Efficient High-Utility Itemset Mining Over Variety of Databases: A Survey

U. Suvarna and Y. Srinivas

Abstract High-utility itemset mining (HUIM) is creating new heights of challenge in the areas of research in big data analytics blending data structures, AI, and machine learning techniques to find efficient utility patterns. In the recent past, the merchandise market researchers were pretty much interested in analyzing the purchasing patterns and forecast new profitable area of business. HUIM is that one kind where we get business intelligence and has become the most basic method of finding the knowledge in decision making and optimizing the decisions of market analysis, streaming analysis, biomedicine, mobile computing, stock exchanges, etc. HUIM was initially applied widely in the transactional databases, later applied recursively in incremental databases, further moved to dynamic databases like temporal, spatial, and data stream databases. This chapter highlights the insights of various existing algorithms present in the literature based on several applications and tries to update the various availabilities of different approaches for coining latest research in this domain.

Keywords Associate rule · Itemset/pattern mining · High-utility itemset mining (HUIM) · Frequent · Periodic · Incremental

1 Introduction

Initially before data mining came into existence, huge amount of data was collected on a daily basis and are stored in the databases for years together and of no business interest. It all started with the retailers who were collecting large data on daily day-to-day activities from their grocery stores. Retailers were interested to analyze

U. Suvarna (✉)
Department of CSE, Sri Sivani College of Engineering, Srikakulam 532410, AP, India
e-mail: viksonio@gmail.com

U. Suvarna · Y. Srinivas
Department of IT, Gitam University, Visakhapatnam 530045, AP, India
e-mail: sriteja.y@gmail.com

J. Nayak et al. (eds.), *Soft Computing in Data Analytics*,
Advances in Intelligent Systems and Computing 758,
https://doi.org/10.1007/978-981-13-0514-6_76

their data to learn about the behavior pattern of their customers purchase. This gave life to a new business intelligence method called data mining. Such data which is stored can be used to encourage variety of applications [1] related to business like market promotions, store management, workflow management, and fraud detections.

Data mining (DM) is a skill of extracting the knowledge, discovering the potential information or patterns from the data available in the large databases. Associate rule mining (ARM) data mining is the base for itemset mining. Later, frequent itemset mining (FIM) and HUIM mining patterns came into existence. Practically, the FIM is limited by the intendment of the discovered itemsets and quantity is not considered (all items are viewed as having the same importance) [2], high-utility itemsets introduced utility [3], a measure which is a weight/profit associated with each itemset. In [4–6], examples explain that all FIMs are not profitable and all high profits need not be frequent. Due to this fact, many itemsets emerged like [7] rare itemsets, [8] infrequent itemsets, and [9] closed itemsets, [10] lattice-based mining. Most types of itemsets use minimal frequencies and maximal frequencies for the algorithm. The HUIM algorithms are mainly developed using

Table 1 HUIM application on various databases and their references

Application databases	Process to apply HUIM by	References
Relational	Looking patterns or extracting knowledge from multiple tables (relations) from a relational database	[11]
Transaction	A set of transaction records, each with a time stamp, an identifier and a set of items. Associated with the transaction files could also be descriptive data for the items	[6, 8, 12 – 23]
Time Series	Hidden patterns that have complex characteristics (complex, non-periodic, irregular, chaotic) time series data, arrays of numbers indexed by time	[24–26]
Temporal	Mining from existing sequence of data items (state transitions), stores data related to time instances (past, present, and future) using a valid time	[18, 27, 28]
Data Stream	Mining knowledge structures (high-dimensional) from continuous, massive, dynamic, speed data records	[12, 29, 18, 30– 32, 28, 33]
Incremental	Iterative or recursive process of mining the results of previous queries from the database (partial data loads)	[34, 35, 32, 26, 36– 38]
Sequential	Sequential values delivered which are statistical in nature, mostly discrete values	[10, 39, 40]
Periodic	Finding frequent periodic patterns (partial/complete)	[24, 19, 25, 26, 41]
On-shelf	Having positive or negative unit profits. The shelf time of items is considered which are no longer in use (negative unit of profits)	[12, 42]
Sky mine	Mining text file, where skyline is the HUIs of each line	[43]
Regular	Appearing at regular intervals defined by user	[30, 33]
Uncertain	The noisy, irregular, and not certain data which is complex in nature unlike precise databases	[44]

