

Article

# Ignoring Internal Utilities in High-Utility Itemset Mining

Damla Oguz 

Department of Computer Engineering, Izmir Institute of Technology, Izmir 35430, Turkey; damlaoguz@iyte.edu.tr

**Abstract:** High-utility itemset mining discovers a set of items that are sold together and have utility values higher than a given minimum utility threshold. The utilities of these itemsets are calculated by considering their internal and external utility values, which correspond, respectively, to the quantity sold of each item in each transaction and profit units. Therefore, internal and external utilities have symmetric effects on deciding whether an itemset is high-utility. The symmetric contributions of both utilities cause two major related challenges. First, itemsets with low external utility values can easily exceed the minimum utility threshold if they are sold extensively. In this case, such itemsets can be found more efficiently using frequent itemset mining. Second, a large number of high-utility itemsets are generated, which can result in interesting or important high-utility itemsets that are overlooked. This study presents an asymmetric approach in which the internal utility values are ignored when finding high-utility itemsets with high external utility values. The experimental results of two real datasets reveal that the external utility values have fundamental effects on the high-utility itemsets. The results of this study also show that this effect tends to increase for high values of the minimum utility threshold. Moreover, the proposed approach reduces the execution time.

**Keywords:** data mining; itemset mining; high-utility itemset mining



**Citation:** Oguz, D. Ignoring Internal Utilities in High-Utility Itemset Mining. *Symmetry* **2022**, *14*, 2339. <https://doi.org/10.3390/sym14112339>

Academic Editors: Tzu Chuen Lu, Wun-She Yap and Biswapati Jana

Received: 25 September 2022

Accepted: 1 November 2022

Published: 7 November 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Digital tools in our daily lives create large amounts of data. Large amounts of data by themselves are not useful; therefore, they have to be mined to extract data that will be used in various aspects. One common way to extract information is via association rule mining [1], which is made up of the following two steps: the frequent patterns are found in the first step, and strong rules among these patterns are generated in the second step. When the patterns are itemsets in a transactional dataset, the first step is known as frequent itemset mining (FIM). A popular example is when the associations between the items are discovered in market basket analysis. For example, {*milk*, *bread*} is called a frequent itemset if *milk* and *bread* appear together at least as many times as a defined number of transactions. This knowledge can be used in the market basket analysis with different strategies, such as sales promotion, positioning of items, or cross-selling. Generating the rules among the frequent itemsets is straightforward [2]. For this reason, many researchers have reported on frequent itemset mining over the past 25 years [3–5].

About a decade after the proposal of FIM, high-utility itemset mining (HUIM) has been proposed to find itemsets, which have high utilities. A utility function is employed in HUIM to decide if an itemset has high utility or not [6,7]. For the market analysis, a typical utility function considers both the internal utility values, which are the quantities of the items, and the external utility values, which are the profits gained from the items. HUIM rests on the idea that an itemset can be infrequent due to the high prices of its items, but regardless, it can provide a high profit, such as the itemset {*caviar*, *champagne*}. HUIM is more difficult than FIM since there are many candidate itemsets to be evaluated against the utility function and high-utility itemsets dispersedly exist in the search space [8]. There are many studies about HUIM that generally aim to reduce the execution time or search space [9–18].

Due to the symmetric effect of internal and external utilities of items on the utility function, it can be said that some itemsets have high utilities just because of their high

internal utility values. However, rather than HUIM, these itemsets may be found by using FIM algorithms, which utilize the anti-monotone a priori property. From a different perspective, there is also another important challenge in HUIM. Typical HUIM algorithms find many high-utility itemsets; however, it would not be plausible to use all of them for marketing strategies. Finding the itemsets sold together with high external utility values could be used more beneficially since they represent much smaller sets but their utility values come from the profits of the items. For this reason, this study proposes an approach that discovers high-utility itemsets by ignoring the internal utility values of items to find itemsets with high external utilities that satisfy the minimum utility threshold. The objectives of this paper are two-fold: (i) to analyze the effects of external utility values in HUIM, (ii) to discard the quantities of items in each transaction to discover high-utility itemsets due to their high profits. We evaluated our proposal on two real datasets with various minimum utility thresholds. Experimental results show that our proposal is effective and has a shorter execution time than HUIM.

The rest of the paper is structured as follows. The next section briefly presents preliminaries and a review of the related work on high-utility itemset mining. Section 3 explains the proposed approach, which is called high-utility itemset mining without internal utilities (HUIM-WOIU). Section 4 details the experiments and discusses the analysis of the proposed approach. The paper concludes with Section 5.

