International Journal of Engineering in Computer Science

E-ISSN: 2663-3590
P-ISSN: 2663-3582
www.computersciencejournals.com/ijecs
IJECS 2025; 7(1): 15-18
Received: 01-11-2024
Accepted: 06-12-2024

Arkan A Ghaib
Department of Information Technologies, Management Technical College, Southern Technical University, Basrah, Iraq

Corresponding Author:
Arkan A Ghaib
Department of Information Technologies, Management Technical College, Southern Technical University, Basrah, Iraq

Unified search and return algorithm (USRA) for enhanced high utility pattern mining

Arkan A Ghaib

DOI: https://doi.org/10.33545/26633582.2025.v7.i1a.150

Abstract
As data scales in size and complexity, traditional algorithms for pattern mining face challenges in efficiently managing both searches and return operations. This paper introduces the Unified Search and Return Algorithm (USRA), a novel approach that integrates both functionalities into a cohesive framework. By leveraging hybrid data structures and dynamic pruning techniques, USRA achieves significant improvements in runtime and memory efficiency. The proposed algorithm is evaluated against state-of-the-art methods on real-world and synthetic datasets, demonstrating superior performance in terms of scalability and resource utilization. These results establish USRA as a robust solution for high utility pattern mining in dynamic databases.

Keywords: Data mining, high utility pattern, pattern mining, itemsets mining

1. Introduction
High utility pattern mining (HUPM) has become an important family of methods for discovering informative patterns in transaction databases, incorporating the frequency and utility of items into consideration. Unlike the general frequent pattern mining approaches, which only latches on to the occurrence of the item, HUPM also takes the importance of each item into account, thus making it an excellent choice for non-binary and large scale datasets [1].

Previous studies in HUPM have investigated several approaches to improve efficiency and scalability. For static databases, tree-based and list-based structures were proposed by the UP-Growth and HUI-Miner algorithms [2] respectively to reduce the number of candidates and the number of scans of the database. For dynamic environments, d2HUP [3] and HUP-Miner [4] proposed with efficient pruning techniques along with utility-list data structure to enhance performance. But those methods don't work so well with data that is time-sensitive, where newer transactions are more relevant than older transactions.

To tackle the difficulties presented by time-decaying databases, damped window models have been introduced. In these models, outdated transactions contribute to the score; however, their significance diminishes over time through a decay factor, thereby assigning more weight to more recent data. The DHUPL algorithm [5] which used a list-based structure to mining high utility pattern for incremental environment without generating the candidate pattern and also required lesser space to store tables compared to other high utility pattern miners. These approaches are promising but are missing search & return capabilities integrated within, which is a must-have in any data-driven applications today.

Building upon these advances, we present the Unified Search and Return Algorithm (USRA). USRA abstracts the utility patterns found in the dataset with a high utility pattern mining approach while presenting a fast search mechanism based on different indexed list structure with the help of dynamic pruning to seamlessly process large and dynamic datasets.

This paper has the following contributions:

It consists of a joint search and return operations, merging the potential efficient mining and adaptable query.

Indexed lists and damped utility models as a hybrid data structure for scalability

Extensive evaluations showing marked improvements over state-of-the-art approaches.

2. Related Work

High utility pattern mining (HUPM) [6] has received considerable research effort in recent years, which aims to address the limitations of conventional frequent pattern mining by considering utility measures to capture patterns of high importance. Several techniques have been proposed to improve the performance of HUPM over time, targeting static and dynamic databases.

Liu *et al*. (2005) [7] proposed an algorithm called Two-Phase based on transaction-weighted utility (TWU), which pruned low-utility patterns from the beginning of the mining process, thereby alleviating the mining overhead. Al-Hamodi *et al*. Among them, Two-Phase is improved by EFP-Growth (2016) [8], MRFP-Growth (2016) [9] and EN-list [10] as it adopts a tree-based tree structure with even better pruning strategies to reduce the number of candidate patterns generated. Fournier *et al*. (2015) [11] proposed the Incremental High Utility Pattern (IHUP) mining algorithm that maintained global utility lists for dynamically processing incremental updates of dynamic datasets. Lin *et al*. (2016) [3], which described d2HUP using a more efficient, advanced method of pruning to lessen recomputation when data is given to it to scale to larger datasets. Zhao *et al*. That was LIHUP [12] algorithm, which effectively promotes utility patterns to be updated in a list-based fashion with considerably little memory and runtime. Le *et al*. (2019) [5] Presented Damped High Utility Pattern mining with List structure (DHUPL) based on using the damped window model for more up-to-date transactions should be considered with decay factors for their corresponding periods, to ensure they are relevant in time-sensitive databases. This approach generated no candidates and showed improved run time and memory usage. Kim *et al*. (2023) proposed an efficient high utility occupancy pattern mining algorithm called HUOMIL [13] that utilized indexed list structures to lower computational costs and speed up pattern extension. Enhancing the efficiency of mining Distributed Utility itemsets for big data (IDUIM) [14]. The proposed technique is an enhancement of the Distributed Utility item sets Mining (DUIM) [15] algorithm which added several enhancements. the second improved prepost algorithm that is HPrepost, was the image of prepost, introduced with the standard of Mapreduce programming model. Efficient mining of frequent itemsets with reduced runtime and space.