Table 2 Application domain

Domain application	References
Education	[7]
Mobile computing	[45]
Streaming	[30, 42]
Market basket analysis	[17, 20, 46]
Stock exchange	[47]

Table 3 Application techniques

Technique	References
Upper bound	[48]
Constraint-based	[20]
Minimal frequencies	[49]
Maximal frequencies	[50, 51]
Flexible	[24, 31]
Erasable	[29, 32]
Conceptual	[29, 36, 52]

four aspects: (a) on the kind of database application (Table 1), (b) possible data structures or novel structures (Sect. 3.2), (c) problem or nature of the application domain (Table 2), and (d) applying techniques like upper/lower bounds, constraints, minimal/maximal (Table 3), optimal in combination with the above three aspects.

The rest of the chapter has been organized as follows: Sect. 2 describes the HUIM, Sect. 3 describes the applications of HUIM, advancements of HUIM covered in Sect. 4, and Sect. 5 is all about the critical analysis and suggestions. Finally, Sect. 6 concludes the work with some necessary future directions.

2 High-Utility Itemset Mining

R. Agrawal et al. (1993) explain the associate rule mining (ARM) which is a traditional mining technique to find the frequent interesting patterns among any nature of a dataset or a database and generates rules that forecast the occurrence of an item based on the co-occurrences of other items in the transaction.

2.1 *Frequent Itemsets/Interesting Itemsets*

To find interesting/ frequent itemsets, we need to know the itemset, frequent itemset, and support measures. A frequent item (FI) [2] is the itemset which has the specified minimum support, and it is the percentage of transactions containing the itemset.

2.1.1 Itemset

An itemset is a collection of one or more items that appear in a transaction database. The main objective of FIM is to find frequent itemsets that appear in a transactional database.

2.1.2 K-Itemset

An itemset that contains k items in its itemset, e.g., {A} – 1 itemset, {A, B, C} – 3 itemset, and so on. The total no. of possible items is 2^K, e.g., 3-itemset{X, Y, Z} = {∅, X, Y, Z, XY, XZ, YZ, XYZ}.

2.2 FIM

An itemset whose Support (S) $\geq$ min_Supp-Threshold.

2.2.1 Support Count (σ)

The frequency of an occurrence of an itemset.

2.2.2 Support (S)

The fraction of the transactions that contain an itemset (Eq. 1).

$$S = \frac{\sigma(\text{items})}{\text{no.of transactions}} = \frac{P(X \cup Y)}{T}, \text{X \& Y are items} \tag{1}$$

2.2.3 Support% (S %)

The percentage fraction of transactions that contain an itemset (Eq. 2).

$$S\% = \frac{(\sigma)}{T} * 100 \tag{2}$$

2.3 High-Utility Item/Pattern Mining (HUIM)

An itemset is said to be high-utility itemset if it is frequent item and possesses utility factors [6]. Simply, HUIM is an extension to FIM with Utility. A satisfaction factor that a consumer experiences from a product or service is called utility. It is difficult to measure a qualitative concept such as utility, but economists tried to quantify it to be simple; it is the profit, weight, or value associated with every itemset in the database. Simply, utility is a profit (or) support (or) confidence (or) value (or) weight. Utility of itemset (U) (Eq. 3) can be defined as a product of external utility (eu) which is the consequence of the distinct items and the internal utility (iu), which is the consequence of the items in transactions.

$$\text{Utility of itemset}(\text{U}) = \text{External utility}(\text{eu}) \times \text{Internal utility}(\text{iu}) \quad (3)$$

3 Application of HUIM in Research

HUIM algorithms can be implemented effectively when there are proper requirements and specifications available to perform the research. We need to know about few aspects before designing HUIM algorithm like the kind of the database used, the possible data structures, any novel methods for computation analysis, the nature of the application domain, and the possible smart techniques that can be applied to the data structures, so that we come up with a scalable, reliable, flexible, speed and an algorithm that consumes very less memory with minimal or one database scan (s).

3.1 Application of HUIM in Various Types of Databases

Numerous kinds of databases are used for data mining like relational, transactional, temporal, time series, spatial, multimedia, Web mining. Table 1 highlights the various databases which are extensively used for high-utility itemset mining.

3.2 Possible Data Structures and Novel Methods

Various approaches like top down, bottom up, both ways parallel, level wise, depth first, mapping with hashing or indexing, using ordered/unordered/labeled structures, pruning strategies can be applied on the data structures and novel methods as discussed.

