Approach for Mining in Lossless Representation of Closed Itemsets

Sonal D. Tamaskar  
M.E. Student  
Department of Computer Science & Engineering Information Technology  
HVPM’s college of Engineering and Technology

Prof. Anjali B. Raut  
Associate Professor & Head of the Department  
Department of Computer Science & Engineering  
HVPM’s college of Engineering and Technology

Abstract

Mining high utility item sets (HUIs) from databases is an important data mining task, which refers to the discovery of item sets with high utilities (e.g. high profits). However, it may present too many HUIs to users, which also degrades the efficiency of the mining process. 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 item sets (CHUIs), which serves as a compact and lossless representation of HUIs. We propose efficient algorithms based on Apriori CH (Apriori-based algorithm for mining High utility Closed + item sets), Apriori HC-D (Apriori HC algorithm with discarding unpromising and isolated items) and CHUD (Closed + High Utility Item set Discovery) to find this representation.

Keywords: Frequent Item Set, Closed+ High Utility Item Set, Lossless and Concise Representation, Utility Mining, Data Mining

I. INTRODUCTION

Data mining involves six common classes of tasks:

1) Anomaly detection (Outlier/change/deviation detection) – The identification of unusual data records, that might be interesting or data errors that require further investigation.

2) Association rule learning (Dependency modelling) – Searches for relationships between variables. For example, a supermarket might gather data on customer purchasing habits. Using association rule learning, the supermarket can determine which products are frequently bought together and use this information for marketing purposes. This is sometimes referred to as market basket analysis.

3) Clustering – is the task of discovering groups and structures in the data that are in some way or another "similar", without using known structures in the data.

4) Classification – is the task of generalizing known structure to apply to new data. For example, an e-mail program might attempt to classify an e-mail as "legitimate" or as "spam".

5) Regression – attempts to find a function which models the data with the least error.

6) Summarization – providing a more compact representation of the data set, including visualization and report generation.

A. Frequent Itemset Mining

Frequent itemset mining. FIM[1][2] is an interesting branch of data mining that focuses on looking at sequences of actions or events. In FIM, the base data takes the form of sets of instances (also called transactions) that each has a number of features (also called items). The task for the FIM algorithm is then to find all common sets of items, defined as those itemsets that have at least a minimum support (exists at least a minimum amount of times). The original algorithm for mining frequent itemsets, which was published in 1993 by Agrawal and is still frequently used. This algorithm functions by first scanning the database to find all frequent 1-itemsets, then proceeding to find all frequent 2-itemsets, then 3-itemsets etc. At each iteration, candidate itemsets of length n are generated by joining frequent itemsets of length n – 1; the frequency of each candidate itemset is evaluated before being added to the set of frequent itemsets. There exist several alternatives to this algorithm, e.g. the FP-growth algorithm, which finds frequent itemsets through building prefix trees. Once a set of frequent itemsets has been found, association rules can be generated. Association rules are of the form A→B, and could be read as “A implies B”. Each association rule has support (how common the precondition is in the dataset), confidence (how often the precondition leads to the consequence in the dataset) and lift (how much more common the consequence is in instances covered by the rule compared to the whole dataset).

B. Utility Mining

Mining high utility itemsets from a database refers to the discovery of itemsets with high utility like profits. Although a number of relevant algorithms have been proposed in recent years, they incur the problem of producing a large number of candidate itemsets for high utility itemsets. Such a large number of candidate itemsets degrades the mining performance in terms of execution time.
and space requirement. The situation may become worse when the database contains lots of long transactions or long high utility itemsets.

To address these challenges, utility mining [3], [8], [16] emerges as an important topic in data mining. 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 [3], [9], crossmarketing in retail stores and biomedical applications [10].

Developing a concise and complete representation of HUIs poses several challenges:

1) Integrating concepts of concise representations from FIM into HUI mining may produce a lossy representation of all HUIs or a representation that is not meaningful to the users.
2) The representation may not achieve a significant reduction in the number of extracted patterns to justify using the representation.
3) Algorithms for extracting the representation may not be efficient. They may be slower than the best algorithms for mining all HUIs.
4) It may be hard to develop an efficient method for recovering all HUIs from the representation.

