Improved EFIM Algorithm to Efficiently Mine High Utility Itemset

Harsh Vasani , Ashwin Raiyani , Ravirajsinh Vaghela
Department of Computer Engineering, RK University, Rajkot, India.
Head Of The Department, School of Computer Engineering, RK University, Rajkot,India
Assistant Professor, School of Engineering, RK University, Rajkot,India

1.

High-utility itemset mining (HUIM) is an important data mining task with broad applications. In this paper, we propose a novel algorithm named EFIM (EFficient high-utility Itemset Mining), which introduces a number of new ideas to more efficiently discovers high-utility itemsets both in terms of execution time and memory. EFIM trust on two upper-bounds named *sub-tree utility* and *local utility* to more effec- tively prune the search space. It also introduces a novel array-based util- ity counting technique named *Fast Utility Counting* to calculate these upper-bounds in linear time and space. Moreover, to decreses the cost of database scans, EFIM come up with efficient database projection and trans- action merging techniques. An extensive experimental study on var- ious datasets appears that EFIM is in general two to three orders of magnitude faster and consumes up to eight times fever memory.

Keywords: High-utility mining · Itemset mining · Pattern mining

2. INTRODUCTION

2.1. Data mining

Data mining is a process to extract the implicit information and knowledge which is potentially useful and people do not know in advance, and this extraction is from the mass, incomplete, noisy, fuzzy and random data.[1]

The difference between the data mining and the traditional data analysis (such as query, reporting and online application of analysis) is that the data mining is to mine information and discover knowledge on the premise of no clear assumption.

In addition to industry-driven demand for standards and interoperability, professional and academic activity has also made considerable contributions to the evolution of the methods and models; an article published in a 2008 issue of the International Journal of Information Technology and Decision Making summaries the results of a literature survey which traces and analyze this evolution.[1]

Data mining is the use of automated data analysis techniques to uncover previously undetected relationships among data items. Data mining often involves the analysis of data stored in a data warehouse. The major data mining techniques are regression, classification and clustering.

Data mining approach is quickly adapted and used in a large number of domains such as health care, pharmacy, business, finance, biological data analysis, web data analysis. The main data mining tasks are as follows:

- **Classification:** Classification is the process of finding models that analyze and classify a data item into various predefined classes.
- **Prediction:** It is the process of predicting value of specific attributes.
- **Regression**: Regression is statistical process of mapping a data item to a real-valued divination variable.
- **Clustering:** It is a process of identifying or grouping a set of physical objects or items into classes of similar objects.
- **Summarization:** Summarization is a process of finding a compact description for a subset of data.
- **Association:** Identify the significant dependencies between data attributes and associate them with each other.

2.2. Background

Database and knowledge discovery communities [1] have focused on mining frequent patterns from data streams. A stream data are an unbounded and infinite sequence of data elements continuously arrived at a speedy rate.

High utility itemsets mining extends frequent pattern mining. The first step of High Utility Itemset mining is finding all the item set that frequently appears in a dataset. The second step is to find all itemsets which utility have larger than a user specified value of the minimum utility. The value or profit associated with the every item in a database is addressed the utility of that itemset.

In the transaction database [2], there are two types of utilities for items, the internal utility and the external utility. The internal utility of an item represents the importance of an item in the transaction, for example, the quantity of an item purchased in the transaction. The external utility of an item is defined according to the user objectives, for example, the unit profit value of an item, which is not available in transaction.

The usefulness of item sets is not considered in common, frequent itemset mining. For example, a consumer buys multiple items of different quantities in a transaction. In general, each item is associated with a certain amount of profits. For instance, given the profit of “bread and butter” are 5 that of “birthday cake” is 30. The “bread and butter” occurred in 6 transactions and “birthday cake” occurred in 2 transactions in a transactional database. Infrequent itemsets mining, the occurred frequency of “bread and butter” is 6 and the “birthday cake” is 2. If the user specifies a minimum threshold of 3, “bread and butter” will be frequent while “birthday cake” will be infrequent. Nevertheless, the total profit of “birthday cake” is 60 and that of “bread and butter” is 30, “birthday cake” contributes more than “bread and butter” does. Because frequent itemsets mining considers only whether the items have occurred or not, the associated profits are ignored in the mining. Frequent itemsets represent itemsets of high frequency. In comparison, some infrequent itemsets may contribute more profits to the total profit in the database than frequent itemsets do.

