An efficient HUIBA Algorithm for Mining High Utility Itemset

M.S.Bhuvaneswari[1], N.Balaganesh[2]

[1,2] Assistant Professor(Sl.Grade), Mepco Schlenk Engineering College,Sivakasi, India

[1]bhuvaneswari@mepcoeng.ac.in, [2]balaganesh@mepcoeng.ac.in

Abstract: *Utility Mining is to spot the itemsets with highest utilities, by considering profit, quantity, cost or other user preferences. Mining High Utility itemsets from a transaction database is to seek out itemsets that have utility above a user-specified threshold. Bio inspired algorithm is extremely efficient for mining High Utility Itemset(HUI), but it will not find all HUI in the database and the quality is poor within the number of discovered HUI. A replacement framework using BA algorithm is proposed to rectify this issue. The proposed algorithm is more efficient in terms of quality and convergence speed when put next to other algorithms.*

Keywords: **Itemset Mining, Utility Itemset Mining, High Utility Itemset**

1. Introduction

An emerging topic, within the field of knowledge mining is Utility Mining. The most objective of Utility Mining is to spot the itemsets with highest utilities, by considering profit, quantity, cost or other user preferences. Mining High Utility itemsets from a transaction database is to seek out itemsets that have utility above a user-specified threshold. In many real-life applications, high-utility itemsets incorporate rare items. Rare itemsets provide useful information in several decision-making domains like business transactions, medical, security, fraudulent transactions, retail communities. As an example, in an exceedingly supermarket, customers purchase microwave ovens or frying pans rarely as compared to bread, soap powder, soap. But the previous transactions yield more profit for the supermarket.

In utility mining, profit is associated with each item in the transaction database. Each item can occur multiple times in the transaction. Utility of an item can be calculated, by adding the product of the item's profit along with its frequency of occurrence in the transaction. High utility itemset are itemsets which is no longer less than minimum support threshold specified by user. High utility itemset mining is the process of mining all high utility itemsets in the given transaction database.

There are many algorithms used for mining high utility itemsets. Two phase algorithms[1] are used for mining HUIs based on transaction weighted downward closure property. In this approach the database is scanned multiple times which increases the time as well as memory. To avoid multiple database scans and candidate generation, the pattern growth and tree-based algorithm is used for mining HUIs efficiently. Examples are IHUP-tree and UP-tree.

Bio-inspired Algorithms repeatedly improves the candidate solution based on the given measure of quality. Under bio-inspired algorithm there are three algorithms for mining high utility itemset: i)Genetic algorithm ii) Particle Swarm algorithm and iii) Bat algorithm. In this technique echolocation behavior of bats is used. It is used for differing the pulse rate and loudness of emissions of bats. In this each individual represents the bat. Each bat is randomly generated. It will update its velocity, position and frequency to obtain a best solution. When a bat is nearing its prey the loudness of bat is reduced and its pulse rate is increased so that it's not visible to its prey. All bats update their frequency position and pulse rate until the termination condition is reached or the best solution is found.

Bat inspired algorithms is basically used for solving non-linear global optimization problems. It is extended to solve multi objective optimization problems[2]. The advantage of using Bat algorithm is to solve the Engineering Optimization problems which are multi objective and multidisciplinary with complex constraints. HACE theorem proposed by [3] aims to explore data relationships which are dynamic and changing by having the characteristics of heterogeneous, autonomous sources with distributed and decentralized control. To find interesting, pairwise generalized rules of association connecting multiple ontology concepts, a new class of hierarchical interestingness measures was proposed by [4] TKU (mining Top-K Utility itemsets) and TKO (mining Top-K utility itemsets in One phase) [5] deals with mining high utility itemsets without the need to set min_util.

In HUITW [6], a replacement approach was proposed which uses pruning and bagging methods to boost performance for high utility pattern mining. Tram Tran[7] proposed a Maximum Capturing (MC) algorithm for text clustering based on frequent weighted utility itemsets (FWUI). When dealing with large database, frequent itemset mining is characterized by poor runtime performance. So, to overcome this issue Youcef Djenouri and Marco Comuzzi[8] proposed an approach to improve the quality of the solution by randomized search of solution space including the intrinsic feature of FIM. Zhicong Kou et al[9] proposed a binary particle swarm optimization (BPSO-ARM) to reveal the hidden relationships between system capabilities and product characteristics.

