ISSN 2395-1621

# A Survey on Mining High Utility Itemsets

[#1]Khushali Kumari, [#2]Prof. A.R. Deshpande

[1]kumari.khushali@yahoo.in
[2]ardespande@pict.edu

[#1]Student, Computer Department,
[#2]Prof., Computer Department

S.P.P.U University
P.I.C.T., Pune, India

**ABSTRACT**

**In association rule mining, an itemset is a set of items, and each itemset represents a particular transaction. For example, in e-commerce application, an itemset represents a set of items that a customer buy in one transaction. Frequent Itemset Mining (FIM) is a popular data mining task that is essential to a wide range of applications. Given a transactional database, FIM consists of discovering frequent itemsets .i.e. groups of items (itemsets) appearing frequently in transactions. However, an important limitation of FIM is that it assumes that each item cannot appear more than once in each transaction and that all items have the same importance (weight, unit profit or value). To address these issues, the problem of High-Utility Itemset Mining (HUIM) has been defined. As opposed to FIM, HUI considers the case where items can appear more than once in each transaction and where each item has a weight (e.g. unit profit). Therefore, it can be used to discover itemsets having a high-utility (e.g. high profit), that is High-Utility Itemsets. An itemset is called high utility itemset (HUI) if its utility is no less than a user-specified minimum utility threshold min_util. Discovering high-utility itemsets in transaction databases is a popular data mining task. A limitation of traditional algorithms is that a huge amount of high-utility itemsets may be presented to the user. To provide a concise and lossless representation of results to the user, the concept of closed high-utility itemsets was proposed.**

ARTICLE INFO

**Article History**

Received: 15th March 2018

Received in revised form :

15th March 2018

Accepted: 17th March 2018

**Published online :**

**21st March 2018**

**Keywords : high-utility mining, itemset mining, pattern mining.**

## I. INTRODUCTION

**M**ining frequent itemsets from a transaction database refers to the discovery of the itemsets which frequently appear together in the transactions. The main objective of Utility Mining is to identify the itemsets with highest utilities above a user-specified threshold, by considering profit, quantity, cost or other user preferences. If the support of an itemset exceeds a user-specified minimum support threshold, the itemset is considered as frequent. Most frequent itemset mining algorithms employ the downward closure property of itemsets. However, the unit profits and purchased quantities of items are not considered in the framework of frequent itemset mining. The basic meaning of utility is the interestedness/ importance/profitability of items to the users. Traditional FIM algorithms do not consider information about the purchase quantities of items and unit profits of items. Thus, FIM algorithms would discard this information and only find frequent itemsets, rather than finding those yielding a high profit. As a result, many uninteresting frequent itemsets generating a low profit may be discovered, and many rare itemsets generating a high profit may be missed. To address this issue, the problem of High-Utility Itemset Mining (HUIM) has been defined. As opposed to FIM, HUIM considers the case where items can appear more than once in each transaction and where each item has a weight (e.g. unit profit). The goal of HUIM is to discover itemsets having a high-utility (a high importance, such as a high profit), that is High-Utility Itemsets.

High-utility itemset mining has emerged as an important research topic in data mining in recent years, and has inspired several other important data mining tasks such as high-utility sequential pattern mining. Beside market basket analysis, HUIM and its variations have a wide range of applications such as mobile commerce, click stream analysis, biomedicine and cross-marketing .

Several algorithms have been proposed for HUIM. However, an important limitation of traditional HUIM

 

algorithms is that they often produce a huge amount of high-utility itemsets. Hence, it can be very time consuming for users to analyze the output of these algorithms. Moreover, this makes HUIM algorithms suffer from long execution times and even fail to run due to huge memory consumption or lack of storage space. To address this issue, it was recently proposed to mine a concise and lossless representation of all HUIs named closed high-utility itemsets (CHUIs). The concept of CHUI extends the concept of closed patterns from FIM. A CHUI is a HUI having no proper supersets that are HUIs and appear in the same number of transactions. This latter representation is interesting since it is lossless (it allows deriving all HUIs). Furthermore, it is also meaningful for real applications since it only discovers the largest HUIs that are common to groups of customers.