3.2.1 Tree Structure

Many data structures are used for mining, the tree structure used avidly in HUIM which includes various parsing and pruning techniques (top down, bottom up, both ways parallel) [4]. Lexographic trees [12], suffix trees [24], pruning trees [53], and other strategies can be applied on trees for effective mining.

3.2.2 List Structure

The list structure which is an abstract data type that became popular in HUIM from [13] which used utility list which can generate HUIM without candidate generation.

3.2.3 Set Structure

This is also an abstract data type like list but does not allow duplicate values and more effective than list for key-based algorithms [29].

3.2.4 Lattice Structure

A lattice or a crystal structure can be used for N-dimensional data. The approaches are like level wise (similar to tree structures) [14].

3.2.5 Virtual Hyperlink Data

A kind of virtual machine introspection links various spaces more efficient than the index mapping. Hyperlink map can be set in the map layer of any structure [15].

3.2.6 Novel Structures in Machine Learning

Novel structures in machine learning are very adaptable to data mining concepts where the solutions are more realistic and optimal. The data is mapped into pre-defined classes and groups. In recent times, evolutionary algorithms [16] are emerging to work on high-utility itemsets which are machine learning genetic algorithms with binary string data structures. In genetic algorithm [54], HUI is generated from previous stage which has been considered as input and it would identify actual high-utility items using fitness function and threshold set by user.

3.2.7 Indexing and Hashing

Indexing is a data structure to retrieve records from database based on some attribute that needs indexing and hashing on the other side, a data structure which is used to index and retrieve items in a database using a hash function key. These are the methods which are used in mapping along with the above data structures, e.g., 1: set structure with indexing data structures is used in [29] for mining efficient TEP (Top-K erasable pattern) mining.

e.g., 2: GA with binary strings data structure used in multiobjective evolutionary approach [16] for mining frequent and high-utility itemsets. The same can be incrementally or recursively applied for effective and fast HUIM.

3.3 Application Domain Area

Various domains like education, medical, mobile, streaming, Web, market, stocks, finance are applying HUIM for effective outcomes. Listed Table 2 has few domains that applied HUIM.

3.4 Applying Techniques

Various techniques listed in Table 3 that can be applied along with the discussed databases and data structures for different domain applications.

4 Advancements in HUIM

HUIM is utilized on various datasets like transactional, time series, incremental and data streams, and the researches driven in these areas are highlighted in Table 4 of the chapter.

4.1 Transactional Databases

From expensive FP tree algorithm [55] without candidate generation, HUIM supports anti-monotone property [3] and improved by adding a mathematical model to the HUIM which used on rare itemsets by Yao [6]. Liu [56] introduced a two-phase algorithm for HUIM using weighted transaction closures in the first phase pruning and filtering high-utility itemsets and second phase with an extra database scan with common partitioning which can work in parallel on multiple processes challenging

Table 4 A Decade List of advancing algorithms. [Rec.—Recursive, Incre.—Incremental, Freq.—Frequent, Pat.—Patterns, ep.—Erasable Patterns, Vir.-Virtual, Evol. Evolutionary, Upb.—Upper Bound, alg.—algorithm]

Year	Authors	Algorithm	References	Year	Authors	Algorithm	References
2007	R. P. Gopalan	CTU Mine	[57]	2014	P. F. Viger	FHM (co-occurrence)	[58]
2007	C. S. K. Leung	CAN Tree (Freq. Pat.)	[37]	2014	Wei Song	Dynamic Pruning HUIM	[59]
2008	K. Cheung	Top-K patterns	[60]	2015	Unil Yun	Incre.HUPID Tree	[35]
2008	Y. Li	Isolated HUIM	[61]	2015	V. S. Tseng	Rec. UDHUP	[31]
2008	T. Liang	Temporal HUIM	[28]	2015	Ashish Gupta	FPPM (time series)	[24]
2008	T. P. Hong	Fast Incre. FPTree	[34]	2016	P. F. Viger	PHM (Periodic)	[19]
2009	C. F. Ahmed	HUIM Incre databases	[62]	2016	V. S. Tseng	TKU and TKO (Top-K)	[18]
2010	C. F. Ahmed	HUIM Data stream	[30]	2016	Unil Yun	Novel Upb. AvgUIM	[21]
2010	Hua Fu	MHUI-BT, MHUI-TID	[12]	2017	C. F. Ahmed	WPPM (time series)	[25]
2010	Jyoti Pillai	Temporal WHUIM	[52]	2017	Bay Vo	NFWI (N-list weight)	[11]
2010	V. S. Tseng	UP Growth	[63]	2017	Bay Vo	Lattice HUIL, HAR	[14]
2011	C. H. Weng	Fuzzy IM (education)	[7]	2017	P. F. Viger	SPHUIM (2 phase short)	[20]
2011	Ashish Gupta	Minimal Infreq HUIM	[49]	2017	Lei Zhang	MOEA alp [Evol.Alg]	[16]
2011	T. P. Hong	Effective HUIM	[64]	2017	Srinivas	HUPM Vir. HyperLink	[15]
2012	M. Liu	Utility List + HUI Miner	[13]	2017	Unil Yun	SinglePass EPM (ep.)	[32]
2013	V. S. Tseng	UP Growth +	[17]	2017	Bay Vo	TEP and TEPUS (ep.)	[29]