## 2. Preliminaries and Related Work

We will detail the terminology before discussing the details of existing and proposed approaches. Frequent patterns are called *frequent itemsets* when they exist in a transactional dataset. To become a frequent itemset, the set of items must appear together as more than a threshold, and *support* is the typical measure to decide if an itemset is frequent or not. The support of an itemset  $\{a, b\}$  in a transactional dataset  $D$  refers to the percentage of transactions that contain both  $a$  and  $b$  together as shown in Equation (1), where  $P$  denotes the probability. Basically, it is the percentage of the transactions in which the itemsets appear together to all transactions. The *support count* in Equation (2) is another metric in FIM, which is also known as the occurrence frequency of an itemset.

$$\text{support}(\{a, b\}) = P(a \cup b) \quad (1)$$

$$\text{support\_count}(\{a, b\}) = \text{support\_count}(a \cup b) \quad (2)$$

*Support* represents a percentage while the *support count* is a number; however, both of them are used to evaluate the frequency of an itemset. An itemset is a frequent itemset if its support count satisfies a defined *minimum support count threshold* (i.e., the support of the itemset satisfies the corresponding *minimum support threshold*) [2]. FIM [1] is commonly used in the market basket analysis to find the itemsets that are purchased together. It is studied extensively and several algorithms exist in the literature. a priori [19], Eclat [20], and FP-Growth [21] are three important frequent itemset mining algorithms.

Although FIM is useful, high-utility itemset mining [7] can also be applied to quantitative transaction databases. Let  $D = \{T_1, T_2, \dots, T_m\}$  denote a set of transactions, which is called a *transaction dataset*, denoted by  $D$ . Similarly, a *quantitative database*  $D_q$  is also a set of transactions, but transactions are enriched by the item's quantity and weight, which imply the item's relative importance. For each item  $i$  in the set of items in the dataset,  $p(i)$  refers to the external utility, which shows the weight of relative importance, and  $q(i, T_t)$  refers to the internal utility of the item for transaction  $T_t$ . Both utility values are positive integers.

Typically, the utility measure employed in HUIM uses internal and external utilities and refers to the amount of profit generated by each itemset in the market basket analysis. Equation (3) is used to calculate the utility of item  $i$  in transaction  $j$ , which is denoted as  $u(i, T_j)$ . The utility of an itemset  $X$  in a transaction  $T_j$  is found according to Equation (4), if  $X \subseteq T_j$ . Otherwise,  $u(X, T_j)$  is equal to 0. An itemset that has the same or a greater utility value than a defined *minimum utility threshold* is referred to as a high-utility itemset. The

*minimum utility threshold* corresponds to the *minimum support* (or *support count*) *threshold* in FIM. A detailed example is presented in Section 3.

$$u(i, T_j) = p(i) \times q(i, T_j) \quad (3)$$

$$u(X, T_j) = \sum_{i \in X} u(i, T_j) \quad (4)$$

The main difference between FIM and HUIM can be explained as follows. Itemsets with high frequencies cause the main issue in FIM while HUIM allows users to find itemsets by making use of a defined utility function [7,8]. The a priori property that is used by FIM states that if an itemset is frequent, all of its nonempty subsets must also be frequent [2]. However, this is not the case for HUIM. A superset of an itemset can be a high-utility itemset even if it can have a smaller utility value than the threshold. Therefore, HUIM is more complicated than FIM, because the utility measure it uses is not anti-monotone or monotone. Additionally, several candidate itemsets are generated to be evaluated against the utility function [8].

UP-Growth [10], HUI-Miner [12], FHM [13], FHM+ [14], EFIM [15], HMiner [22], ULB-Miner [23], and HUIM-SU [17] are some HUIM algorithms that aim to reduce the execution time. FHM+ is different from the others because it is the first algorithm that mines high-utility itemsets that have length constraints. Minimum length and maximum length constraints can be employed for a reduction in the search space. Some evolutionary and heuristic algorithms to find high-utility itemsets have also been proposed to shorten execution times [24–27]. In brief, the execution time for HUIM can be long and several studies have focused on this aspect to find the itemsets more efficiently.

Discovering a large number of high-utility itemsets is another important problem in HUIM. It cannot be possible to benefit from all discovered high-utility itemsets. Some alternative utility measures have been proposed [28–30], but to the best of our knowledge, there are no studies on HUIM that analyze the effects of an external utility discarding the internal utility. This study is a step toward the investigation of such an asymmetric approach.

### 3. Method

As previously discussed, the symmetric contributions of both internal and external utility values can detect an itemset with low profits as a “high-utility” itemset. We propose the method “high-utility itemset mining without internal utilities”, HUIM-WOIU, which considers only the profit of each item while ignoring the quantity values in each transaction.