This was effective but did not have integrated search and return functionalities. However, the majority of these algorithms are standalone, mining or search only, and cannot be integrated into a single energy-efficient framework. The USRA (Unified search and return Algorithm), extends this approach by implementing both AF, USRA algorithm builds on hybrid indices and dampened utility models to overcome these limitations of past work, delivering a stable fit for large, dynamic, and high-dimensional datasets.

3. Preliminaries

3.1 High Utility Pattern Mining

High utility pattern mining (HUPM) aims to extract all itemsets from a transactional database that produce high utility. Every transaction has two essential item attributes: Internal utility, which tells you how much of an item is there in a transaction, and external utility, which indicates how valuable/profitable an item is. The utility of an item would be the internal utility multiplied by the external utility. Rather, for a specific itemset or pattern its utility is calculated as a sum of the utilities of all the items in the considered itemset in the qualified transactions. A pattern is classified as a high utility pattern (HUP) if its total utility exceeds some user-defined minimum utility threshold. HUPM is commonly employed in various applications, including retail market basket analysis, customer behavior modeling and inventory management.

3.2 Damped Utility Model

Dynamic databases often have cases where newer transactions are more significant than older transactions. The damped utility model considers the age of transactions and applies a decay factor (with) to its utility. This reduces the impact of older transactions on the utility calculation, highlighting the significance of recent information. Damped utility of a transaction with age is calculated as:

where is the transaction's initial utility and is the time elapsed since the transaction occurred. The decay factor determines how quickly the significance of older transactions drops off, with smaller values of resulting in a steep drop of the significance relative to older transactions and values closer to 1 making the decay more gradual. The damped utility model can affect a steady equilibrium between how potent historical relevance is and how trendy it is, hence it fits most dynamic and real -time surroundings.

3.3 Global List Data Structure

The global list data structure is a simple yet powerful data structure for storing and managing important information about items and patterns. For every global list, each entry contains Transaction ID (TID): Provides a reference to which transaction the item belongs \ Utility: The utility of the item found at position i in the transaction Remaining Utility: The remaining utility of items after position i in the transaction with respect to their sort order Damped Utility: the utility of the transaction after applying the decay factor This structure map for a global list allows very fast mining of patterns from the data. It also provides incremental updates by adapting new transactions and updating existing ones based on the effect of the decay factor. The anti-monotonic property of utility — that is, any subset of a low-utility patter cannot itself be a high-utility pattern — allows the global list structure to avoid redundant calculations.

3.4 The Proposed Data Structure (USRA) for Mining High Utility Patterns

The new global lists are then used with the Unified Search and Return Algorithm 1 (USRA) to efficiently mine for high utility patterns.

Steps

1. Candidate Generation

Global lists are used to identify candidate patterns.

Prune low-utility patterns at an early stage using the anti-monotone property.

2. Pattern Extension

Pattern are extended recursively by concatenating the entries in the global lists.

For every extended pattern, calculate utilities and damped utilities.

3. Pruning
With the remaining utility, prune the pattern that is not being able to achieve the utility threshold.

4. Query Handling
Real-time user query processing through the indexed format of the global lists.
Return the patterns that match the query criteria.

Algorithm 1: SARA Pseudocode
Input: D; Original transactional database ΔD: Incremental transactions δ: Minimum utility threshold **Output** : High-Utility Increment H(ΔD) (HIPs) Output: High Utility Patterns D: → Construct global lists (GILS) The first step is to scan D to compute utility and damped utility values: b. Set GILS to all items that are in D. d. Insert TID, utility, remaining utility and damped utility into GILS. D: Append ΔD to GILS a. There are new transactions in ΔD b. Use decay factor f to decay existing utilities. c. The GILS will be updated with new transaction entries. d. Remove outdated entries whose utility < δ. Mine high utility patterns: a. From GILS, generate candidate patterns. b. Recursively expand patterns using GILS. Prune patterns using remaining utility upper bounds. Process user queries: a. Patterns in GILS that match query criteria. b. Return refined results. Mine the HUPs mining pattern over the utility threshold δ.

4. Experimental Results
4.1 Features of the Real Datasets
To test the proposed method, we carried out experiments on three datasets from the real world, with different characteristics.

Table 1: Characteristics of the Authentic Datasets

Dataset	Number of Transactions	Number of Items	Average Transaction Length	Domain
Retail	88,162	16,470	10.3	Retail Market
Kosarak	990,002	41,270	8.1	Online Clicks
Chainstore	1,112,949	46,086	7.2	Grocery Sales

4.2 Experimental Setup and Datasets
The experiments were done in a controlled setting:
Hardware generic specification: Intel Core i7 (3.6 Ghz), 16 GB RAM, Windows 10.
Implementation of the data is done through a simple write up in python, along with your data analysis
The proposed method (USRA) was compared against state-of-the-art algorithms as follows:
DHUPL: List structure based High Utility Pattern Mining.
Using Indexed Lists to Mine High Utility Occupancies (HUOMIL).