large databases with high scalability. Shankar [65] worked on novel algorithms which find low utility high frequency (LUHF) and other possibilities LULF, HULF, HUHF and is very scalable, but the time and memory consumption is more.

Tseng [63] improvised FP growth to an utility factor UP Growth and UP Growth + trees (2013) [17] and introduced Top-K TKU algorithm [18] for Top-K items which is close to optimal case, which reduced the candidate generation to a greater extent, and UP Growth tree generates candidate itemsets only with two database scans. Yet, it is complex due to its evaluation of tree structure, and Liu (2012) introduced utility list with HUIMiner which completely avoided candidate generation without trees. One of the economical models that is produced and used widely. Viger (2014) came up with FHM [58], strict patterns PHM [19] to flexible patterns with short two-phase SPHUIM [20] were popular performing periodic mining by filtering the non-periodic HUIs and is much faster than FHM and adopting flexible techniques for effective patterns. Unil Yun (2016) introduced novel algorithms with only two scans for building the DFS (depth-first) search-based process recursively and constructing list structures of itemsets with (k + 1) lengths from k-length list, and pruning uses tight upper bounds. BayVo (2017) coming with new trend of methods using weighted N-list structures [11], lattices [14] concentrating more on the semantics of high-utility itemsets to extract effective HUI's. HUIM using multiobjective evolutionary algorithms by Zhang [16] is new evolution in transactional databases which can reduce the time, increase speed, and reduce scan to an optimal extent using genetic algorithms.

4.2 Time Series Databases, Data Streams, Temporal and Incremental Databases

Time Series: J. Han et al. (1999) proposed efficient mining of partial periodic patterns (previous algorithms worked only with full patterns) in time series databases and used apriori property and max-subpattern hit property and shared mining of multiple periods. Mining partially would need only two scans over the time series database and efficient in mining long periodic patterns. This is limited only to categorical data with single level of abstraction. Later, **Chanda et al**. (**2015**) introduced an efficient approach to mine flexible periodic patterns in time series databases. It uses suffix tree for flexibility which simultaneously handles various starting positions throughout the sequences (previously proposed PFPM developed by Phillipe was not flexible). The processing time, scalability, and quality of mined patterns are performed very well using this algorithm. The running time is more and can be concise using a better framework. In 2017, again improvised with a new framework for mining weighted periodic patterns in time series databases. It is an effective pruning strategy with minimal time period. Weighted periodic pattern mining (WPPM) can effectively mine three types of weighted periodic pattern (single, partial, full) in a single run. Although it uses downward property, it can extend using more sharing properties to work more effectively.

Data streams: **Li et al**. (**2010**) discovered fast and memory-efficient mining of high-utility itemsets from data streams with and without negative item profits. Using the on-shelf mining of mining HUI with negative item profits (to know previous items) and also positive item profits on data streams (Previously positive items used LexTree-2HT), this algorithm provides better identified profit items and re-running is still a challenge. **Tseng et al**. (**2015**) proposed UDHUP-apriori which uses apriori-like approach to recursively derive UDHUPs and UDU-list structures without candidate generation also performs pruning, 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. **Vo et al**. (**2017**) proposed efficient algorithm for mining top rank K erasable patterns using pruning strategies and subsume concepts. A set data structure with indexing is used for TEP (top rank—KEPMining) and mines Top rank KEPs; unlike Plist used in erasable patterns, this algorithm uses a dynamic set structure (dPidset) to reduce the memory usage and a dynamic threshold pruning strategy to accelerate the mining process. Also uses index strategy to further speed up the mining time and reduce the memory usage yet works on sparse data.