Table 1 lists ten transactions of a sample quantitative database  $D_q$ , which contains the set of items  $I = \{s, t, v, w, x, y, z\}$ . The external utility values of the items are listed in Table 2. There are only two items in each transaction for simplicity. Looking at this table, it is straightforward to calculate the utility values of items in each context. For example, for item  $\{s\}$  in transaction  $T_3$  the utility value is  $u(s, T_3) = 2 \times 1 = 2$ . The utility of itemset  $\{s, v\}$  in transaction  $T_1$  is  $u(\{s, v\}, T_1) = u(s, T_1) + u(v, T_1) = 2 \times 1 + 1 \times 6 = 8$ . The utility of itemset  $\{s, v\}$  in  $D_q$  is  $u(\{s, v\}) = u(s) + u(v) = u(s, T_1) + u(s, T_3) + u(s, T_4) + u(s, T_7) + u(s, T_8) + u(v, T_1) + u(v, T_3) + u(v, T_4) + u(v, T_7) + u(v, T_8) = 4 + 2 + 4 + 4 + 4 + 6 + 3 + 5 + 1 + 3 = 36$ . The utilities of the itemsets are listed in Table 3, all calculated in the same fashion.

Suppose that the minimum utility threshold is set to 35, which is 25% of all profits. In this case, there are three high-utility itemsets,  $\{s, v\}$ ,  $\{x, y\}$ , and  $\{t, w\}$ . Although the external utilities of  $\{s\}$  and  $\{v\}$  are low,  $\{s, v\}$  is a high-utility itemset due to the high co-occurrence of the items in the database. For higher minimum utility thresholds, the itemsets that are composed of low internal utility values become low-utility itemsets. For example, when the minimum utility threshold is set to 39, which is 28% of all profits,  $\{s, v\}$  is no longer a high-utility itemset. This means that  $\{s, v\}$  was previously discovered as a high-utility itemset due to high internal values of  $s$  and  $v$ . Furthermore, a high number of co-occurring itemsets with high internal utility values causes a high number of high-utility itemsets. In this case, interesting or important high-utility itemsets can be overlooked. In addition, they can increase the total execution time.

**Table 1.** A quantitative database,  $D_q$ .

TID	Transaction
$T_1$	(s, 2), (v, 6)
$T_2$	(t, 1), (w, 1)
$T_3$	(s, 1), (v, 3)
$T_4$	(s, 2), (v, 5)
$T_5$	(t, 1), (w, 1)
$T_6$	(x, 1), (y, 1)
$T_7$	(s, 2), (v, 1)
$T_8$	(s, 2), (v, 3)
$T_9$	(x, 1), (y, 1)
$T_{10}$	(s, 2), (z, 2)

**Table 2.** External utility values for the items in  $D_q$ .

Item	External Utility
s	2
t	15
v	1
w	9
x	12
y	10
z	3

**Table 3.** Utilities of itemsets in  $D_q$  according to HUIM.

Itemsets	Utilities
(t, w)	48
(x, y)	44
(s, v)	36
(s, z)	10

Our method proposes the calculations of utility values of itemsets without considering their internal utilities. The utility of item  $i$  in transaction  $j$  is calculated as in Equation (5) where  $p(i)$  refers to the external utility. Equation (6) shows the calculation of the utility of an itemset  $X$  in a transaction  $T_j$ . This approach yields high-profit itemsets regardless of their quantities. For example, the utility of itemset  $\{s, v\}$  in transaction  $T_3$  is  $u'(\{s, v\}, T_1) = u'(s, T_1) + u'(v, T_1) = 2 + 1 = 3$ . The utility of  $\{s, v\}$  in the database is  $u'(s, v) = u'(s) + u'(v) = u'(s, T_1) + u'(s, T_3) + u'(s, T_4) + u'(s, T_7) + u'(s, T_8) + u'(v, T_1) + u'(v, T_3) + u'(v, T_4) + u'(v, T_7) + u'(v, T_8) = 2 + 2 + 2 + 2 + 2 + 1 + 1 + 1 + 1 + 1 = 15$ .

$$u'(i, T_j) = p(i) \quad (5)$$

$$u'(X, T_j) = \sum_{i \in X} u'(i, T_j) \quad (6)$$

The results of the proposed approach are listed in Table 4. When the minimum utility is set to 28, which is 25% of all profits without internal utilities, two high-utility itemsets are found:  $\{t, w\}$  and  $\{x, y\}$ . In contrast to existing methods,  $\{s, v\}$  is disregarded since the external utility values are not high enough. According to these results, it can be concluded that there are two itemsets, namely,  $\{t, w\}$  and  $\{x, y\}$ , which have high external utility values and are sold together.

