently discovering high utility item sets from transactional databases is proposed. 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 item sets and maintaining important information related to utility patterns within databases are proposed. High utility item sets can be generated from UP-Tree efficiently with only two scans of original databases. Several strategies are proposed for facilitating the mining processes of UP-Growth and UP-Growth+ by maintaining only essential information in UP-Tree [22]. 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. 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.

L. Wang et al (2012) studied the important problem of extracting frequent item sets from a large uncertain database, interpreted under the Possible World Semantics (PWS) [19,20]. This issue is technically challenging, since an uncertain database contains an exponential number of possible worlds. By observing that the mining process can be modeled as a Poisson binomial distribution, developed an approximate algorithm, which can efficiently and accurately discover frequent item sets in a large uncertain database. The important issue of maintaining the mining result for a database that is evolving (e.g., by inserting a tuple). Specifically, Incremental mining algorithms, which enable Probabilistic Frequent Itemset (PFI) results to be refreshed. This reduces the
need of re-executing the whole mining algorithm on the new database, which is often more expensive and unnecessary.

C.W. Wu et al (2012) addressed all of the above challenges by proposing an efficient algorithm named TKU for Top-K Utility item set mining [21]. A novel framework for mining top-k high utility item sets has been considered. An algorithm named TKU is proposed for efficiently mining the complete set of top-k high utility item sets in the database without specifying min_util threshold. Five new strategies are proposed for effectively raising the threshold at different stage of the mining process. The first four strategies effectively raise the threshold during the mining process to prune the search space and reduce the number of candidates. The last strategy effectively reduces the number of candidates that need to be checked in.

Preliminaries:

Definition 1. A transaction comprises one or more non-repeated attributes. An attribute is associated with an interval and a probability density function, which assigns a probability to each value in the interval.

Definition 2. Suppose an attribute A in a transaction T is associated with a quantitative interval IA and a probability density function PA. The existential probability of an interval IAS ∈ IA is the possibility that the values in IAS appear in T. This is defined as the integral of the density over IAS, denoted by ExProb(IAS,T).

Definition 3. A quantitative interval I of attribute A, with a range from m to n is represented by [m,n]. A U2 pattern comprises one or more non-repeated attributes, each of which is associated with an interval; a U2 pattern comprised of interval I1,I2,...,In is denoted as [I1,I2,...,In]. A U2 pattern is frequent if its expected support (see Definition 4) exceeds the minimum support specified by the user. Thus, a frequent U2 pattern represents the intervals where the actual values locate with high probability.

Definition 4. For a given U2 pattern Pat, the expected support, i.e., the expected number of transactions that contain Pat in the database, is denoted as ExSupport(Pat). Let TPat be the set of transactions containing Pat, and let an interval x of a transaction TInPat ∈ TPat correspond to one of the intervals in Pat. Thus, ExSupport(Pat) = |TPat | i=1 x∈TPatIA in Pat ExProb(x,TInPat) (1) where |TPat| denotes the number of transactions in TPat.

Definition 5. The utility of an item ij in a transaction Tq is defined as u(ij, Tq) = q(ij, Tq) × pr(ij).

Definition 6. The utility of an itemset X occurring in Tq is denoted as u(X, Tq), which can be defined as p(X, Tq) = p(Tq), where p(Tq) is the corresponding probability of Tq.

Definition 7. The utility of an itemset X in a transaction Tq is denoted as u(X, Tq), and defined as u(X, Tq) = i,j∈X u(ij, Tq) (2).

Definition 8. The utility of an itemset X in D is denoted as u(X), and is defined as u(X) = X⊆D u(X, Tq).

Definition 9. The potential probability of an itemset X in D is denoted as Pro(X), and defined as Pro(X) = X⊆D Tq∈D p(X, Tq).

Definition 10. The transaction utility of transaction Tq is denoted as tu(Tq), and defined as tu(Tq) = m j=1 u(ij, Tq) (3), where m is the number of items in Tq.

Definition 11. The total utility in D is the sum of all transaction utilities in D and is denoted as TU, which can be defined as TU = Tq∈D tu(Tq).

Definition 12. (Downward closure property of high probability itemsets). If an itemset is a high probability itemset in an uncertain database, then this itemset respects the downward closure property in this database.