Incremental: **Yun et al**. (**2014**) implemented incremental HUIM with static and dynamic databases. A high-utility patterns in incremental databases tree (HUPID Tree) uses single database scan, and a restructuring method with a novel data structure called tail-node information list (TIList) in order to process incremental databases more efficiently. It uses the effective HUI, but with weighted maximal itemsets, we can work more efficiently in distributed and dynamic databases. In 2017, improved more efficient algorithm for mining high-utility patterns from incremental databases where any structure takes a minimum of two scans and [66] this is discovered to perform in just one scan. This algorithm is unable to handle dynamic databases effectively due to more complexity. It needs a change in the pattern mining to handle dynamic and complex incremental databases, a step far ahead to handle the dynamic databases using single pass-based efficient erasable pattern mining using list data structure on dynamic incremental databases. Erasable pattern mining (EPM) which is applied on sparse data can also be extended to dense data streams.

5 Critical Analysis, Suggestions

C. F. Ahmed and T. P. Hong experimented HUIM using various databases (incremental [34, 62], data stream [30], time series [25]) by applying trees with inverse and weighted techniques, but the time stamp of incremental updates, and statistical distribution sampling is not absolute to a maximum extent. Tseng [63, 17] on the other side worked with transactional databases from UP Growth to data streams on recursive up-to-date HUIM [31] and finding Top-K TKU algorithm for efficient HUIM [18] and worked on pruning tree strategies and lists, but there is a lot of scope to work on utility mining tasks to discover different types of Top-k high-utility patterns such as Top-k high-utility episodes, closed, infrequent patterns, Top-k high-utility Web access, and mobile patterns. Viger [58, 19, 20] vigorously worked with HUIM based on periodic patterns, and they are strict and too long or too short; shorter constraint would allow utility measure to reduce no. of unnecessary patterns found and filters the most useful and minimal top patterns, but constraints have to still be effectively managed. Unil Yun and Bay Vo worked on dynamic databases [35], novel algorithms [21], and lattices [14]. Unil Yun exposed various new techniques using upper bounds for dynamic pruning; using a single upper bound may lead to weak bounds, and statistically approximate upper bounds are not as tight when applied for real data; Bay Vo used utility confidence framework to obtain semantic relationships among the high-utility itemsets. More integration is required in the HUI phase apart from its low runtime and memory consumption. Most of the algorithms either work on sparse data if they have the most efficient mining with respect to scanning, scalability, speed, and memory utilization factors. If they work on dense databases, they compromise on any one of the non-functional requirements. If the algorithm works best in dense as well as speed and memory, it lacks in dynamism. Handling complex, yet maintaining the empirical results with high utility is still a challenge. The recent come up is erasable patterns [29, 32, 67] with one pass algorithms. One pass takes input for only one time showing a O(1) and storing the data at the worst take O(n) or < O(n), and the time consumption is at the max O(n). The latest method being one pass algorithms using various data structures, its highly suggestible to work on more evolutionary algorithms with above combination of techniques (Sect. 3) to work upon dense datasets and take more challenge on novel ideas with optimal, one pass recursive methods attaining high performance and minimal memory usage to mine the utility patterns.

6 Conclusion

The research works contributed by various authors basing on HUIM are analyzed and presented in this chapter. Most of the previous research related to HUIM was confined only to transactional databases where the nature of data is certain, but we have applications where voluminous, dynamic, continuous data with high speeds, uncertainty, and database-centric mining are also in need of applying HUIM. In this chapter, we tried our level best to give review on various algorithms and their methodologies and covered variety of databases (data stream, temporal, time series, incremental, relational databases along with transactional databases) considering the functional (data nature, structure, relationships, etc.) and non-functional (scalability, performance, reliability, limitations, etc.) requirements of all kinds of databases and how effective HUIM can be done using the comparison techniques. The survey of the above research works ensures that most of the works were carried out and moving toward one pass techniques and single database scans. However, to face the challenges associated with one pass algorithms, potential or periodic HUIM based on data structures like tree, list along with novel methods, evolutionary algorithms were considered to be appropriate. As an extension to have robust identification of most utilized itemsets, one needs to develop algorithms that focus mainly on can trees along with novel structures and soft computing techniques to find an efficient periodic mining algorithm in interactive and constraint-based models which will be focused in our future works.