## II. LITERATURE SURVEY

**Two-Phase** (Liu et al., 2005) is an algorithm for discovering **high-utility itemsets** in a transaction database containing utility information. **High utility itemset mining** has several applications such as discovering groups of items in transactions of a store that generate the most profit. A database containing utility information is a database where items can have quantities and a unit price. Although these algorithms are often presented in the context of market basket analysis, there exist other applications. **High utility itemset mining** is a much more difficult problem than frequent itemset mining. Therefore, algorithms for high-utility itemset mining are generally slower than frequent itemset mining algorithms. The Two-Phase algorithm is an important algorithm because it introduced the concept of mining **high utility itemset** by using two phases by first overestimating the utility of itemsets in phase I and then calculating their exact utility in phase II. However, there are now some more efficient algorithms. For efficiency, it is recommended to use a more efficient algorithm such as **EFIM** and is one of the most efficient algorithm for this problem.

**IHUP** (Ahmed et al., TKDE 2009) is an algorithm for discovering **high-utility itemsets** in a transaction database containing utility information. Note that the original IHUP algorithm is designed to be incremental. In this implementation of IHUP can only be run in batch mode. Also note that more efficient algorithm have been recently proposed such as FHM (2014) and HUI-Miner (2012). These latter algorithms outperforms IHUP by more than an order of magnitude. **High utility itemset mining** is a more difficult problem than frequent itemset mining. Therefore, **high-utility itemset mining** algorithms are generally slower than frequent itemset mining algorithms. The **IHUP (2009)** algorithm was the fastest algorithm for high-utility itemset mining in 2009. However, more efficient algorithm have been recently proposed. **UPGrowth** (2010) is an improved version of IHUP. The **HUI-Miner (2012)** algorithm outperforms UPGrowth (2009) by more than an order of magnitude, and more recently **the FHM algorithm (2014)** was shown to be up to six times faster than HUI-Miner. More recently, the **EFIM** algorithm (2015) was proposed and was shown to outperform IHUP, and other recent algorithms such as FHM (2014), HUI-Miner (2012), HUP-Miner (2014).

**UP-Growth** (Tseng et al., KDD 2010) is an algorithm for discovering **high-utility itemsets** in a transaction database containing utility information. **UP-Growth+** (Tseng et al., KDD 2012) is an improved version. Those two algorithms are important algorithms because they introduce some interesting ideas. However, recently some more efficient algorithms have been proposed such as **FHM** (2014) and **HUI-Miner** (2012). These latter algorithms were shown to be more than 100 times faster than UP-Growth+ in some cases. **High utility itemset mining** is a more difficult problem than frequent itemset mining. Therefore, high-utility itemset mining algorithms are generally slower than frequent itemset mining algorithms. The **UP-Growth (2010)** algorithm was the fastest algorithm for **high-utility itemset mining** in 2010. However, more efficient algorithm have been proposed. The **HUI-Miner (2012)** was shown to be up to 100 times faster than UP-Growth, and more recently **the FHM algorithm (2014)** was shown to be up to six times faster than HUI-Miner. More recently, the **EFIM** algorithm (2015) was proposed and was shown to outperform UPGrowth+ and other recent algorithms such as FHM (2014), HUI-Miner (2012), HUP-Miner (2014).

**FHM** (Fournier-Viger et al., ISMIS 2014) is an algorithm for discovering **high-utility itemsets** in a transaction database containing utility information. **High utility itemset mining** has several applications such as discovering groups of items in transactions of a store that generate the most profit. A database containing utility information is a database where items can have quantities and a unit price. Although these algorithms are often presented in the context of market basket analysis, there exist other applications. **High utility itemset mining** is a more difficult problem than frequent itemset mining. Therefore, high-utility itemset mining algorithms are generally slower than frequent itemset mining algorithms. **The FHM algorithm** was shown to be up to six times faster than **HUI-Miner** (also included in SPMF), especially for sparse datasets (see the performance section of the website for a comparison). But the **EFIM** algorithm greatly outperforms FHM.