The rest of the paper is organized as follows: Section 2 discusses about the overall design of the proposed approach. Section 3 discusses about the results and Section 4 gives an overall view about the proposed approach.

2. Proposed High Utility Itemset Mining Approach

The architectural design for the High Utility Pattern Mining using HUIBA Algorithm is shown in Figure 1.

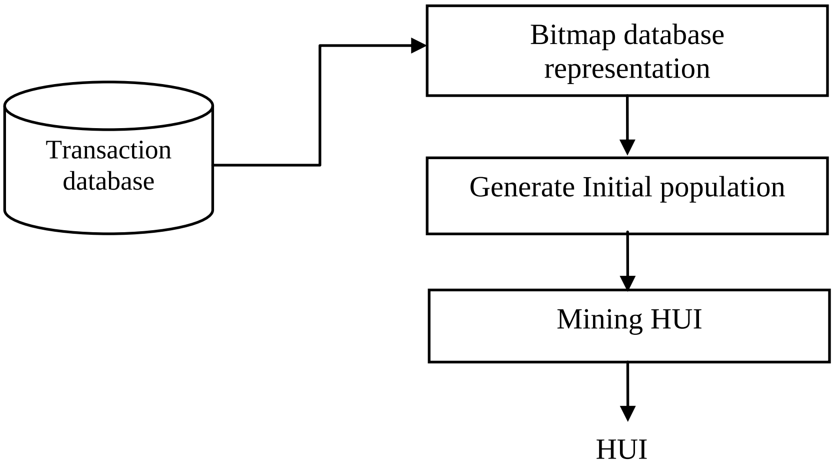

Figure 1. Architectural design for the High Utility Pattern Mining using HUIF-BA Algorithm

The input is given in the form of transaction database along with utility information, minimum utility value and maximum number of iterations. The transaction database is then converted into bitmap which consists of 0s or 1s. The output obtained from the first phase is bitmap and it is then sent for producing set of individuals using promising encoding vector checking process and roulette wheel selection. In this process the chromosomes are produced and are discarded if they are unpromising. If it is promising then the process continues. Then the output obtained from this phase is given as input to last phase where HUIs are obtained. The various modules involved in the work are as follows:Bitmap database representation, Generate initial population, Mining high utility itemsets.

2.1 Bitmap database representation

The original database is converted into a bitmap which is used for mining high utility itemsets. Let D be transaction database and bitmap of D is N*M Boolean matrix B(D), where N represents the number of rows and M represents the number of columns. These entries will be set of 0s or 1s. The entry will be set to 1 if and only if that particular item is in that transaction else the entry will be set as 0. In this bitmap matrix the bitmap cover of an item is written as Bit(item) and for an itemset X is written as Bit(X)=bitwise-AND (bit(item)). It is a bit vector which had been got from the bitwise AND operations on the bitmap covers of all the items included in that itemset X. If there are more than one itemsets such as X and Y, then it is written as Bit (XUY) =bitwise-AND of Bit(X) &Bit(Y). Let's consider an example of transaction database and how the bitmap is produced.

Table 1. Profit table

Item	**a**	**b**	**c**	**d**	**e**	**f**
Profit	3	9	1	5	6	1

Table 2. Transaction database

Tid	**Transactions**	**TU**
T1	(a,1), (c,18), (e,1)	27
T2	(b,6) (d,1), (e,1), (f,1)	66
T3	(a,2), (c,1), (e,1)	13
T4	(d,1), (e,1)	11
T5	(c,4), (e,2)	16

In Table 1 the profit for individual items is present in the transaction. In Table 2 the transactions along with their transaction utility is present. In this transaction table there are five transactions with Transaction id uniquely differentiating them and transaction id is represented as Tid. Here a, c and e are the items present in the transaction T1. In transaction T1, (a,1) specifies that item a has occurred one time, similarly (c,18) specifies that c has occurred eighteen times and (e,1) specifies that e has occurred one time. The itemset value will be multiplied with the profit value of the item.