The restraint of frequent itemset mining is, it assumes (1) an item can be only appear one time in a transaction (2) all the items have the same importance or weight (e.g. Profit). So in that case it ignores rare itemset having higher profit. To get solution of this issue, the High-Utility Itemset Mining (HUIM) is useful technique.

1. Scope and Objective of the work

Scope:

In current market, every item in the big market which has a momentous selling and rate and a single customer will be interested in buying multiple copies of the same item. Therefore, discover only traditional frequent patterns in a database and it the alogrithm don't have capablities to find most profit making itemsets which cover big percentage of total profit in a retail business. For that utility mining is used.

Objective:

- Find all high utility itemset efficiently from transaction database.
- For large dataset it occupies efficient memory space.
- Find all high utility itemset having utility values above the given threshold.

2 Applications of High Utility Itemset Mining

- Website clickstream analysis
- Cross marketing in retail stores
- Online e-commerce management
- Finding important patterns in biomedical applications Finding important patterns in biomedical applications

3 Open Issues

There are several open issues of High Utility Itemset Mining.

Support Threshold

If we apply a lower threshold value, then it generates more number of Itemsets being declared as frequent. It gives a negative effect on the computational complexity.

Number of Items

If the number of items increases, more space is needed for storing the support count of items. But also, if the number of frequent items grows, then the computation will increase.

Candidate Generation

Candidate Generation depends on which data structure is used by the algorithm. If more candidates are generated, then it requires more memory to store it for further performance.

Join Operation

If numbers of join operations are higher than it takes more time to prune it.

3 HIGH UTILITY ITEMSET MINING FRAMEWORKS

3.1 High Utility Itemset Mining

(HUIM) is an important itemset from a database whose frequency data mining problem, which detects frequent pattern higher than the user-specified threshold and having higher profit.[2]

The following Diagram shows the chain process of calculating and displaying a high utility Itemsets. From this Fig. , the comparison of frequent Itemset with given threshold value and by considering a profit, gives High utility Itemsets.