**Table 4.** Utilities of itemsets in  $D_q$  according to HUIM\_WOIU.

Itemsets	Utilities
(t, w)	48
(x, y)	44
(s, x)	15
(s, z)	5

As the last example, consider Table 5, which is the same database in Table 1 without quantities and external utilities. Suppose that the minimum support is 25%, meaning that a frequent itemset must at least appear in three transactions. In this case, the support counts of itemsets are given in Table 6 and there is only one frequent itemset,  $\{s, v\}$ . As explained previously, this itemset is a high-utility itemset due to the high internal values of  $s$  and  $v$  in HUIM while it is disregarded in HUIM-WOIU. That is to say, frequent itemsets with low external utility values may be found as high-utility itemsets due to their high internal utility values.

**Table 5.** Database  $D$  without quantities and external utilities.

TID	Transaction
$T_1$	(s, v)
$T_2$	(t, w)
$T_3$	(s, v)
$T_4$	(s, v)
$T_5$	(t, w)
$T_6$	(x, y)
$T_7$	(s, v)
$T_8$	(s, v)
$T_9$	(x, y)
$T_{10}$	(s, z)

**Table 6.** Support count of itemsets in  $D$ .

Itemsets	Support Counts
(s, v)	5
(t, w)	2
(x, y)	2
(s, z)	1

From a different point of view, the mining approach can differ according to the data we have. Figure 1 shows the decision tree of the itemset mining approach according to the data limitations. Both the internal and external utility values of items are needed for HUIM, while HUIM-WOIU can be applied when the external utilities are known. Without external and internal utility values, only FIM can be done.

Without any data constraints, our proposal can be concluded as follows. FIM should be used when the aim is to find frequently co-occurring itemsets or when the dataset is not quantitative. HUIM should be used when all high-utility itemsets should be found. A high number of high-utility itemsets are generated in this case. HUIM-WOIU should be used to find the highly profitable itemsets with high external utilities. Compared to HUIM, HUIM-WOIU generates fewer candidate itemsets and high-utility itemsets. Moreover, the high-utility itemsets provide high utility without their sold quantities.

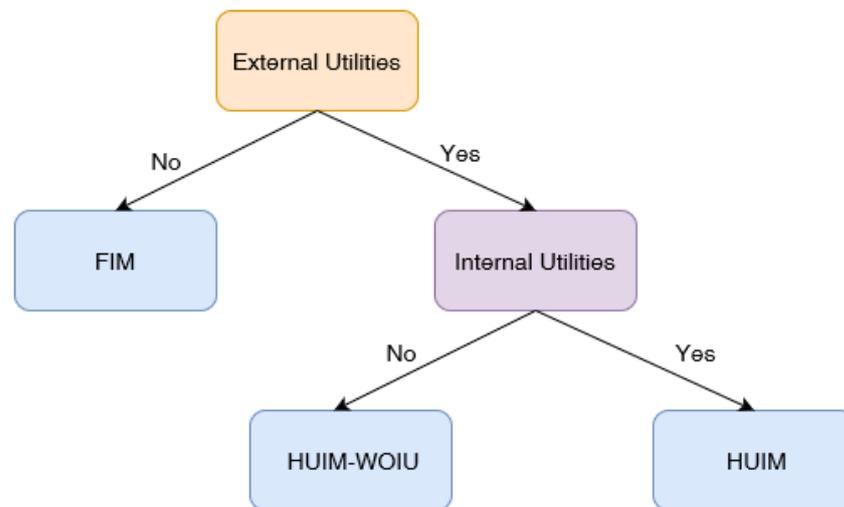


Figure 1. Decision tree according to the data.

#### 4. Experimental Analysis

It is vital to test the method using real datasets. Most of the datasets in the literature are published with their internal and external utility values multiplied, which makes it impossible to factorize them. Foodmart and chainstore datasets were chosen because they are the only ones that make their external and internal utility values available. The number of transactions in these datasets, the number of distinct items, and the average length of each transaction are given in Table 7. The original datasets along with the external and internal values of their items are available online (<https://www.philippe-fournier-viger.com/spmf/index.php?link=datasets.php>, accessed on 24 February 2021). We used the FHM+ algorithm [14], which exists in the SPMF data mining library [31], and generated the foodmart and chainstore datasets without internal utilities in the SPMF format. We performed the experiments on a computer with a 2.50 GHz processor and 8 GB of RAM. We found the high-utility itemsets according to HUIM-WOIU and HUIM for both datasets.

Table 7. Properties of datasets.

Dataset	Number of Transactions	Number of Distinct Items	Average Transaction Length
foodmart	4141	1559	4.4
chainstore	1,112,949	46,086	7.2

The results of HUIM-WOIU were compared to the results of HUIM as follows. Let the set of high-utility itemsets found by our proposed method be denoted by  $P_w$ . For each item  $i$  in high-utility itemsets in  $P_w$ , the high-utility itemsets found by HUIM were filtered if that itemset contained the item  $i$ . Among the filtered results, the high-utility itemset with the highest utility was expected to be in  $P_w$ , which we call a match. The matching score is the ratio of the number of matching high-utility itemsets to the number of high-utility itemsets in  $P_w$ . The score shows the effects of external utility values on the high-utility itemsets. We also compared the total execution times.