The utility of transaction a in T1 is obtained by u (a, T1) =1*3=3. Likewise, each itemset value in the transaction will be multiplied with the profit value and the results will be summed up and that is known as Transaction utility.

The transaction utility in transaction T1 is obtained by TU (T1) = u (ace, T1) where u (ace, T1) = u (a, T1) + u (c, T1) + u (e, T1). Similarly, the TU of all transactions are obtained. The transaction weighted utility of the item a is obtained by TWU (a) = TU(T1) + TU(T3) =27 + 13 = 40. Similarly, the TWU for all the items are calculated.

If the transaction weighted utility of an itemset is less than the minimum utility it is removed from the transaction database. The transaction database is reorganized in Table 3.

Table 3. Reorganized transaction database

Tid	Transactions	TU
T1	(c,18), (e,1)	24
T2	(b,6) (d,1), (e,1), (f,1)	66
T3	(c,1), (e,1)	7
T4	(d,1), (e,1)	11
T5	(c,4), (e,2)	16

The bitmap of these transactions is depicted in Table 4.5. If an item is present in that transaction it is marked as 1 otherwise it is marked as 0. In transaction T1 the items c and e are only present so they are marked as 1, other items are not present so they are marked as 0. In a similar way bitmap for remaining transactions are also obtained and is tabulated in Table 4.

Table 4. Bitmap representation

Tid	b	c	d	e	f
T1	0	1	0	1	0
T2	1	0	1	1	1
T3	0	1	0	1	0
T4	0	0	1	1	0
T5	0	1	0	1	0

2.2 Generate initial population

The initialization will be done initially in a random manner with PS individuals. The database will be the input along with the population size (PS) .PS will be calculated using roulette wheel selection method. The scanning will be done to determine the length-1 HTWUIs. Then, the bitmap representation of the pruned database will be constructed. The random number of 1’s will be assigned for each individual in the ith bit vector where numi is an integer between 1 and the number of length-1HTWUIs.

If a length-1 HTWUIs TWU value is small, it has a higher probability of being selected in an individual. The promising encoding vector checking (PEVC) pruning strategy is implemented only if the numi> 1. Every bit in a bit vector corresponds to a length-1 HTWUI, so one or more transactions obviously contain each length-1 HTWUI. The PEVC strategy is described below.

The encoding vector is used for individual representation. The encoding vector consists of 0’s and 1’s which corresponds to whether an item is absent or present in an individual. The individuals can be chromosome, particle or bat. If an individual's corresponding j^{th} position contains 1, an item in the j^{th} position is present in potential HUI, otherwise this item will not be included in a potential HUI, and it is discarded. The size of the encoding vector each individual represents is equal to the number of 1-HTWUIs in the database.

Let V_{ev} be an encoding vector made up of 0s or 1s, and let Y be the element collection that $_{Vev}$ represents. If Bit(Y) consists of zeros only, V_{ev} is considered as an uncompromising encoding vector (UPEV), otherwise, it is considered as promising encoding vector (PEV). If a newly generated encoding vector is a UPEV, computations of the fitness value will be ignored. This technique is called the PEV check (PEVC) pruning strategy.

Firstly, calculate the number of 1s in the encoding vector, and identify which items these 1s represent, next the bitwise-AND operation is done to all bitmap covers of items in V which is initialized by the bitmap cover of the first item. Followed by that we need to perform the bitwise-AND operation with the bitmap cover of the next item. If the resulting bit vector is an unpromising encoding vector, this item cannot be included in the final bit vector. Lastly there is backtracking of the result of the bitwise-AND operation.

Assume V is a vector and it takes the value as V=000111 and X is the itemset represented by V. If Bit(X)=000000 then V is called as unpromising encoding vector. If Bit(X) consists of 0s and 1s then it is called as promising encoding vector checking

2.3 Mining High Utility Itemsets

The individuals generated are given as input to this phase. The promising vector will be calculated based on the previous module and the itemsets with utilities higher than minimum utility will be kept and the next population can be obtained until the maximum iteration limit is reached. Then the HUIs will be obtained.

