A Boolean-Pattern Based Approach for High Utility Itemset Mining with Negative Unit Profits under Length Constraints

Ye-In Chang[1], Chia-En Li[2*], Yi-Jun Wang[1]

[1]*Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung, Taiwan*
[2]*Department of Information Management, Cheng Shiu University, Kaohsiung, Taiwan*
**lichiaen@gmail.com*

Keywords: High Utility Itemset Mining, Negative Unit Profits, Boolean Pattern List, Bitwise Operations, Pruning Strategy

Abstract

High utility itemset mining is a critical technique for identifying itemsets that generate significant value by considering both quantity and unit profit. Yet, it often faces challenges in real-world scenarios where items carry negative unit profits due to discounts or promotions. Existing methods for handling negative profits and length constraints often rely on loose upper-bound estimates and repeated database scans, resulting in high computational overhead. This study presents a novel algorithm that efficiently mines high utility itemsets with negative unit profits under length constraints. The proposed approach utilizes a Boolean-pattern list structure that represents transactions as bit sequences, enabling rapid processing through bitwise operations. A primary contribution of this work is the development of a new upper-bound measure, the Negative Transaction Weighted Utility, which accounts for both positive and negative values to provide tighter bounds and more effective pruning of the search space. Experimental evaluations on both synthetic and real-world datasets indicate that this method significantly reduces the number of generated candidates and improves execution time compared to state-of-the-art algorithms. This approach offers a practical, scalable solution for advanced market-basket analysis in complex retail environments.

1 Introduction

High Utility Itemset Mining (HUIM) aims to identify itemsets in a transaction database whose utility values exceed a user-defined threshold, referred to as High Utility Itemsets (HUIs). Traditional HUIM methods generally assume static, positive item profits across all transactions; however, real-world retail environments are more complex, as item profits may vary by transaction and can even be negative due to discounts, promotions, or bundled pricing strategies. To better reflect practical scenarios, recent research has extended HUIM to support non-static and negative unit profits, enabling more realistic modeling of transaction data. HUIM has important applications in retail analytics, including hypermarkets, supermarkets, and convenience stores, where it can guide marketing and merchandising decisions. By analyzing transaction databases, retailers can identify profitable item combinations and optimize product placement. For example, items such as cereal and milk are frequently purchased together, and placing them in proximity can increase joint sales. HUIM provides a systematic framework for discovering such patterns, ultimately supporting revenue growth and more effective business strategies [1, 5].

The rest of the paper is organized as follows. Section 2 surveys existing well-known High Utility Itemset Mining (HUIM) algorithms in the static profit database and the non-static profit database. Section 3 presents the proposed boolean-pattern list based algorithm. Section 4 compares our algorithm and the EHNL algorithm. Finally, we give a conclusion and point out the future work.

2. Related Work

High Utility Itemset Mining (HUIM) was proposed to consider both the unit profit and quantity of items to discover High Utility Itemsets (HUIs) in a database. However, many traditional HUIM algorithms assume that databases contain only items with positive profits [2, 4, 6, 7]. Consequently, these algorithms cannot be applied to databases with negative item profits, as their pruning strategies fail when encountering negative values. To address this, algorithms such as FHN [1] and GHUM [3] were proposed to mine HUIs in databases containing negative unit profits. However, these methods overlook length constraints. Without length limitations, long itemsets can easily qualify as HUIs, leading to the generation of an excessive number of candidate itemsets. To resolve this, the EHNL algorithm [5] was proposed, which introduces length constraints to restrict the minimum and maximum lengths of the mined itemsets. Furthermore, the aforementioned algorithms [1, 3, 5] utilize the Positive Transaction Weighted Utility (PTWU) as an upper bound to prune candidates, thereby reducing the search space and processing time. However, the PTWU calculation considers only items with positive profits, completely ignoring negative items. By incorporating the utility values of negative items, this upper bound can be tightened significantly. Therefore, to improve upon the efficiency of the EHNL algorithm, this study proposes a novel list-based algorithm to mine HUIs with both positive and negative profits efficiently.

Our approach introduces a Boolean-pattern list to compress database information. This boolean structure enables the use of rapid bitwise AND operations to efficiently discover size-2 itemsets. Additionally, we explicitly utilize negative profits to establish a tighter upper bound, further reducing candidate generation. We also explore the impact of sorting input transactions by presenting two versions of our approach: a basic algorithm that requires only a single database scan without sorting, and an improved version that incorporates sorting to optimize performance. Extensive experimental evaluations demonstrate that our proposed algorithm significantly outperforms the EHNL algorithm in efficiently mining HUIs with positive and negative items.