##### 4.1. Experiments

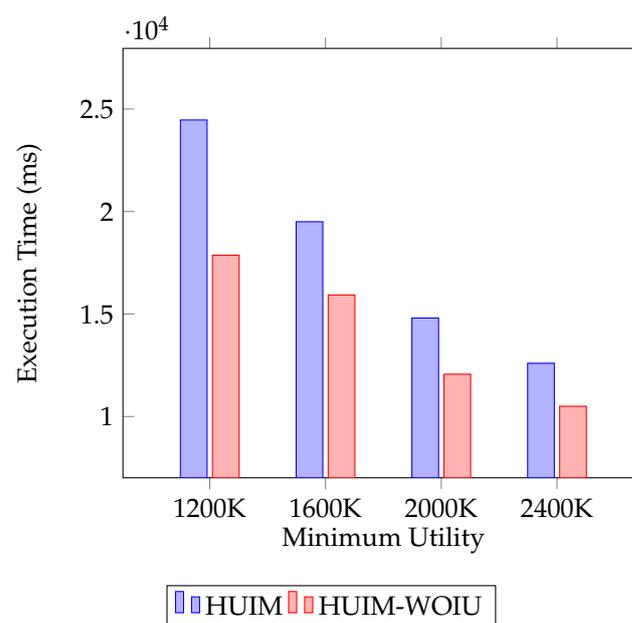
Table 8 shows the match success (in percentage points) for the chainstore datasets for different minimum utility thresholds, and different minimum and maximum lengths. As the table shows, the matching scores were between 92% and 100%. The average matching score was 98.4%. Furthermore, the highest utility itemsets for each minimum utility and minimum/maximum length in HUIM-WOIU are the same in HUIM. Figure 2 shows the comparison of our approach and HUIM in terms of execution time. In all minimum utility thresholds, HUIM-WOIU is faster

than HUIM, since it generates fewer candidates and high-utility itemsets. Figure 3 depicts the gained speed-up when HUIM-WOIU is used instead of HUIM. The speed-up decreases as the increase in minimum utility because the effect of the external utility on the HUIM increases. The results show that the same highest utility itemsets are found when the internal utility values are ignored. It means that external utility is more effective on the highest utility itemset than the internal utility for the chainstore dataset. The results related to the matching score also reveal that internal utility values affect the utility values of itemsets less than the external utility values.

Similar experiments were done on the foodmart dataset. As shown in Table 9, the matching scores were between 78% and 100% for different minimum utility thresholds, and minimum/maximum lengths. Although the lowest matching score was lower than the chainstore, the average matching score was 92%. As the minimum utility increased, the matching score typically increased due to the decrease in the effect of internal utility values. Since the aim of HUIM-WOIU is to find the high-utility itemsets with high external utility values, the results verify the appropriateness of the proposal. The execution times of the two approaches are given in Figure 4. As shown from the figure, the execution time of HUIM-WOIU was faster than HUIM in all minimum utility thresholds (since there were fewer high-utility and candidate itemsets in HUIM-WOIU). Figure 5 shows the speed improvement of HUIM-WOIU, and the speed-up decreased as the minimum utility threshold increased. It reveals that the effects of the external utility values on HUIM increased as the minimum utility threshold increased. The results are similar to the results of the chainstore dataset. They show that our approach finds high-utility itemsets with high external utilities and its execution time is faster than HUIM.

**Table 8.** Matching score for chainstore dataset.

Minimum Utility	Min Length = Max Length	Matching Score (%)	Same Highest Utility Itemset
1200K	2	97	✓
1200K	3	100	✓
1600K	2	100	✓
1600K	3	100	✓
2000K	2	92	✓
2000K	3	100	✓
2400K	2	100	✓



**Figure 2.** Execution times for the chainstore dataset.

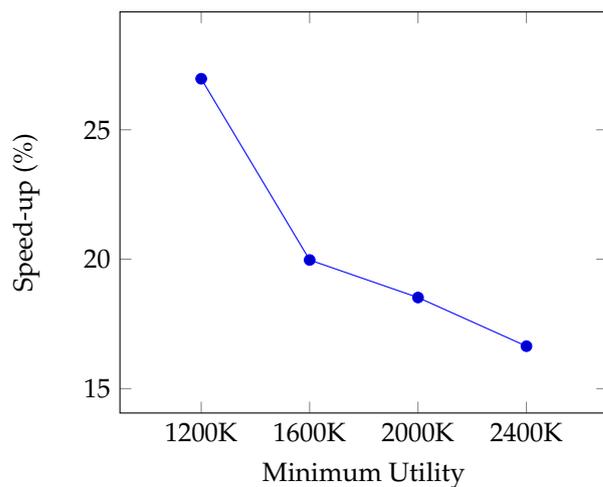


Figure 3. Speed-up for chainstore dataset.

