Volume 5, Issue 12, December 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Literature Review on a Unified Approach for Mining in Lossless Representaion of Closed Itemsets

Sonal D. Tamaskar *
Student of Master of Engineering in (CS & IT)
HVPM's College of Engineering and Technology,
Amravati, Maharashtra, India

Prof. Anjali B. Raut
Associate Professor and Head of the Department of (CSE)
HVPM's College of Engineering and Technology,
Amravati, Maharashtra, India

Abstract*— Data mining, often called as Knowledge discovery in databases (KDD), aims at the discovery of useful information from large collections of data. To achieve high efficiency for the mining task and provide a concise mining result to users, we propose a novel framework in this paper for mining closed+ high utility itemsets (CHUIs), which serves as a compact and lossless representation of HUIs. We propose three efficient algorithms named AprioriCH (Apriori-based algorithm for mining High utility Closed + itemsets), AprioriHC-D (AprioriHC algorithm with Discarding unpromising and isolated items) and CHUD (Closed+ High Utility Itemset Discovery) to find this representation. Further, a method called DAHU (Derive All High Utility Itemsets) is proposed to recover all HUIs from the set of CHUIs without accessing the original database.

Keywords*— Frequent itemset, closed+ high utility itemset, lossless and concise representation, utility mining, data mining.

I. INTRODUCTION

Data Mining has wide applications in many areas such as banking, medicine, scientific research and among government agencies. Classification is one of the commonly used tasks in data mining applications. Example- Personal Health Record (PHR) service is an emerging model for health information exchange. It allows patients to create, update and manage personal and medical information. Also they can control and share their medical information with other users as well as health care providers.

Frequent Itemset mining (FIM) [1], [2], is a fundamental research topic in data mining. One of its popular applications is market basket analysis, which refers to the discovery of sets of items (itemsets) that are frequently purchased together by customers. Removing redundant itemsets produces a condensed representation of all frequent itemsets. The best known condensed representations are closed itemsets (or generators/free sets) and non-derivable itemsets. The collection of non-derivable itemsets often is smaller than the collection of closed sets, but given an arbitrary dataset either one may contain fewer itemsets. However, in this application, the traditional model of FIM may discover a large amount of frequent but low revenue itemsets and lose the information on valuable itemsets having low selling frequencies. These problems are caused by the facts that (1)FIM treats all items as having the same importance/unit profit/weight and (2) it assumes that every item in a transaction appears in a binary form, i.e., an item can be either present or absent in a transaction, which does not indicate its purchase quantity in the transaction. Hence, FIM cannot satisfy the requirement of users who desire to discover itemsets with high utilities such as high profits. To address these issues, utility mining [2], [3], [4], [5], [7], emerges as an important topic in data mining.

In utility mining, each item has a weight (e.g. unit profit) and can appear more than once in each transaction (e.g. purchase quantity). The utility of an itemset represents its importance, which can be measured in terms of weight, profit, cost, quantity or other information depending on the user preference. An itemset is called a high utility itemset (HUI) if its utility is no less than a user-specified minimum utility threshold; otherwise, it is called a low utility itemset. Utility mining is an important task and has a wide range of applications such as website click stream analysis [2], cross marketing in retail stores [7] mobile commerce environment and biomedical applications [3]

II. LITERATURE REVIEW

Several researchers have done the research in many areas:

R. Agrawal et al in [1] proposed Apriori algorithm, it is used to obtain frequent itemsets from the database. In miming the association rules we have the problem to generate all association rules that have support and confidence greater than the user specified minimum support and minimum confidence respectively. The first pass of the algorithm simply counts item occurrences to determine the large 1-itemsets. First it generates the candidate sequences and then it chooses the large sequences from the candidate ones. Next, the database is scanned and the support of candidates is counted. The second step involves generating association rules from frequent itemsets. Candidate itemsets are stored in a hash-tree. The hash tree node contains either a list of itemsets or a hash table. Apriori is a classic algorithm for frequent itemset mining and association rule learning over transactional databases. After identifying the large itemsets, only those itemsets are allowed which have the support greater than the minimum support allowed. Apriori Algorithm generates lot of

candidate item sets and scans database every time. When a new transaction is added to the database then it should rescan the entire database again. Here introduce three efficient algorithms AprioriHC (An Apriori-based algorithm for mining High utility Closed+ itemsets), AprioriHC-D (AprioriHC algorithm with Discarding unpromising and isolated items) and CHUD (Closed+ High Utility itemset Discovery) for mining CHUIs. They rely on the TWU-Model [2], [5], [7] and include strategies to improve their performance. All algorithms consist of two phases named Phase I and Phase II. In Phase I, potential closed high utility itemsets (PCHUIs) are found, which are defined as a set of itemsets having an estimated utility (e.g. TWU) no less than abs_min_utility. In Phase II, by scanning the database once, CHUIs are identified from the set of PCHUIs found in Phase I and their utility unit arrays are computed. The AprioriHC and AprioriHC-D are based on Apriori [1] and the Two-Phase [5] algorithms. They use a horizontal database and explore the search space of CHUIs in a breadth-first search. The algorithm AprioriHC is regarded as a baseline algorithm in this work and AprioriHC-D is an improved version of AprioriHC. On the other hand, the proposed algorithm CHUD is an extension of Eclat [8] and DCI-Closed [6] algorithms. The CHUD algorithm considers vertical database and mines CHUIs in a depth-first search.
In the following, we present algorithms.