Proposed System For Mining Potential High Utility Itemsets:

Mining continuous example from univariate questionable information streams, which have a quantitative interim for each characteristic in an exchange and a likelihood thickness work showing the conceivable outcomes that the qualities in the interim show up. Numerous information streams involve streams of univariate dubious information. Mining information streams is one of the significant subjects in the information mining territory; there exists numerous critical themes, for example, mining continuous examples from information.
streams and information stream arrangement. Reason to progress in equipment and programming systems is to, vast volumes of information streams are recorded, for example, the snap floods of web logs, call records in a call focus, and every day retail exchanges. Furthermore, the direction way of a moving item, e.g., cell phone, can likewise be dealt with as a constant information stream. As opposed to a static dataset, an information stream has three time-related attributes. Information stream is a ceaseless and unbounded information stream. The substance of an information stream changes after some time. Restricted estimate blunders are accessible. The third property is required on the grounds that just recover inexact outcomes, since the aggregate sum of every single past data is very substantial to prepare. An essential and developing theme in the regular example mining territory is mining unverifiable information. As of now, most existing reviews manage exact information, where things are either present in, or truant from, an exchange (Boolean information mining), or where each quality of an exchange is related with a quantitative esteem (quantitative information mining). As opposed to exact information, unverifiable information does not record the exact estimation of a quality or the presence of a thing. There are three sorts of dubious information in the writing: univariate unverifiable information, thing set indeterminate information, and tuple questionable information. Univariate questionable information has a quantitative interim for each property in an exchange, which is joined by a likelihood thickness work demonstrating the potential outcomes that the qualities in the interim show up. At that point, a likelihood thickness capacity is expressly or certainly relegated to the interim to show the likelihood that each esteemed results exists in the quantitative interim.

Workflow Illustration:

![Workflow Illustration](image)

**Fig. 1: System Architecture**

The proposed solution for mining potential high utility item sets effectively is described as follows.

Mining High utility itemsets from exact databases and from dubious databases is one of the real issues in market crate investigation.

Contrasting and exact databases, dubious databases acquire the issue in creating more number of potential itemsets which differs from likelihood esteem. The term Potential high utility itemsets (utility) alludes to the mining comes about in light of client intrigue like benefit, amount. Mining potential itemsets is more prominent than least edge esteem and their base likelihood esteem in indeterminate databases corrupts in mining results and proficiency [17].

The point is to enhance mining comes about and their speed of separating valuable data. The target of spilling in mining is to remove valuable data with less memory utilization. In existing framework, the high utility item sets are removed utilizing PHUI-UP and PHUI-List without utilizing spilling idea. Despite the fact that, it removes the high utility item sets it corrupts in memory utilization and finding vital item sets.). In proposed framework, the items in the Transactions are splitted in light of the base interim utilizing ExactU2Stream utilizing sliding window method. Mining potential high utility item sets a Calculation named PHUI-UP (Potential High Utility Item sets Upper Bound) and PHUI-List (potential high utility item sets PU-list-based mining calculation) is utilized. Additionally utilize likelihood utility rundown structure (PU-List) to enhance adaptability of mining critical successive item sets[18] is explained in Fig.1.

5.1. **U2TransTree:**

Once the items are purchased by customers the base interval is initialized. Based on the base interval, the frequent item sets are found from the database. The streaming concept is used here to split the data. For every base interval, the frequent item sets are considered. The sliding window protocol technique will be used. This
protocol will bring the frequent items from previous intervals. So the previous information is extracted. From this information, the U2P tree construction is constructed. The frequent item sets are drawn and implemented in tree structure. This streaming concept is done for effective mining. Along with the frequent mining the probability value for that item is also extracted. Minimum threshold value will be set according to their interest. The item sets are extracted above the threshold value. The algorithm of U2PTransTree is shown below in Fig. 2.

Input: Uncertain Database, Minimum support threshold (s)
Output: U2P transtree
Let the list of base intervals LBI be null
Create the root node of U2PTransTree UTT
for each incoming batch IB (or incoming time unit TU) do
  for each transaction Tran in IB (or TU) do
    for each interval Int formed by two consecutive bound values in A do
      update LBI by inserting Int into LBI
      end for
    end for
  end for
  Update Tran and UTT according to LBI
  Insert Tran into UTT
  end for
  Remove the obsolete nodes when the window slides forward
end for
Update U2P transtree

Fig. 2: U2PTransTree.

5.2. Potential High Utility Itemset-Upperbound:

From the high utility item sets from U2P TransTree construction, the PHUI-UP algorithm (potential high-utility item sets upper-bound-based mining algorithm) is presented to mine potential high-utility item sets (PHUIs) using a level-wise search. In level wise approach, calculate Utility, Transaction utility, Transaction Weighted Utility, High Transaction Weighted Utility. In this stage, set a user defined minimum Threshold value to extract the useful information from the streaming part. These are considered as promising item. The algorithm of PHUI-Upper Bound is shown below in Fig. 3.