Table 9. Matching score for the foodmart dataset.

Minimum Utility	Min Length = Max Length	Matching Score (%)
4000	2	87
4000	3	78
5000	2	78
5000	3	92
6000	2	100
6000	3	100
7000	2	100
7000	3	100

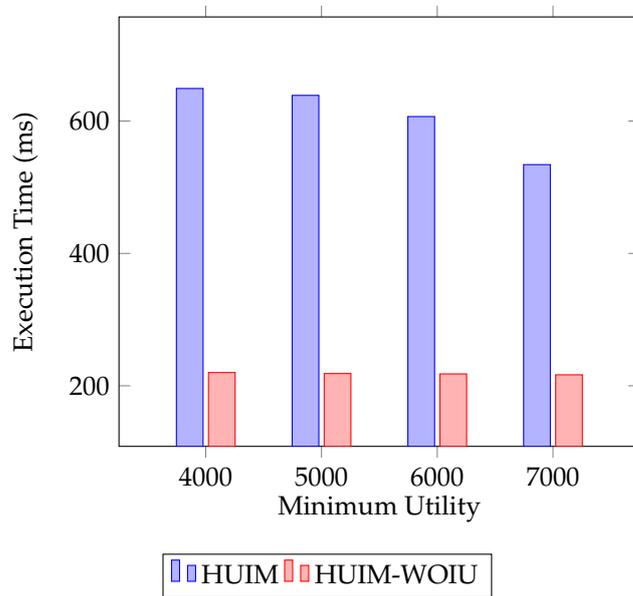
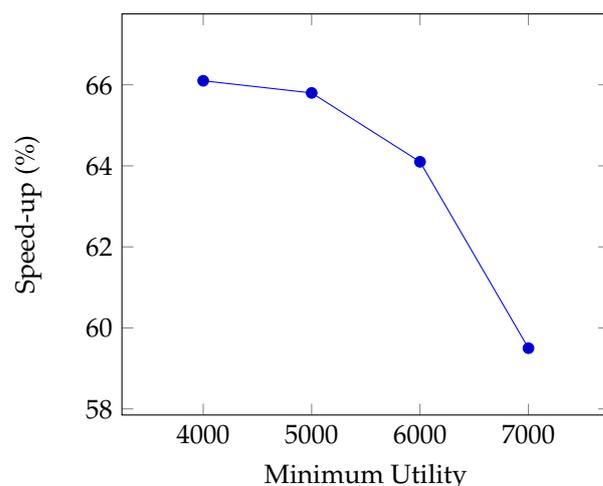


Figure 4. Execution times for the foodmart dataset.



**Figure 5.** Speed-up for foodmart dataset.

#### 4.2. Discussion

The experiments were done on two real datasets (chainstore and foodmart) with different minimum utility thresholds. We compared our proposal with the traditional high-utility itemset approach. To do that, we ran the FHM+ algorithm on the two real datasets for HUIM. Then, we generated two datasets without internal utility values using the chainstore and the foodmart datasets for our proposal and ran the FHM+ algorithm on them. As a comparison parameter, we calculated the matching score for each experiment, which evaluated the impact of external utility values on the high-utility itemsets. The matching score was 100% for five of seven cases for the test results on the chainstore. Besides the success in the matching score, the highest utility itemsets of the chainstore were the same for HUIM and HUIM-WOIU. It shows that external utility has a higher impact on the highest utility itemset than the internal utility. However, the same effect does not exist in the foodmart, which can be related to the low number of transactions and distinct items. Table 7 shows that the chainstore contains a large number of transactions and distinct items while the foodmart is a relatively small dataset with a limited number of distinct items. For this reason, the internal utility values can affect the highest-utility itemset. Nevertheless, the matching score was 100% for four of eight cases for foodmart, and the average matching score was 92%. We also compared the execution times of HUIM and HUIM-WOIU. In all minimum utility thresholds, HUIM was faster than HUIM-WOIU. In brief, ignoring internal utility values speeds up the execution time. The speed-up of our proposal compared to HUIM increased as the minimum utility threshold decreased in both datasets. This decrease reveals the impacts of external utilities becoming more important for high minimum utility thresholds. In other words, external utility values are more important than internal utility values for the itemset to be 'high utility' in a minimum utility threshold.