References

1. Tan, P.N., Steinbach, M., Kumar, V.: Introduction to Data Mining, pp. 1 (2009)
2. Foumier Viger, P.: An Introduction to High utility Mining. In: A blog by Philippe Fournier-Viger about data mining, data science, big data (2015)
3. Chan, R., Yang, Q., Shen, Y.-D, Mining high utility itemsets. In: Proceedings of the 3rd IEEE International Conference on Data Mining, pp. 19–26 (2003)
4. Bhattacharya, S., Dubey, D.: High Utility Itemset Mining. Int. J. Emerg. Technol. Adv. Eng. ISSN **2**(8), 2250–2459 (2012)
5. Agrawal, R., Imielinski, T., Swami, A.: Mining Association rules between sets of items in large database. In: ACM SIGMOID International Conference on Management of Data (1993)
6. Hamilton, H.J., Yao, H.: Mining itemset utilities from transactional database. Data Knowl. Eng. 88–96 (2006)
7. Weng, C.-H.: Mining fuzzy specific itemsets for educational data. Knowl. Based Syst. **24**(5), 697–708 (2011)
8. Cagliero, L., Garza, P.: Infrequent weighted itemset mining using frequent pattern growth. IEEE Trans. Knowl. Data Eng. 1–14 (2013)
9. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: Proceedings of the 20th International Conference on Very Large Data Bases (VLDB '94), pp. 487–499 (1994)
10. Wang, J., Han, J.: BIDE: efficient mining of frequent closed sequences.. In: The 20th International Conference on Data Engineering, pp. 79–90 (2004)
11. Bui, H., Vo, B., Nguyen, H., Nguyen, T.-A., Hong, T.P.: A weighted N-list-based method for mining frequent weighted itemsets. Expert Syst. Appl. (2017)
12. Li, H.-F., Huang, H.-Y., Lee, S.-Y.: Fast and memory efficient mining of high-utility itemsets from data streams: with and without negative item profits. Knowl. Inf. Syst. **28**(3), 495–592 (2010)
13. Liu, M., Qu, J.: Mining high utility itemsets without candidate generation. In: CIKM 2012 Proceedings of the 21at ACM Conference on Information and Knowledge Management, pp. 55–64 (2012)
14. Mai, T., Vo, B., Nguyen, L.T.T.: A lattice-based approach for mining high utility association rules. Inf. Sci. **399,** 81–97 (2017)
15. Srinivas, K.M.: HMiner: Efficiently mining high utility itemsets. Expert Syst. Appl. **90**, 168–183 (2017)
16. Zhang, L., Fu, G.L., Cheng, F., Qiu, J.F., Su, Y.: A multi-objective evolutionary approach for mining frequent and high utility itemsets. Appl. Soft Comput. (2017)
17. 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)
18. Tseng, V.S., Wu, C.-W., Viger, Yu, P.S.: Efficient algorithms for mining top-k high utility itemsets. IEEE Trans. Knowl. Data Eng. **28**(1), 54–67 (2016)
19. Fournier-Viger, P., Lin, J.C.-W., Duong, Q.-H., Dam, T.-L.: PHM: mining periodic high-utility itemsets. In: ICDM2016, Advances in Data Mining, Applications And Theoretical Aspects, pp. 64–79 (2016)
20. Lin, J.C.W., Zhang, J., Foumier Viger, P., Hong, T.P. Zhang, J.: A two-phase approach to mine short-period high-utility itemsets in transactional databases. Adv. Eng. Inf. **33**, 29–33 (2017)
21. Yun, U., Kim, D.: Mining of high average-utility itemsets using novel list structure and pruning strategy. Future Gener. Comput. Syst. **68**, 346–360 (2017)
22. Yao, H., Hamilton, H.J.: Mining itemset utilities from transaction databases. Data Knowl. Eng. **59**(3), 603–626 (2006)