9

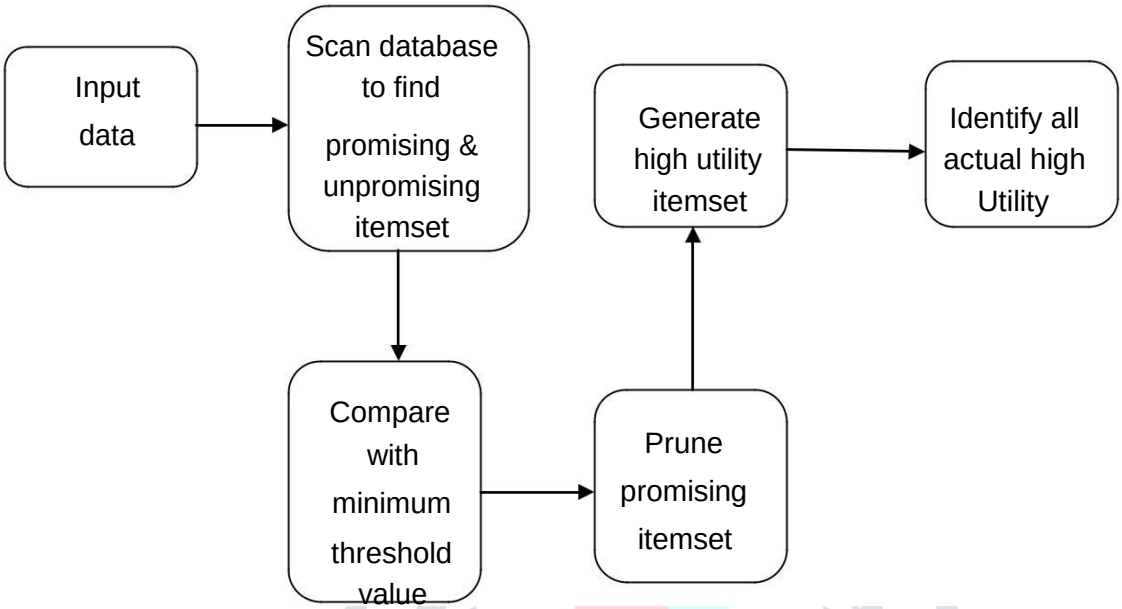

Figure 2.1 Data Flow Diagram

4. LITERATURE SURVEY

4.1 Literature survey

1. Two-Phase

Two-Phase method maintains a *Transaction-weighted Downward Closure Property*. Thus, only the combinations of high transaction weighted utilization itemsets are added into the candidate set at each level during the level-wise search. Phase I may overestimate some low utility itemsets, but it never underestimates any itemsets. In phase II, only one extra database scan is performed to filter the overestimated itemsets. Example of Two-Phase is given below. [3].

Two-phase Algorithm Example

Table 1. Transaction dataset

Transactions	Set of Items	Trans. Utility	Item Utility for this Trans.

Transaction 1	{3 5 1 2 4 6}	30	{1 3 5 10 6 5}
Transaction 2	{3 5 2 4}	20	{3 3 8 6}
Transaction 3	{3 1 4}	8	{1 5 2}
Transaction 4	{3 5 1 7}	27	{6 6 10 5}
Transaction 5	{3 5 2 7}	11	{2 3 4 2}

Each lines of the **database** is:

- A set of items (column no. one from the table),
- The addition of the utilities (e.g. profit) of these items in this transaction (column no. two from the table),
- The utility for every item for this transaction (e.g. profit produced by this item for this Transaction) (column no. three from the table).

The customer named "**Transaction1**" bought items 3, 5, 1, 2, 4 and 6. The amount of money spent for every item is respectively 1, 3, 5, 10, 6 and 5.

The total amount of money spent in the transaction is 1 + 3 + 5 + 10 + 6 + 5 = 30. For eg,

The Utility of the itemset {1 4} in trans. T1 is 5 + 6 = 11

The Utility of the itemset {1 4} in trans. T2 is 0 (not appear)

The Utility of the itemset {1 4} in trans. T3 is 5 + 2 = 7

The Utility of the itemset {1 4} in trans. T4 is 0 (not appear)

The Utility of the itemset {1 4} in trans. T5 is 0 (not appear)

The **utility of an itemset in a database** is the additions of its utility in all transactions where it become visible.

For eg, the utility of {1 4} in the database is the utility of {1 4} in T1 plus the utility of {1 4} in T3,

u {1,4} = u(T1) + u(T3) = 11 + 7 = 18

2. Hui-Miner

Liu & Qu (2012) proposed HUI-Miner algorithm [6]. HUI-Miner algorithm is a high utility itemset with a list data structure, which is called utility list. An algorithm first creates an initial utility list for itemsets of the length 1 for guaranteed items. After that the HUI-Miner algorithm constructs recursively a utility list for every itemset of the length k using a pair of utility lists for itemsets of the length k-1. For (HUIM), each utility list for an itemset keeps their information of TIDs for all of transactions containing the itemset, utility values of the item set in the transactions, and the sum of utilities of the remaining items that can be included to super itemsets of the itemset in the transactions. The only advantage of HUI-Miner is that it avoids the costly candidate generation and utility computation.

3. FHM

Philippe Fournier-Viger (2014) proposed FHM algorithm [6]. proposed FHM algorithm extends the Hui-Miner Algorithm. proposed FHM algorithm is a Depth-first search Algorithm. proposed FHM algorithm relies on utility-lists to calculate the exact utility of itemsets. proposed FHM algorithm integrates a novel strategy named EUCP (Estimated Utility Co-occurrence Pruning) to reduce the number of joins operations when mining high-utility itemsets using the utility list data structure. Estimated Utility Co-Occurrence Structure (EUCS) stores the TWU of all 2-itemsets. proposed FHM algorithm built during the initial database scans. EUCS represented as a triangular matrix or hashmap of hashmaps. The memory footprint of the EUCS structure is small. FHM is **up to 6 times faster** than HUI-Miner.