3 The Boolean-Pattern-List-Based Algorithm

This section presents the proposed algorithm, specifically designed to efficiently mine High Utility Itemsets (HUIs) in databases with both positive and negative unit profits. To overcome the computational limitations and memory overhead of existing methods, our approach introduces a compressed data structure alongside an enhanced pruning strategy.

The example database DB is shown in Table 1. The database DB consists of transactions, which are denoted as DB = $\{T_1, T_2, T_3, \ldots, T_n\}$, where n is equal to the number of transactions. The set of m distinct items is denoted as $I = \{i_1, i_2, i_3, \cdots, i_m\}$. Each item i_j is assigned a profit, which is shown in Table 2. A unique identifier TID is assigned to each transaction T_j, which includes a set of items from the set I ($T_j \subseteq I$). Moreover, every item has a quantity value in a transaction T_j.

Table 1 An example database with transaction data DB

TID	Items	Quantities
T_1	*a, b, c, f*	3, 2, 1, 1
T_2	*b, c, d*	3, 2, 3
T_3	*b, d, e, f*	3, 4, 2, 2
T_4	*a, d*	4, 2
T_5	*a, b, d*	5, 2, 1
T_6	*e, f, g*	2, 2, 1
T_7	*a, c, e, f*	6, 3, 2, 1

Table 2 Profits of items in database DB

Item	*a*	*b*	*c*	*d*	*e*	*f*	*g*
Profit	2	-2	4	5	-3	8	2

3.1 The Structure

The proposed algorithm employs a Boolean-pattern list to compress and store transaction data. Unlike traditional data structures, the list represents transaction itemsets as Boolean bit sequences (0s and 1s). This compact representation significantly reduces both memory consumption and the time required to traverse itemsets across transactions. Furthermore, this Boolean architecture enables the algorithm to leverage rapid bitwise AND operations, drastically accelerating the discovery and evaluation of size-2 itemsets.

3.2 The Pruning Strategy

To efficiently reduce the search space, we employ two distinct pruning strategies to filter out unpromising candidate itemsets based on their respective profit types:

- Case 1: Positive Transaction Weighted Utility (PTWU Pruning). For itemsets consisting entirely of items with positive unit profits, we utilize the PTWU. If the PTWU(X) of an itemset X is less than the user-defined minimum utility threshold (min_UT), then X and all of its supersets cannot be High Utility Itemsets (HUIs) and are safely pruned from the search space.
- Case 2: Negative Transaction Weighted Utility (NTWU Pruning). For itemsets containing items with negative unit profits, we introduce the NTWU. Unlike the redefined Transaction Weighted Utility (TWU), which simply ignores negative values, NTWU(Y) explicitly incorporates these negative profits into the calculation. Consequently, NTWU establishes a tighter upper bound.

3.3 The Mining Process

The execution of the proposed algorithm consists of sequential data insertion and search phases. During data insertion, the algorithm first determines the optimal bit length required to represent transaction itemsets based on the provided profit table. Subsequently, transactions are sequentially inserted into the list. Notably, the basic version of our algorithm requires only a single database scan, entirely bypassing the need for initial sorting operations. The complete mining workflow is illustrated in Fig. 1.

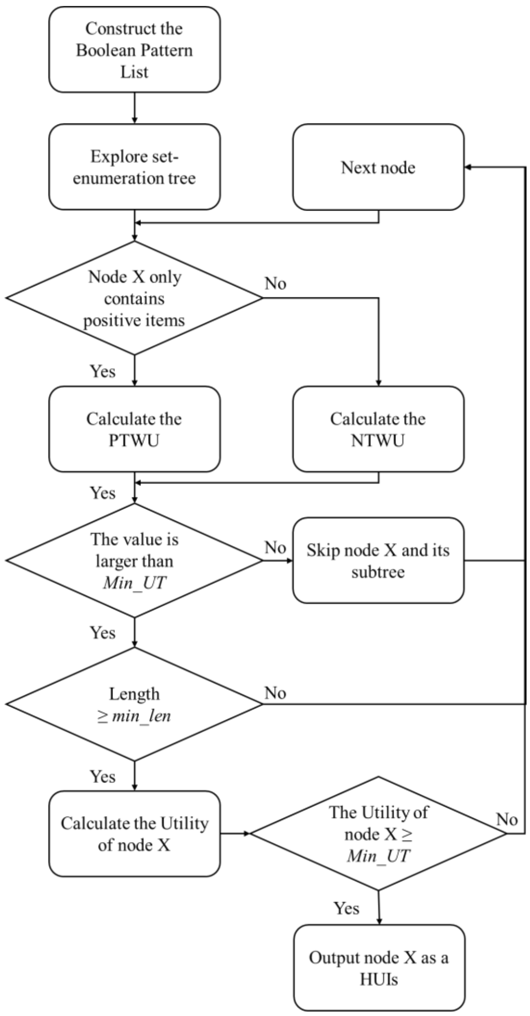

Fig. 1 The flowchart of the mining process

4 Performance

We use a real dataset, chess, to compare the performance of the proposed Boolean-Pattern-List-Based Algorithm and the EHNL algorithm. The results are shown in Fig 2 and Fig 3.