Table 2: Algorithm Comparison in Terms of Time and Memory

Dataset	Metric	DHUPL	HUOMIL	USRA
Retail	Runtime (sec)	35.2	29.8	18.6
	Memory (MB)	320	290	200
Kosarak	Runtime (sec)	89.4	74.3	50.2
	Memory (MB)	480	410	260
Chainstore	Runtime (sec)	112.5	96.7	62.8
	Memory (MB)	620	550	340

4.3 Experimental Results with Different Data Sizes
Scalability was assessed through personal step-wise data expansion for each data set.

Table 3: Results with Increasing Data Size

Dataset	Data Size (%)	Runtime (sec)	Memory Usage (MB)
Retail	50	9.2	100
	100	18.6	200
Kosarak	50	25.1	130
	100	50.2	260
Chainstore	50	31.4	170
	100	62.8	340

5. Discussion
As shown, the proposed method (USRA) significantly outperforms the baseline algorithms in either runtime or memory efficiency. USRA works great on large scale datasets because damped window model fits the very large data well and the global list structure supports handdling large data. Additionally, its capability to update the global lists on-the-fly means flexibility when data changes.

6. Conclusion
In this paper, we introduced a list-based incremental high utility pattern mining mechanism using the damped window model. The method we propose, USRA, combines global, advanced list data structures with a decay mechanism that allows for prioritization of more recent data while still keeping older data in a readily accessible location. The experimental results show substantial runtime and memory usage reductions over the existing algorithms. In the future, the work will extend to the method in distributed environments and the study of this method in various areas of applications (for example, healthcare and IoT systems).

7. References

1. Gan W, Lin CW, Fournier-Viger P, Chao HC, Tseng V, Yu P. A survey of utility-oriented pattern mining. IEEE Trans Knowl Data Eng. 2019;31(12):2259-2273. doi: 10.1109/TKDE.2019.2942594.
2. Liu M, Qu J. Mining high utility itemsets without candidate generation. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM); 2012 Oct 29 - Nov 2; Maui, Hawaii, USA. New York: ACM; 2012. p. 55-64.
3. Liu J, Wang K, Fung BCM. Mining high utility patterns in one phase without generating candidates. IEEE Trans Knowl Data Eng. 2016;28(5):1245-1257.

4. Krishnamoorthy S. Pruning strategies for mining high utility itemsets. Expert Syst Appl. 2015;42(5):2371-2381.
5. Nam H, Yun U, Vo B, Truong T, Deng ZH, Yoon E. Efficient approach for damped window-based high utility pattern mining with list structure. IEEE Access. 2020;8:50958-50968.
6. Krishnamoorthy S. Pruning strategies for mining high utility itemsets. Expert Syst Appl. 2015;42(5):2371-2381.
7. Liu Y, Liao WK, Choudhary A. A two-phase algorithm for fast discovery of high utility itemsets. In: Advances in Knowledge Discovery and Data Mining: 9th Pacific-Asia Conference, PAKDD 2005; May 18-20, 2005; Hanoi, Vietnam. Berlin: Springer; 2005. p. 689-695.
8. Al-Hamodi AA, Lu SF. MapReduce Frequent Itemsets for Mining Association Rules. In: 2016 International Conference on Information System and Artificial Intelligence (ISAI); 2016 Jun 13-16; Hangzhou, China. IEEE; 2016. p. 281-284.
9. Al-Hamodi AA, Lu S. MRFP: Discovery frequent patterns using MapReduce frequent pattern growth. In: 2016 International Conference on Network and Information Systems for Computers (ICNISC); 2016 Apr 14-16; Gwangju, Korea. IEEE; 2016. p. 298-301.
10. Ghaib AA, Nahi AA. Enhancing N-List Structure and Performance for Efficient Large Dataset Analysis. 2024.
11. Fournier-Viger P, Lin J-CW, Gueniche T, Barhate P. Efficient incremental high utility itemset mining. In: Proceedings of the 5th International Conference on Advances in Social Media, Big Data, and Information Systems (ASE Big Data); 2015 Oct 7-9; Montreal, Canada. p. 53.
12. Yun U, Ryang H, Lee G, Fujita H. An efficient algorithm for mining high utility patterns from incremental databases with one database scan. Knowl-Based Syst. 2017;124:188-206.
13. Kim H, Ryu T, Lee C, Kim S, Vo B, Lin JCW, *et al*. Efficient method for mining high utility occupancy patterns based on indexed list structure. IEEE Access. 2023;11:43140-43158.
14. Ghaib AA, Alsalhi YEA, Hayder IM, Younis HA, Nahi AA. Improving the Efficiency of Distributed Utility Item Sets Mining in Relation to Big Data. J Comput Sci Technol Stud. 2023;5(4):122-131.
15. Al-Hamodi AA, Lu S. A novel approach for fast mining frequent itemsets using N-list structure based on MapReduce. arXiv preprint arXiv:1704.04599. 2017 Apr 15.