4. CTU-Mine

Erwin et al observed that the conventional candidate-generate-and-test approach for identifying high utility itemsets is not suitable for dense data sets[4]. Their work proposes a novel algorithm CTU-Mine that mines high utility itemsets using the pattern growth approach. A similar argument is presented by Yu et al. Existing algorithms for high utility mining are column enumeration based adopting an Apriori-like candidate set generation-and-test approach and thus are inadequate in datasets with high dimensions.

5. UP-Growth

To overcome this issue the algorithm produce very large number of itemsets, UP-Growth has been inventing and it uses PHU model[5].To reduce the number of candidate itemsets, the algorithm applies four strategies, DGU, DGN, DLU, and DLN. In the backend, it builds a tree structure which name is UP Tree, with two database scans and conducts (HUIM) itemsets. In the different words,

an algorithm demanding for three database scan for discovering(HUIM). In the first one database scan, TWU values of every item are collected.In next database scan, items which had a law TWU value than the user-specified minimum utility threshold is removed from every transaction.

6. THUI

A novel method, namely THUI (Temporal High Utility Itemsets) –Mine was proposed by V.S. Tseng et al for mining temporal high utility itemsets from data streams efficiently and effectively [3]. The novel contribution of THUI-Mine is that it can effectively identify the temporal high utility itemsets by generating fewer temporal high transaction weighted utilization 2-itemsets such that the execution time can be reduced substantially in mining all high utility itemsets in data streams. In this way, the process of discovering all temporal high utility itemsets under all time windows of data streams can be achieved effectively with limited memory space, fever candidate itemsets and CPU I/O time. This meets the critical requirements on time and space efficiency for mining data streams. The experimental results show that THUI-Mine can discover the temporal high utility itemsets with high performance and fewer candidate itemsets as compared to other algorithms under various experimental conditions. Moreover, it performs scalable in terms of execution time under large databases. Hence, THUI-Mine is promising for mining temporal high utility itemsets in data streams.

7. EFIM

EFIM(Efficient high-utility itemset mining) introduces some new thoughts to more efficiently discovers high-utility itemsets for both in terms of execution time and storage[7].An algorithm relies on two upper-bounds whose name is sub-tree utility and local utility to more effectively prune the search space. An algorithm also introduces a novel strategy technique called Fast Utility Counting to calculate these upper-bounds in linear time and space. Transaction merging is obviously desirable. To find uniform transactions in O(n) time, sort the actual database according to a new total order T on transactions. Sorting is achieved in time and is performed only once. Projected databases generated by EFIM are often very small due to transaction merging.

4.2 Comparison of Existing Algorithm

In the previous section we introduced the overview of Data Mining, Frequent Itemset Mining and High Utility Itemset Mining. A comparison of the different Algorithms, Techniques, approaches and limitations that has been explained in various research publications which have been given in this section.

Table 2 Survey Table

Sr No	Studies	Year	Algorithm	Dataset	Limitation
1	Souleymane Zida, Philippe Fournier-Viger, Jerry Chun-Wei Lin, Cheng-Wei Wu, Vincent S. Tseng	2009	CTU-MINE ALGORITHM(Pattern Growth Approach)	Transactional Dataset	it only considers the pattern growth approach. There is no any concept of the threshold value
2	Guo-Cheng Lan, Tzung-Pei Hong, Vincent S. Tseng	2010	TWU ALGORITHM(transaction weighted Utility) (Pattern Growth Approach)	Transactional Dataset	it was not applicable for real use for the supermarket and for other promotion application.
3	Vincent S. Tseng, Bai-En Shie, Cheng-Wei Wu, and Philip S. Yu, Fellow	2013	UP growth algorithm & UP+ growth algorithm	Transactional Dataset	it took more time for result when a bulk of transaction have arrived for utility
4	Philippe Fournier-Viger1, Cheng-Wei Wu2, Souleymane Zida1, Vincent S. Tseng2	2014	it uses FHM (Fast High-Utility Miner) algorithm	Transactional Dataset	for huge dataset still it was a time consuming
5	Souleymane Zida, Philippe Fournier-Viger, Jerry Chun-Wei Lin, Cheng-Wei Wu, Vincent S. Tseng	2015	it uses FIM & EFIM algorithms	Transactional Dataset	Time-consuming

5 PROPOSED APPROACH

5.1 Problem Definition

- For Candidate generation For Candidate generation
- Lower Support - Large candidate generation
- Higher Support – Very few candidate generation
- How to store efficiently and access generated items

5.2 Proposed Approach

(HUIM) is an important data mining task,in which different algorithms have been proposed to perform this task efficiently. The problem is to find all high utility itemset with higher profit, and whose support is higher than the minimum support threshold given by the users.