23. Song, M., Sanguthevar, R.: A transaction mapping algorithm for frequent itemsets mining. IEEE Trans. Knowl. Data Eng. **18**, 472–481 (2006)
24. Chanda, A.K., Ahmed, C.F., Samiullah, M., Saha, S., Nishi, M.A.: An efficient approach to mine flexible periodic patterns in time series databases. Eng. Appl. Artif. intell. **44**, 46–63 (2015)
25. Chanda, A.K., Ahmed, C.F., Samiuallah, M., Leung, C.K.: A new framework for mining weighted periodic patterns in time series databases. Expert Syst. Appl. **79**, 207–224 (2017)
26. Han, J., Dong, G., Yin, Y.: Efficient mining of partial periodic patterns in time series databases. In: The 15th International Conference on Data Engineering, pp. 106–115 (1999)
27. Wang, L., Meng, J., Xu, P., Peng, K.: Mining temporal association rules with frequent itemsets trees. Appl. Soft Comput. (2017)
28. Liang, T., Liang, T.: An efficient algorithm for mining temporal high utility itemsets from data streams. J. Syst. Softw. **81**(7), 1105–1117 (2008)
29. Le, T., Vo, B., Baik, S.W.: Efficient algorithms for mining top-rank-k erasable patterns using pruning strategies and the subsume concept. Eng. Appl. Artif. Intell. **68**, 1–9 (2018)
30. Ahmed, C.F., Tanbeer, S.K., Jeong, B.S.: Mining regular patterns in datastreams. Database Syst. Adv. Appl. 399–413 (2010)
31. Tseng, V.S., Lin, J.C.W., Gan, W., Hong, T.P.: Efficient algorithms for mining up-to-date high-utility patterns. Adv. Eng. Inf. **29**(3) 648–661 (2015)
32. Lee, Gangin, Yun, Unil: Single-pass based efficient erasable pattern mining using list data structure on dynamic incremental databases. Future Gener. Comput. Syst. **80**, 12–28 (2018)
33. Ahmed, C.F., Tanbeer, S.K., Jeong, B.S.: Mining regular patterns in datastreams. IEEE Trans. Inf. Syst. **E91-D**(11), 2568–2577 (2008)
34. Hong, T.P., Lin, C.W., Wu, Y.L.: Incrementally fast updated frequent pattern trees. Expert Syst. Appl. **34**(2), 2424–2435 (2008)
35. Yun, Unil, Ryang, Heungm: Incremental high utility pattern mining with static and dynamic databases. Appl. Intell. **42**(2), 323–352 (2015)
36. Lin, C.W., Hong, T.P., Lan, G.C., Wong, J.W., Lin, W.-Y.: Incrementally mining high utility patterns based on pre-large concept. Appl. Intell. **40**(2), 343–357 (2014)
37. Leung, C.K.S., Khan, Q.I., Li, Z., Hoque, T.: Can tree: a canonical order tree for incremental frequent-pattern mining. Knowl. Inf. Syst. **11**(3), 287–311 (2007)
38. Leung, C.K.S., Khan, Q.I., Hoque, T.: Cantree: a tree structure for efficient incremental mining of frequent patterns. In: Proceedings of the ICDM 2005, pp 274–281. IEEE Computer Society Press, Los Alamitos, CA (2005)
39. Tzvetkov, P., Yan, X., Han, J.: TSP: mining top-k closed sequential patterns. KAIS **7**(4), 438–457 (2005)
40. Agrawal, R., Srikant, R.: Mining sequential patterns. In: The 11th International Conference on Data Engineering, pp. 3–14 (1995)
41. Uday Kiran, R., Krishna Reddy, P., Kitsuregawa, M.: Efficient discovery of periodic-frequent patterns in very large databases. J. Syst. Softw. **112**, 110–121 (2016)
42. Fournier Viger, P., Dam, T.-L., Li, K., Duong, Q.-H.: An efficient algorithm for mining top K On-shelf high utility itemsets. Knowl. Inf. Syst. **52**(3), 621–655 (2017)
43. Goyal, V., Surekha, A., Patel, D.: Efficient skyline itemset mining. In: Proceedigs of the Eighth International Conference on Computer Science & Software Engineering, pp. 119–24. ACM, (2015)
44. Lin, J., Gan, W., Fournier-Viger, P., Hong, T.P., Tseng, V.S.: Efficient algorithms for mining high-utility itemsets in uncertain databases. Knowl. Based Syst. **96**, 171–187 (2016)
45. Shie, B.-E., Hsiao, H.-F., Tseng, V.S., Philip S.Y.: Mining high utility mobile sequential patterns in mobile commerce environments. In: International Conference on Database systems for Advanced Applications, DASFAA 2011, pp. 224–238 (2011)
46. Pillai, J.: User centric approach to itemset utility mining in Market Basket Analysis. Int. J. Comput. Sci. Eng. **3**(1) (2011)