Fig. 2 A comparison of the number of candidates of the chess database under different *Min_UT* values

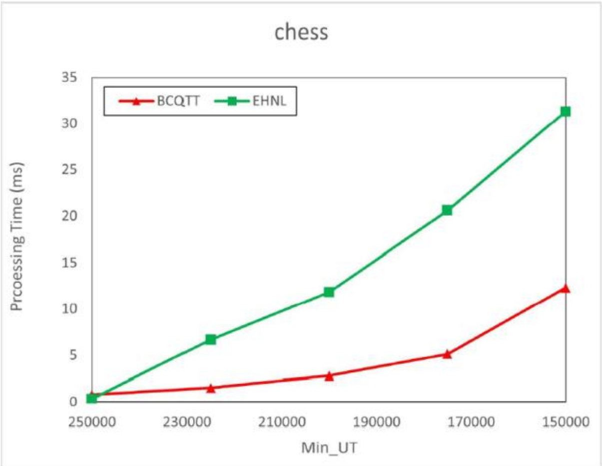

Fig. 3 A comparison of the processing time of the chess database under different *Min_UT* values

In Fig. 2, we compare the number of candidate patterns generated by the Boolean-Pattern-List-Based Algorithm and the EHNL algorithm across six threshold values using the chess dataset. As the *Min_UT* value decreases, the number of candidate patterns in both algorithms also increases. However, for each of the six threshold values, the Boolean-Pattern-List-Based Algorithm consistently produces fewer candidate patterns than the EHNL algorithm. Additionally, as the *Min_UT* value becomes smaller, the gap between the number of candidates from the two algorithms widens.

In Fig. 3, we present the processing times of our algorithm, Boolean-Pattern-List-Based, and the EHNL algorithm across six different threshold values in the chess dataset. The time performance gradually increases as the *Min_UT* value becomes smaller in both our algorithm and the EHNL algorithm. However, as the *Min_UT* value continues to decrease, the gap in the number of candidates also grows, and as a result, the processing time of our algorithm becomes shorter than that of the EHNL algorithm.

5 Conclusion

In this study, we proposed an efficient Boolean-Pattern-List-Based algorithm for mining high utility itemsets with positive and negative profits. By utilizing Boolean patterns to store transaction data, the proposed structure significantly accelerates the itemset search process. Additionally, our pruning strategy leverages the NTWU measure to explicitly account for negative items, leading to a substantial decrease in candidate generation. As demonstrated, our approach successfully mines high utility itemsets more efficiently than the EHNL algorithm in terms of processing time and candidate reduction. For future work, we plan to extend this approach to incremental databases containing items with negative utilities, further enhancing its applicability to dynamic, real-world scenarios.

6 References

[1] Jerry Chun-Wei Lin, P. Fournier-Viger, and W. Gana, "FHN: An Efficient Algorithm for Mining High-utility Itemsets with Negative Unit Profits," Knowledge-Based Systems, 2016, 111, pp. 283–298

[2] S. Krishnamoorthy, "HMiner: Efficiently Mining High Utility Itemsets," Expert Systems with Applications, 2017, 90, pp. 168–183

[3] S. Krishnamoorthy, "Efficiently Mining High Utility Itemsets with Negative Unit Profits," Knowledge-Based Systems, 2018, 145, pp. 1–14

[4] J. F. Qu, P. Fournier-Viger, M. Liu, B. Hang, and F. Wang, "Mining High Utility Itemsets Using Extended Chain Structure and Utility Machine," Knowledge-Based Systems, 2020, 208, pp. 1–14

[5] K. Singh, A. Kumar, S. S. Singh, H. K. Shakya, and B. Biswas, "EHNL: An Efficient Algorithm for Mining High Utility Itemsets with Negative Utility Value and Length Constraints," Information Sciences, 2019, 484, pp. 40–70

[6] T. Wei, B. Wang, Y. Zhang, K. Hu, Y. Yao, and H. Liu, "FCHUIM: Efficient Frequent and Closed High-utility Itemsets Mining," IEEE Access, 2020, 8, pp. 109928–109939

[7] U. Yun, H. Nam, G. Lee, and E. Yoon, "Efficient Approach for Incremental High Utility Pattern Mining with Indexed List Structure," Future Generation Computer Systems, 2019, 95, pp. 221–239