#### 5. Conclusions

Most studies in high-utility itemset mining have aimed to reduce execution times. However, there are two related and important challenges in high-utility itemset mining: (i) frequently sold together itemsets in large quantities are referred to as high-utility itemsets, and (ii) traditional high-utility itemset mining algorithms find many high-utility itemsets. In this paper, we focused on the meaning of high-utility itemsets in the market basket analysis. The main motivation was to discover an itemset as a high-utility itemset because of its external utility value, rather than its quantity sold. For this reason, our proposed approach ignores the internal utilities of items for high-utility itemset mining, thereby addressing the mentioned challenges. The experiments were done on two real datasets for various minimum utility thresholds. The results of the experiments show that the proposal is effective on the datasets tested and it speeds up the execution time.

In conclusion, the proposed HUIM-WOIU discovers itemsets that provide the defined profits separately from the quantity sold in each transaction. It finds fewer high-utility itemsets, which make more sense. Furthermore, it speeds up the execution time. This study showed that the external utility value has a great impact on deciding high-utility itemsets. As part of future work, we will aim to generate synthetic datasets with controlled properties to further analyze the impacts of external utility values in HUIM. We also plan to consider the opposite point of view and analyze the impacts of internal utility values on HUIM.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Dataset used in this study are available or can be generated from: <https://www.philippe-fournier-viger.com/spmf/index.php?link=datasets.php>, accessed on 24 February 2021.

**Conflicts of Interest:** The author declares no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

FIM	Frequent Itemset Mining
HUIM	High-Utility Itemset Mining
HUIM-WOIU	High-Utility Itemset Mining without Internal Utility

## References

1. Agrawal, R.; Imielinski, T.; Swami, A.N. Mining Association Rules between Sets of Items in Large Databases. In Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, DC, USA, 26–28 May 1993; Buneman, P., Jajodia, S., Eds.; ACM Press: New York, NY, USA, 1993; pp. 207–216. [CrossRef]
2. Han, J.; Kamber, M.; Pei, J. *Data Mining: Concepts and Techniques*, 3rd ed.; Morgan Kaufmann: Burlington, MA, USA, 2011.
3. Fournier-Viger, P.; Lin, J.C.W.; Vo, B.; Chi, T.T.; Zhang, J.; Le, H.B. A survey of itemset mining. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2017**, *7*, e1207. [CrossRef]
4. Luna, J.M.; Fournier-Viger, P.; Ventura, S. Frequent itemset mining: A 25 years review. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2019**, *9*, e1329. [CrossRef]
5. Chee, C.H.; Jaafar, J.; Aziz, I.A.; Hasan, M.H.; Yeoh, W. Algorithms for frequent itemset mining: A literature review. *Artif. Intell. Rev.* **2019**, *52*, 2603–2621. [CrossRef]
6. Yao, H.; Hamilton, H.J.; Butz, C.J. A foundational approach to mining itemset utilities from databases. In Proceedings of the 2004 SIAM International Conference on Data Mining, Lake Buena Vista, FL, USA, 22–24 April 2004; SIAM: Philadelphia, PA, USA, 2004; pp. 482–486.
7. Yao, H.; Hamilton, H.J. Mining itemset utilities from transaction databases. *Data Knowl. Eng.* **2006**, *59*, 603–626. [CrossRef]
8. Fournier-Viger, P.; Lin, J.C.-W.; Chi, T.T.; Nkambou, R. A Survey of High Utility Itemset Mining. In *High-Utility Pattern Mining*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 1–45.
9. Liu, Y.; Liao, W.K.; Choudhary, A. A two-phase algorithm for fast discovery of high utility itemsets. In Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, Hanoi, Vietnam, 18–20 May 2005; Springer: Berlin/Heidelberg, Germany, 2005; pp. 689–695.
10. Tseng, V.S.; Wu, C.W.; Shie, B.E.; Yu, P.S. UP-Growth: An efficient algorithm for high utility itemset mining. In Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 24–28 July 2010; pp. 253–262.
11. Lin, C.W.; Hong, T.P.; Lu, W.H. An effective tree structure for mining high utility itemsets. *Expert Syst. Appl.* **2011**, *38*, 7419–7424. [CrossRef]
12. Liu, M.; Qu, J. Mining high utility itemsets without candidate generation. In Proceedings of the 21st ACM International Conference on Information and Knowledge Management, Maui, HI, USA, 29 October–2 November 2012; pp. 55–64.
13. Fournier-Viger, P.; Wu, C.W.; Zida, S.; Tseng, V.S. FHM: Faster high-utility itemset mining using estimated utility co-occurrence pruning. In Proceedings of the International Symposium on Methodologies for Intelligent Systems, Roskilde, Denmark, 25–27 June 2014; Springer: Berlin/Heidelberg, Germany, 2014; pp. 83–92.
14. Fournier-Viger, P.; Lin, J.C.W.; Duong, Q.H.; Dam, T.L. FHM+: Faster High-Utility Itemset Mining Using Length Upper-Bound Reduction. In Proceedings of the International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, Morioka, Japan, 2–4 August 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 115–127.