For high utility itemset mining various methods are introduced which are used to reduce memory space and time. Existing Efficient high utility itemset mining (EFIM) for the large pattern it generates more candidate itemset. So we improve the existing algorithm to reduce time.

5.3 Proposed Flow Diagram

Step 1: Load the test dataset

Step 2: Find Local Utility

Step 3: Find Transaction weighted utility (TWU)

Step 4: Prepare array for TWU

Step 5: Calculate Sub-tree utility

Step 6: Identify all high utility itemset

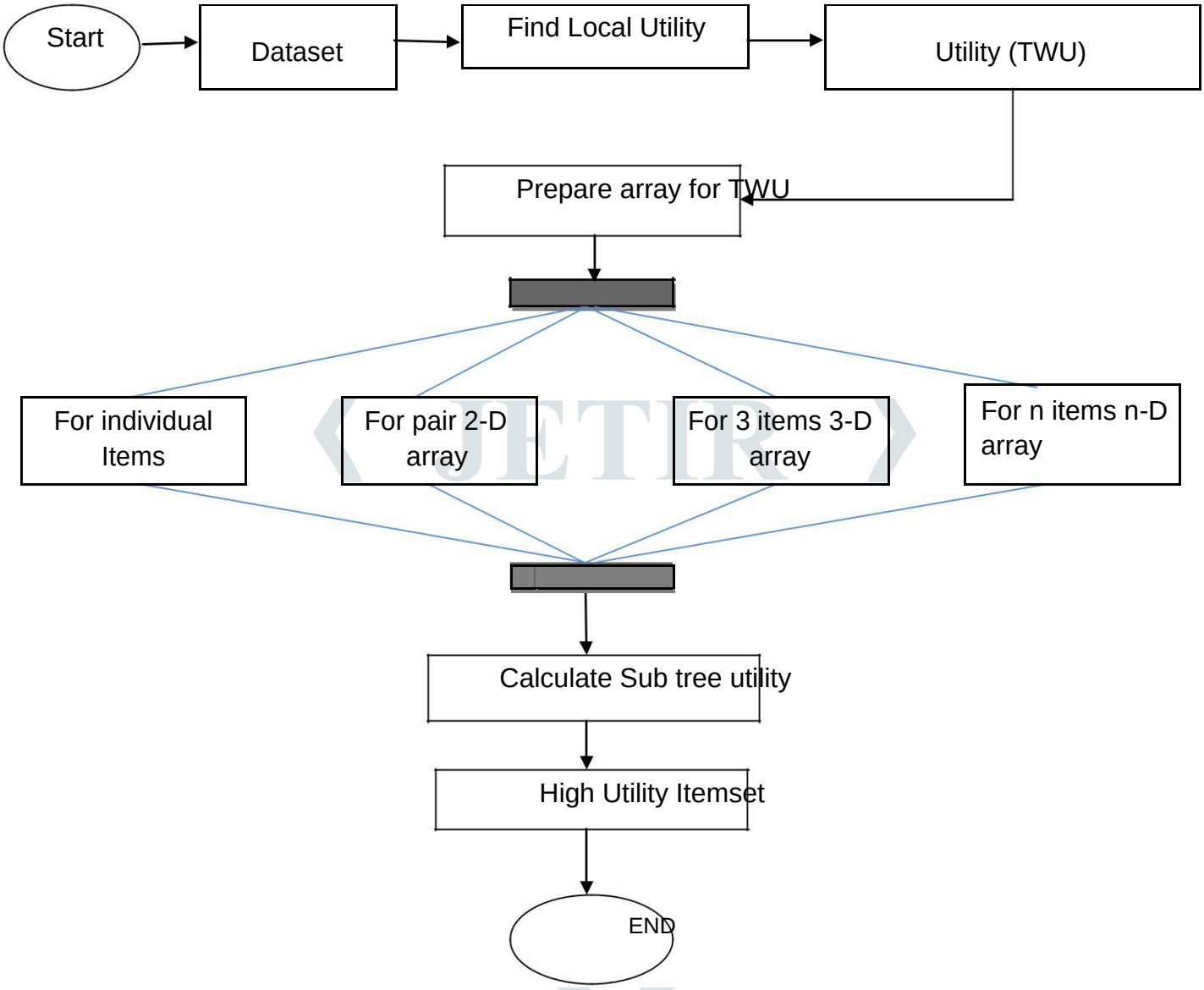

Figure 4.2 Flowchart of the proposed approach

6. CONCLUSION

In Data Mining, Association Rule Mining is one of the most dominant tasks. A huge number of efficient algorithms are available for association rule mining, which contemplate mining of frequent itemsets. But an emerging topic in Data Mining is High Utility Itemset Mining, which integrate utility considerations during itemset mining. Utility Mining covers all aspects of economic utility in data mining and assist in detection of itemset having high utility, like profit.. I implement novel strategy Length Upper-Bound Reduction (LUR) to improve EFIM algorithm that will reduce search space and running

time of algorithm to find HUI efficiently. The result shows that the proposed algorithm takes less time as compare to existing algorithm.

7. REFERENCES

[1] Hemlata Sahu, Shalini Sharma, Seema Gondhalakar, "A Brief Overview on Data Mining Survey", International Journal of Computer Technology and Electronics Engineering (IJCTEE) Volume 1, Issue 3

[2] Maya Joshi, Mansi Patel, "A Survey on High Utility Itemset Mining Using Transaction Databases", International Journal of Computer Science and Information Technologies, Vol. 5 (6),7407-7410,2014.