**HUP-Miner** (Krishnamoorthy, 2014) is an extension of the **HUI-Miner** algorithm (Liu & Qu, CIKM 2012) for discovering **high-utility itemsets** in a transaction database containing utility information. It introduces the idea of partitioning the database and another pruning strategy named LA-prune. A drawback of HUP-Miner is that the user needs to set an additional parameter, which is the number of partitions. Moreover, according to our experiments, HUP-Miner is faster than HUI-Miner but slower than FHM. **High utility itemset mining** has several applications such as discovering groups of items in transactions of a store that generate the most profit. A database containing utility information is a database where items can have quantities and a unit price. Although these algorithms are often presented in the context of market basket analysis, there exist other applications. **High utility itemset mining** is a more difficult problem than frequent itemset mining. Therefore, high-utility itemset mining

 

algorithms are generally slower than frequent itemset mining algorithms. The **HUI-Miner** algorithm was reported as one of the most efficient algorithm for **high utility itemset mining**. HUP-Miner is an extension of HUI-Miner, just like FHM. These two latter are faster than HUI-Miner. However, HUP-Miner introduce a new parameter which is the number of partitions. In our experiment, FHM is faster than HUP-Miner.

**CHUI-Miner** (Wu et al., 2014) is an algorithm for discovering **closed high-utility itemsets** in a transaction database containing utility information. There has been many work on the topic of high-utility itemset mining. A limitation of many **high-utility itemset mining** algorithms is that they generate too much itemsets as output. The CHUI-Miner algorithm was designed to discover only the high-utility itemsets that are **closed**. The concept of **closed itemset** was previously introduced in **frequent itemset mining**. An itemset is closed if it has no subset having the same support (frequency) in the database. In terms of application to transaction database, the concept of closed itemset can be understood as any itemset that is the largest set of items bought in common by a given set of customers. It provides a more details about the motivation for mining closed high-utility itemsets. **High utility itemset mining** is a more difficult problem than frequent itemset mining. Therefore, **high-utility itemset mining** algorithms are generally slower than frequent itemset mining algorithms. The **CHUI-Miner** algorithm was proposed in 2015 to discover only the high-utility itemsets that are closed itemset. It is generally faster than discovering all high-utility itemsets. Thus, this algorithm can in some cases outperform algorithms such as **FHM** and **HUI-Miner,** who discover all high-utility itemsets. The **CHUI-Miner** algorithm is an improved version of the **CHUD** algorithm published in the proceedings of the ICDM 2011 conference.

**EFIM** (Zida et al., 2015) is an algorithm for discovering **high-utility itemsets** in a transaction database containing utility information. **High utility itemset mining** has several applications such as discovering groups of items in transactions of a store that generate the most profit. A database containing utility information is a database where items can have quantities and a unit price. Although these algorithms are often presented in the context of market basket analysis, there exist other applications. **High utility itemset mining** is a more difficult problem than frequent itemset mining. Therefore, high-utility itemset mining algorithms are generally slower than frequent itemset mining algorithms. **The EFIM algorithm** was shown to be **up to two orders of magnitude faster** than the previous state-of-the-art algorithm **FHM, HUI-Miner, d2HUP, UPGrowth+** (also included in SPMF), and consumes **up to four times less**.

For evaluation purpose all type of datasets such as both synthetical and real time dataset can be used. For our system datasets that is going to be used are foodmart which is obtained from microsoft foodMart 2000 and chainstore dataset obtained from NU-Mine-Bench 2.0. Other than this various other datasets can be used such as Mushroom, chess and accident. Characteristics of all these datasets are given table I.

Datasets' characterictics

| Dataset | transaction count | distinct item count | average transaction length |
|---|---|---|---|
| Accidents | 340,183 | 468 | 33.8 |
| BMS | 59,601 | 497 | 4.8 |
| Chess | 3,196 | 75 | 37 |
| Connect | 67,557 | 129 | 43 |
| Foodmart | 4,141 | 1,559 | 1,559 |
| Mushroom | 8,124 | 119 | 23 |

- **Foodmart** is a real-life transaction datasets from retail stores.

## III. SYSTEM REQUIREMENTS

For developing a system that is capable of calculating high utility itemset if transaction with their profits are given, has some minimum hardware and software requirements .

a. Hardware requirement:

- Processor- Intel Quad core,i3,i5 and above.
- Architecture- 32 bit.
- RAM- 2 GB.
- Hard disk- 500 GB.

b. Software requirement:

- Operating System- 32bits windows(7,8)/linux(fedora, ubunto).
- Platform- eclipse.
- Language- Java.