15. Zida, S.; Fournier-Viger, P.; Lin, J.C.W.; Wu, C.W.; Tseng, V.S. EFIM: A fast and memory efficient algorithm for high-utility itemset mining. *Knowl. Inf. Syst.* **2017**, *51*, 595–625. [[CrossRef](#)]
16. Wu, J.M.T.; Lin, J.C.W.; Tamrakar, A. High-utility itemset mining with effective pruning strategies. *ACM Trans. Knowl. Discov. Data (TKDD)* **2019**, *13*, 1–22. [[CrossRef](#)]
17. Cheng, Z.; Fang, W.; Shen, W.; Lin, J.C.W.; Yuan, B. An efficient utility-list based high-utility itemset mining algorithm. *Appl. Intell.* **2022**, 1–15. [[CrossRef](#)]
18. Lin, J.C.W.; Djenouri, Y.; Srivastava, G.; Fourier-Viger, P. Efficient evolutionary computation model of closed high-utility itemset mining. *Appl. Intell.* **2022**, *52*, 10604–10616. [[CrossRef](#)]
19. Agrawal, R.; Srikant, R. Fast algorithms for mining association rules. In Proceedings of the 20th International Conference on Very Large Data Bases, VLDB, Santiago de Chile, Chile, 12–15 September 1994; Citeseer: Princeton, NJ, USA, 1994; Volume 1215, pp. 487–499.
20. Zaki, M.J.; Parthasarathy, S.; Ogihara, M.; Li, W. New algorithms for fast discovery of association rules. In Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD-97), Newport Beach, CA, USA, 14–17 August 1997; Volume 97, pp. 283–286.
21. Han, J.; Pei, J.; Yin, Y. Mining frequent patterns without candidate generation. *ACM SIGMOD Rec.* **2000**, *29*, 1–12. [[CrossRef](#)]
22. Krishnamoorthy, S. HMiner: Efficiently mining high utility itemsets. *Expert Syst. Appl.* **2017**, *90*, 168–183. [[CrossRef](#)]
23. Duong, Q.H.; Fournier-Viger, P.; Ramampiaro, H.; Nørvåg, K.; Dam, T.L. Efficient high utility itemset mining using buffered utility-lists. *Appl. Intell.* **2018**, *48*, 1859–1877. [[CrossRef](#)]
24. Song, W.; Huang, C. Mining high utility itemsets using bio-inspired algorithms: A diverse optimal value framework. *IEEE Access* **2018**, *6*, 19568–19582. [[CrossRef](#)]
25. Nawaz, M.S.; Fournier-Viger, P.; Yun, U.; Wu, Y.; Song, W. Mining high utility itemsets with Hill climbing and simulated annealing. *ACM Trans. Manag. Inf. Syst. (TMIS)* **2021**, *13*, 1–22. [[CrossRef](#)]
26. Song, W.; Li, J.; Huang, C. Artificial fish swarm algorithm for mining high utility itemsets. In Proceedings of the International Conference on Swarm Intelligence, Qingdao, China, 17–21 July 2021; Springer: Berlin/Heidelberg, Germany, 2021; pp. 407–419.
27. Fang, W.; Zhang, Q.; Lu, H.; Lin, J.C.W. High-utility itemsets mining based on binary particle swarm optimization with multiple adjustment strategies. *Appl. Soft Comput.* **2022**, *124*, 109073. [[CrossRef](#)]
28. Yao, H.; Hamilton, H.J.; Geng, L. A unified framework for utility-based measures for mining itemsets. In Proceedings of the ACM SIGKDD 2nd Workshop on Utility-Based Data Mining, Philadelphia, PA, USA, 20 August 2006; Citeseer: Princeton, NJ, USA, 2006; pp. 28–37.
29. Hong, T.P.; Lee, C.H.; Wang, S.L. Effective utility mining with the measure of average utility. *Expert Syst. Appl.* **2011**, *38*, 8259–8265. [[CrossRef](#)]
30. Gan, W.; Lin, J.C.W.; Fournier-Viger, P.; Chao, H.C.; Tseng, V.S.; Philip, S.Y. A survey of utility-oriented pattern mining. *IEEE Trans. Knowl. Data Eng.* **2019**, *33*, 1306–1327. [[CrossRef](#)]
31. Fournier-Viger, P.; Gomariz, A.; Gueniche, T.; Soltani, A.; Wu, C.W.; Tseng, V.S. Spmf: A java open-source pattern mining library. *J. Mach. Learn. Res.* **2014**, *15*, 3389–3393.