The CHUD Algorithm

In this section, we present an efficient depth-first search algorithm named CHUD (Closed+ High Utility itemset Discovery) to discover CHUIs. CHUD is an extension of DCIClosed [6], one of the currently best methods to mine closed itemsets. CHUD is adapted for mining CHUIs and include several effective strategies for reducing the number of candidates generated in Phase I. Similar to the DCI-Closed algorithm, CHUD adopts an Itemset-Tidset pair Tree (IT-Tree) [6], [8] to find CHUIs. In an IT-Tree, each node N(X) consists of an itemset X, its Tidset g(X), and two ordered sets of items named PREV-SET(X) and POST-SET(X). The IT-Tree is recursively explored by the CHUD algorithm until all closed itemsets that are HTWUIs are generated.

PROCEDURE: *AprioriHC-D_Phase*-II

Input: D_I: the database containing no unpromising items;
pCHUI: set of PCHUIs; *abs_min_utility*
Output: the complete set of CHUIs
01. ***for*** ($k := 1$; $L_k \neq \emptyset$; k++)
02. { $L_k := k$-itemsets in *pCHUI*
03. ***for all*** X in L_k ***do***
04. { ***Calculate*** $au(X)$ and utility unit array of X ***from*** D_k
05. ***If*** $au(X) \geq abs\_min\_utility$ ***then*** { ***output*** X }
06. }
07. $D_{k+1} :=$ ***IIDS_Strategy***(D_k, L_k)
08. }

Fig. 1. AprioriHC-D_Phase-II procedure.

The main procedure of CHUD is named Main and is shown in Fig. 6. It takes as parameter a database D and the abs_min_utility threshold. CHUD first scans D once to convert D into a vertical database. At the same time, CHUD computes the transaction utility for each transaction TR and calculates TWU of items. When a transaction is retrieved, its Tid and transaction utility are loaded into a global TU-Table named GTU. As previously defined, an item is called a promising item if its estimated utility (e.g. its TWU) is no less than abs_min_utility. After the database scan, promising items (cf. Definition 17) are collected into an ordered list O = <a1; a2; . . . ; an>, sorted according to a fixed order _ such as increasing order of support. Only promising items are kept in O since supersets of unpromising items are not CHUIs. According to [7], the utilities of unpromising items can be removed from the GTU table. This step is performed at line 2 of the Main procedure. Then, CHUD generates candidates in a recursive manner, starting from candidates containing a single promising item and recursively joining items to them to form larger candidates. To do so, CHUD takes advantage of the fact that by using the total order $\prec$, the complete set of itemsets can be divided into n non-overlapping subspaces, where the kth subspace is the set of itemsets containing the item ak but no item $a_i \prec a_k$ [6]. For each item $a_k \in O$, CHUD creates a node $N(\{a_k\})$ and puts items a1 to ak-1 into PREV-SET({ak}) and items ak+1 to an into POST-SET({ak}). Then CHUD calls the CHUD Phase-I procedure for each node $N(\{a_k\})$ to produce all the candidates containing the item ak but no item $a_i \prec a_k$. After that, the REG strategy is applied by calling the REG_Strategy sub-function, which will be described later. Finally, the Main procedure performs Phase II on these candidates to obtain all CHUIs

ALGORITHM: CHUD

Input: *D*: the database; *abs_min_utility*
Output: complete set of CHUIs
01. ***InitialDatabaseScan***(*D*)
02. ***RemoveUtilityUnpromisingItems***(*O*, *GTU*)
03. ***for each*** item $a_k \in O$ ***do***
04. { Create node $N(\{a_k\})$
05. ***CHUD_Phase-I***($N(\{a_k\})$, *GTU*, *abs_min_utility*)
06. ***REG_Strategy***($g(a_k)$, *GTU*)}
07. ***CHUD_Phase-II***(*D*, *abs_min_utility*)

Fig. 2. CHUD algorithm.

III. CONCLUSION

In this paper, we discussed basic concepts like data mining, frequent itemset mining and mining high utility itemsets. The main objective of utility mining is to identify the itemsets with highest utilities, by considering profit, quantity, cost or other user preferences. We have presented a novel algorithm for high-utility itemset mining. AprioriHCD perform a breadth-first search for mining closed+ high utility itemsets from horizontal database, while CHUD performs a depth-first search for mining closed+ high utility itemsets from vertical database. The strategies incorporated in CHUD are efficient and novel.

REFERENCES

[1] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in Proc. 20th Int. Conf. Very Large Data Bases, 1994,pp. 487–499.

[2] C. F. Ahmed, S. K. Tanbeer, B.-S. Jeong, and Y.-K. 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.

[3]] R. Chan, Q. Yang, and Y. Shen, "Mining high utility itemsets," in Proc. IEEE Int. Conf. Data Min., 2003, pp. 19–26.

[4] A. Erwin, R. P. Gopalan, and N. R. Achuthan, "Efficient mining of high utility itemsets from large datasets," in Proc. Int. Conf. Pacific- Asia Conf. Knowl. Discovery Data Mining, 2008, pp. 554–561

[5] Y. Liu, W. Liao, and A. Choudhary, "A fast high utility itemsets mining algorithm," in Proc. Utility-Based Data Mining Workshop, 2005, pp. 90–99.

[6] C. Lucchese, S. Orlando, and R. Perego, "Fast and memory efficient mining of frequent closed itemsets," IEEE Trans. Knowl. Data Eng., vol. 18, no. 1, pp. 21–36, Jan. 2006.

[7] V. S. Tseng, C.-W. Wu, B.-E. Shie, and P. S. Yu, "UP-Growth: An efficient algorithm for high utility itemset mining," in Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2010, pp. 253–262.

[8] M. J. Zaki and C. J. Hsiao, "Efficient algorithms for mining closed itemsets and their lattice structure," in IEEE Trans. Knowl. Data Eng., vol. 17, no. 4, pp. 462–478, Apr. 2005.