## IV. CONCLUSION

An efficient approach for mining high utility closed itemset could be using support_count consideration. It relies on two new upper-bounds named sub-tree utility and local utility, and an array-based utility counting approach named Fast Utility Counting. Moreover, to reduce the cost of database scans, Closed HUI proposes two efficient techniques named High-utility Database Projection and High-utility Transaction Merging. Lastly, to discover only closed HUIs, three mechanisms are proposed: (1) forward closure checking, (2) backward closure checking, and (3) closure jumping. Datasets for evaluating experimental result would be both synthetical and real. For synthetical foodmart, mushrom and chess dataset will be used

## ACKNOWLEDGEMENT

I also sincerely convey my gratitude to my guide Prof. A.R Deshpande, Department of Computer Engineering for his constant support, providing all the help, motivation and encouragement from beginning till end to make this a grand success.

I am also hugely indebted to my friends for all their help and support. Above all I would like to thank my parents for their wonderful support and blessings, without which I would not have been able to accomplish my goal.

 

## REFERENCES

[1] Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: Proc. Int. Conf. Very Large Databases, pp. 487{499, (1994)

[2] Ahmed, C. F., Tanbeer, S. K., Jeong, B.-S., Lee, Y.-K.: Efficient tree structures for high-utility pattern mining in incremental databases. IEEE Trans. Knowl. Data Eng. 21(12), 1708{1721 (2009)

[3] Fournier-Viger, P., Wu, C.-W., Zida, S., Tseng, V. S.: FHM: Faster high-utility itemset mining using estimated utility co-occurrence pruning. In: Proc. 21st Intern. Symp. on Methodologies for Intell. Syst., pp. 83{92 (2014)}

[4] Fournier-Viger, P., Gomariz, A., Gueniche, T., Mwamikazi, E., Thomas, R.: Efficient Mining of Top-K Sequential Patterns. In: Proc. 9th Intern. Conf. on Advanced Data Mining and Applications Part I, pp. 109{120, Springer (2013)}

[5] Fournier-Viger, P., Gomariz, A., Gueniche, T., Soltani, A., Wu., C., Tseng, V. S.: SPMF: a Java Open-Source Pattern Mining Library. Journal of Machine Learning Research (JMLR), 15, pp. 3389-3393 (2014)

[6] Lan, G. C., Hong, T. P., Tseng, V. S.: An efficient projection-based indexing approach for mining high utility itemsets. Knowl. and Inform. Syst. 38(1), 85{107(2014)}

[7] Song, W., Liu, Y., Li, J.: BAHUI: Fast and memory efficient mining of high utility itemsets based on bitmap. Intern. Journal of Data Warehousing and Mining. 10(1), 1{15 (2014)}

[8] Liu, M., Qu, J.: Mining high utility itemsets without candidate generation. In: Proc.22nd ACM Intern. Conf. Info. and Know. Management, pp. 55{64 (2012)}

[9] Liu, Y., Liao, W., Choudhary, A.: A two-phase algorithm for fast discovery of high utility itemsets. In: Proc. 9th Pacific-Asia Conf. on Knowl. Discovery and Data Mining, pp. 689{695 (2005) }

[10] Tseng, V. S., Shie, B.-E., Wu, C.-W., Yu., P. S.: efficient algorithms for mining high utility itemsets from transactional databases. IEEE Trans. Knowl. Data Eng. 25(8), 1772{1786 (2013)}

[11] Tseng, V., Wu, C., Fournier-Viger, P., Yu, P.: efficient algorithms for mining the concise and lossless representation of closed+ high utility itemsets. IEEE Trans Knowl. Data Eng. 27(3), 726{739 (2015)}

[12] T. Uno, M. Kiyomi, H. Arimura, "LCM ver. 2: Efficient mining algorithms for frequent/closed/maximal itemsets," Proc. ICDM'04 Workshop on Frequent Itemset Mining Implementations, CEUR, 2004.

[13] Wang, J., Han, J., Li, C.: Frequent closed sequence mining without candidate maintenance. IEEE Trans. on Knowledge Data Engineering 19(8), 10421056 (2007)

[14] Yun, U., Ryang, H., Ryu, K. H.: High utility itemset mining with techniques for reducing overestimated utilities and pruning candidates. Expert Syst. with Appl. 41(8), 3861{3878 (2014)}

[15] Zida, S., Fournier-Viger, P.,Wu, C.-W., Lin, J. C. W., Tseng, V.S.: Efficient mining of high utility sequential rules. In: Proc. 11th Intern. Conf. Machine Learning and Data Mining (MLDM 2015), pp. 1{15 (2015)}

[16] http://www.philippe-fournier-viger.com.

 