[3] Sarode Nutan S, Kothavle Suhas R, "An Efficient Algorithm for finding high utility itemsets from online sell", International Research Journal of Engineering and Technology (IRJET) , Volume: 02, Issue: 05 | Aug-2015.

[4] Prashant V. Barhate, S. R. Chaudhari, P. C. Gill, "Efficient High Utility Itemset Mining using Utility Information Record" , International Journal of Computer Applications (0975 – 8887), Volume 120 – No.4, June 2015.

[5] Jyothi Pillai, O.P. Vyas, "Overview Of Itemset Mining And its Application", International Journal of Computer Applications (0975 – 8887),Volume 5– No.11, August 2010.

[6] "CTU-Mine: An Efficient High Utility Itemset Mining Algorithm Using the Pattern Growth Approach "Alva Erwin1, Raj P. Gopalan1, N.R. Achuthan2.IEEE 2007.

[7] "Mining High Transaction-Weighted Utility Itemsets" Guo-Cheng Lan, Tzung-Pei Hong, Vincent S. Tseng. IEEE 2010.

[8] "Efficient Algorithms for Mining High Utility Itemsets from Transactional Databases" Vincent S. Tseng, Bai-En Shie, Cheng-Wei Wu, and Philip S. Yu, Fellow, IEEE 2013

[9] "FHM: Faster High-Utility Itemset Mining using Estimated Utility Co-occurrence Pruning" Philippe Fournier-Viger1, Cheng-Wei Wu2, Souleymane Zida1, Vincent S. Tseng2. Springer 2014.

[10] "EFIM: A Highly Efficient Algorithm for High-Utility Itemset Mining." Souleymane Zida1, Philippe Fournier-Viger1, Jerry Chun-Wei Lin2, Cheng-Wei Wu3, Vincent S. Tseng3. IEEE 2015.

[11] "A Two-Phase Algorithm for Fast Discovery of High Utility Itemsets." Ying Liu, Wei-keng Liao, and Alok Choudhary Springer-Verlag Berlin Heidelberg 2005.

[12] "Mining High Utility Itemsets without Candidate Generation." Mengchi Liu, Junfeng Q–u. 2012.