47. Raina, Mohit, Pandole, Deepak, Patil, Nayan, Patil, Sonal: Mining High utility itemsets of stock transactions. Int. J. Emerg. Trend Eng. Basic Sci. IJEEBS **2**(1), 594–597 (2015)
48. Lin, J.C.-W., Ren, S., Fournier-Viger, P., Hong, T.-P.: EHAUPM: Efficient High Average-Utility Pattern Mining with Tighter upper Bounds, vol. 5, pp. 12927—12940 (2017)
49. Gupta, A., Mittal, A., Bhattacharya, A.: Minimally infrequent itemset mining using pattern-growth paradigm and residual trees. In: COMAD'11 Proceedins of the 17th International Conference on Management of Data, vol. 13 (2011)
50. Yun, Unil, Lee, Gangin: Incremental mining of weighted maximal frequent itemsets from dynamic databases. Expert Syst. Appl. **54**(C), 304–327 (2016)
51. Burdick, D., Calimlim, M., Flannick, J., Gehrke, J., Yiu, T.: MAFIA: a maximal frequent itemset algorithm. IEEE Trans. Knowl. Data Eng. **17**(11), 1490–1504 (2005)
52. Pillai, J., Vyas, O.P., Soni, S., Muyeba, M.: A conceptual approach to temporal weighted item set utility mining. Int. J. Comput. Appl. (0975–8887) **1**(28), 55–60 (2010)
53. Song, W., Liu, Y., Li, J.: Mining high utility itemsets by dynamically pruning the tree structure. Appl. Intell. **40**(1), 29–43 (2014)
54. Saranya, A., Kerana Hanirex, D.: Extraction of High Utility Itemsets using Utility Pattern with Genetic Algorithm from OLTP System. Int. J. Recent Innov Trends Comput. Commun. **3**(3), 1326–1331 (2015)
55. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In: Procs of the 2000 ACM SIGMOD international conference on Management of data, pp. 1–12 (2000)
56. Liu, Y., Liao, W.H., Choudhary, A.: A two-phase algorithm for fast discovery of high utility itemsets. Adv. Knowl. Discov. Data Min. **35**(18), 689–695 (2005)
57. Erwin, A., Gopalan, R.P., Achuthan, N.R.: CTU-mine: an efficient highutility itemset mining algorithm using the pattern growth approach. In: The Seventh International Conference on Computer and Information Technology, pp. 71–76 (2007)
58. Fournier-Viger, P., Wu, C.W., Zida, S., Tseng, V.S.: Fhm: Faster high-utility itemset mining using estimated utility co-occurrence pruning. In: Andreasen, T., Christiansen, H., Cubero, J.-C., Raś, Z. (eds.) Foundations of Intelligent Systems, pp. 83–92. Springer, Berlin, Germany (2014)
59. Song, W., Liu, Y., Li, J.: Mining high utility itemsets by dynamically pruning the tree structure. Appl. Intell. **40**(1) 29–43 (2014)
60. Chuang, K., Huang, J., Chen, M.: Mining top—k frequent patterns in the presence of the memory constraint. VLDB J. **17**, 1321–1344 (2008)
61. Li, Y.: Isolated items discarding strategy for discovering high utility itemsets Isolated items discarding strategy for discovering high utility itemsets. Data Knowl. Eng. **64**(1), 198–217 (2008)
62. 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
63. Tseng, V.S., Wu, C.W., Shie, B.E., Yu, P.S.: UP-Growth: an efficient algorithm for high utility itemset mining. In: The International Conference on ACM SIGKDD, pp. 253–262 (2010)
64. Lin, C.-W., Hong, T.-P., Lu, W.-H.: An effective tree structure for mining high utility itemsets. Expert Syst. Appl. **38**(6), 7419–7424 (2011)
65. Shankar, S., Pursothaman, T., Jayanthi, S.: Novel algorithm for mining high utility itemsets. In: International Conference on Computing, Communication and Networking, St. Thomas, pp. 1–6 (2008)
66. Yun, U., Ryang, H.M., Lee, G., Fujita, H.: An efficient algorithm for mining high utility patterns from incremental databases with one database scan. Knowl. Based Syst. **124,** 88–206 (2017)
67. Huengmo, Ryang, Unil, Yun, Ho, R.K.: Fast algorithm for high utility pattern mining with the sum of item quantities. Intell. Data Anal. **20**(2), 395–415 (2016)