In this, we consider all of these challenges by proposing a condensed and meaningful representation of HUIs named closed high utility itemsets (CHUIs), which integrates the concept of closed itemset into high utility itemset mining. Our contributions are four-fold and correspond to resolving the previous four challenges:

1) The proposed representation is lossless due to a new structure named utility unit array that allows recovering all HUIs and their utilities efficiently.
2) The proposed representation is also compact. Experiments show that it reduces the number of itemsets by several orders of magnitude, especially for datasets containing long high utility itemsets (up to 800 times).
3) We propose efficient algorithms named AprioriHC (Apriori-based algorithm for mining High utility Closed+ itemset), AprioriHC-D (AprioriHC algorithm with Discarding unpromising and isolated items) and CHUD (Closed+ High Utility itemset Discovery) to find this representation. The AprioriHC and AprioriHC-D algorithms employs breadthfirst search to find CHUIs and inherits some nice properties from the well-known Apriori algorithm [2]. The CHUD algorithm includes three strategies named REG, RML and DCM that greatly enhance its performance. Results show that CHUD is much faster than the state-of-the-art algorithms [3], [5], [12] for mining all HUIs.
4) We propose a top-down method named DAHU (Derive All High Utility itemsets) for efficiently recovering all HUIs from the set of CHUIs. The combination of CHUD and DAHU provides a new way to obtain all HUIs and outperforms UP-Growth [12], one of the currently best methods for mining HUIs. Note that the utility constraint is neither monotone nor antimonotone [2], [7].

In other words, a superset of a lowutility itemset can be high utility and a subset of a HUI can be low utility. Hence, we cannot directly use the anti-monotone property (also known as downward closure property) to prune the search space. To facilitate the mining task, Liu et al. introduced the concept of transaction-weighted downward closure (TWDC) [5]

C. Closed High Utility Itemset Mining

The first point is to discuss is how to incorporate the closed constraint into high utility itemset mining. There are several possibilities. First, define the closure on the utility of itemsets. However, this definition is unlikely to achieve a high reduction of the number of extracted itemsets. A second possibility is to define the closure on the supports of itemsets. In this case, there are two definitions.

- Mine all the high utility itemsets first and then apply the closed constraint.
- Mine all the closed itemsets [11] first and then apply the utility constraint.

- Definition 1 (Closed high utility item set): We define the set of closed high utility item sets as \[ HC = \{ X | X \in L, X = \hat{C}(X), u(X) \geq \min_{utility} \}, \] where \( HC = H = \hat{C} \). An item set \( X \) is called non-closed high utility item set iff \( X \in H \) and \( X \notin \hat{C} \).
- Definition 2 (Promising item): An item \( ip \) is a promising item iff \( TWU(ip) \geq abs_{-\min_{utility}} \). Otherwise, it is an unpromising item.
II. PROPOSED METHODOLOGY

A. Aims and Objectives

The aim and objectives are as follows:
1) Using research keyword only, generate closed item sets summary of different type of paper which may or may not dependent on particular domain.
2) The aim is to insert any no. of paper to summarize data.
3) Summarization can be done with sentence based, paragraph based it depends only on keyword so that user can easily generate summary.
4) Generate User friendly system which can be use by any user those want to summarize their research papers.

B. Algorithm: by using CHUD Algorithm [1] Concept We Proposed the Step in Our System as Follows

Input: D as database the research base paper collection.
Output: Summary of closed itemset
Step:
1) Input number of file as dataset D and and save it as .text.
2) Scan the dataset
3) Enter the keyword to find the sentence that contain the entered keyword in dataset. It is done with word to match with sentence i.e pattern mining.
4) Processing The Research Base Paper with Keyword Found and Generate groups based on keyword search.
5) Remove redundancy and recover itemset to get concise and lossless representation of closed itemsets
6) Get Summaries The research keyword data
III. IMPLEMENTATION

Implementation steps:
1) Setup.
2) User Registration.
3) Get research paper as input and search keyword
4) Generate itemsets
5) Generate closed itemsets
6) Summary of closed itemsets.

IV. RESULT DISCUSSION

1) Time Analysis: Time analysis is calculate in second by providing strat time and end time to search keyword entered by user in document.

2) Count Analysis: Count analysis is generated based on session id of searched keyword
3) User Search Keyword: Gives the list of keyword search by user and their count.

![Fig. 3: Count Analysis](image_url)

**V. APPLICATION OF CLOSED ITEM SET**

The technology is now broadly applied for a wide variety of government, research, and business needs. Applications can be sorted into a number of categories by analysis type or by business function. Using this approach to classifying solutions, application categories include:

1) Enterprise Business Intelligence/Data Mining, Competitive Intelligence
2) E-Discovery, Records Management
3) National Security/Intelligence
4) Scientific discovery, especially Life Sciences
5) Sentiment Analysis Tools, Listening Platforms
6) Natural Language/Semantic Toolkit or Service
7) Publishing
8) Automated ad placement
9) Search/Information Access
10) Social media monitoring

**VI. CONCLUSION**

In our system, we addressed the problem of redundancy in high utility item set mining by proposing a lossless and compact representation named closed + high utility item sets, and Proposed efficient summarization result of dataset. Result show that CHUD is much faster than the state-of-the-art algorithms for mining all HUIs.
REFERENCES