Scan D to calculate TU and obtain TWU (ij) and Pro (ij) for each ij ∈ D.
For each ij ∈ D do
  If TWU (ij) ≥ TU × ε ∧ Pro (ij) ≥ |D| × μ then
    HTWPUI1 ← HTWPUI1∪ij;
  End if
  End for
set k ← 2;
While HTWPUIk-1:
  Ck = Apriori_gen (HTWPUIk-1);
  For each k-itemset X ∈ Ck do
    ∗ scan D once for all k-itemset ∗
    Calculate TWU(X) and Pro(X);
  End if
End for
K ← k+1;
End while
HTWPUIs← HTWPUIs∪HTWPUIk;
For each k-itemset X ∈ HTWPUIs do
  ∗ scan D for all candidates ∗
  Calculate the actual utility for X: = u(X);
  If u(X) ≥ TU × ε then
    PHUIs ← PHUIs∪X;
  End if
End for
Return PHUIs; Algorithm

Fig. 3: PHUI Upper-Bound.
5.3. PHUI PU-List Algorithm:

With the help of promising items, the potential high utility item sets are extracted. PHUI-List (potential high-utility item sets PU-list-based mining algorithm) is also used. This latter directly mines PHUIs without generating candidates, probability-utility-list (PU-list) structure, thus greatly improving the scalability of PHUI mining. Now, potential important frequent item sets are extracted along with their probabilities. The Probability Utility algorithm is shown below in Fig. 4.

Scan D to calculate TU and the TWU (i) and Pro (i) value of each item i∈I;
Find I* ← {i∈I | TWU (i) ≥ TU × ε ˄ Pro (i) ≥ |D| × μ};
Sort I* in TWU ascending order;
Scan D once to build the PU-list of each 1-item i∈I*;
Call PHUI-Search(∅, I*, ε, μ);
Return PHUIlist.

Fig. 4: Probability Utility-List

RESULTS AND DISCUSSION

In this section the proposed and existing methods for mining high utility itemset is evaluated in terms of running time, pattern analysis and memory consumption. For the experimental purpose retail dataset is used. Retail dataset is a real life dataset in SPMF format. The retail dataset contain anonymous retail market basket data from an anonymous Belgian retail store. Retail dataset is sparse type of dataset.

A. Running time:

Running time is defined as the amount of time taken by the PHUI-UP, PHUI-List and ExactU2Stream algorithm. It can likewise be observed that the PHUI-List calculation beats the PHUI-UP calculation under different MPs. The runtime of PHUI-UP was diminished pointedly as the MP was expanded, while the runtime of the PHUI-List calculation diminished consistently. This outcome is sensible since when the MP is set higher, less competitors are produced by the PHUI-UP calculation to mine the PHUIs. In spite of the fact that the PHUI-UP calculation utilizes the TWPUDC property to lessen the inquiry space, regardless it receives a level-wise approach and can create a gigantic measure of contender for mining PHUIs. In addition, just a little rate of these hopefuls is PHUIs. In this manner, when the MP is set higher, numerous excess unpromising applicants are pruned early, and therefore the inquiry space and runtime of the PHUI-UP calculation diminishes strongly. Nearly When ExactU2Streaming calculation is utilized, PHUI are mined adequately in view of client’s advantage.

Fig. 5: Comparison of Running Time
It is proved that the ExactU2Stream algorithm has low running time than the other two algorithms. The comparison of runtime is shown in Fig. 5. The proposed algorithm is generally up to almost one or two orders of magnitude faster than PHUI-UP and PU-list and is always faster than the existing results.

**B. Pattern analysis:**

In this section the patterns obtained by the PHUIs and ExactU2Stream methods are analyzed for different minimum potential probability threshold value. No calculation had been beforehand produced for finding PHUIs in questionable datasets. In an unverifiable dataset, when the likelihood of every exchange is set to 1.0, it implies that every exchange has a 100% existential likelihood, and the questionable dataset turns into an exact quantitative dataset. All things considered, PHUI mining calculations deliver an indistinguishable yield from conventional high-utility itemset mining calculations that is the entire arrangement of HUIs. This outcome is contrasted with the planned PHUI-UP and PHUI-List calculations for mining PHUIs. It is proved that the ExactU2Stream algorithm has better pattern analysis than the other algorithm. The comparison of pattern analysis is shown in Fig. 6.

**C. Memory consumption:**

Memory consumption is defined the amount of memory consumed by the PHUI-UP, PHUI-List and ExactU2Stream algorithm. The PHUI-UP calculation requires significantly more memory when the MU or MP is set lower, not at all like the PHUI-List calculation. For instance, PHUI-UP requires 2,100 MB of memory by and large, yet PHUI-List just requires around 225 MB of memory all things considered. The reason is that the PHUI-List calculation uses two methodologies to prune unpromising itemset ahead of schedule, without building their PU-records. It can likewise be watched that the execution crevice between PHUI-UP and PHUI-List is littler when MU is expanded and MP is settled or when MP is expanded and MU is settled. The proposed work will diminish the memory utilization similarly to the current calculation. It is proved that the ExactU2Stream algorithm has better memory consumption than the other two algorithms. The comparison of memory consumption is shown in Fig. 7.
Summary:
The utility mining from uncertain databases allows the owner to find the frequent itemsets based on their own interest which improves the efficiency of mining results in market basket analysis while streaming the uncertain database before extracting high utility itemsets. This will not generate more number of candidate generations since the streaming part comes before mining which will not degrade mining performance.

REFERENCES