BA algorithm uses echolocation behaviour of bats. It is used for differing the pulse rate and loudness of emissions of bats. In this each individual represents the bat. It will update its velocity, position and frequency to obtain a best solution. All bats update their frequency position and pulse rate until the termination condition is reached or the best solution is found. In each population, these 3 values will be updated using the Equations 1-3,

$$f_i = f_{min} + (f_{max} - f_{min})\alpha \qquad (1)$$

$$V_i^{t+1} = V_i^t + (X_i^t - gbest)f_i \qquad (2)$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \qquad (3)$$

where f_i is the frequency of the ith bat (used to adjust the velocity), f_{min}, f_{max} are the minimum/maximum frequencies of the pulses emitted by all bats, $\alpha \in [0,1]$ is a random number, V_i^t, V_i^{t+1} are the velocities of the ith bat at iterations t and t+1, X_i^{t+1}, X_i^t are the locations of the ith bat at iterations t and t +1, and gbest is the current global best location. Bat B_i should lower its loudness when approaching the prey and increase the rate of pulse emission. It can be simulated by the Equations (4) and (5).

$$L_i^{t+1} = \beta L_i^t \qquad (4)$$

$$R_i^{t+1} = R_i^0(1 - exp(-\gamma t)) \qquad (5)$$

where L_i^t, L_i^{t+1} denote the loudness at iterations t and t +1, R_i^0 i is the initial rate of pulse emission, , R_i^{t+1} i is the rate of pulse emission at iteration t +1, and $0 < \beta < 1$, $\gamma > 0$ are constants. Every bat must update repeatedly the velocities, positions, loudness and pulse emission rate before the best solution is found or the maximum number of iterations is reached

3. Results and Discussion

The performance of the system is measured in terms of memory, runtime and the number of high utility itemsets generated.

3.1 Analysis based on runtime

The time taken for identifying the High Utility itemsets by varying the minimum utility value is shown in Figure 2. From the figure it is evident that the minimum utility value and the run time is directly proportional to each other. If the value of minimum utility is high, then the run time value is also high.

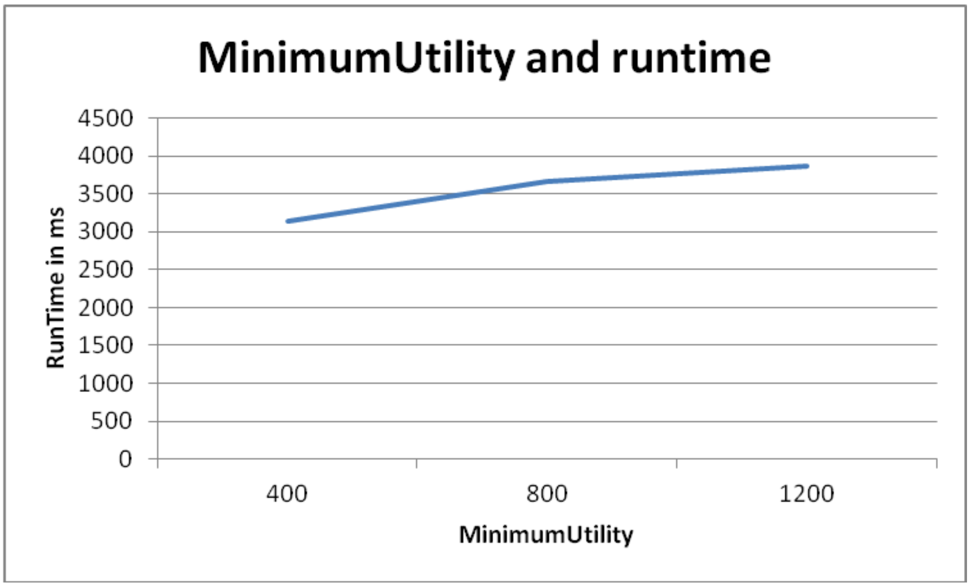

Figure 2. Runtime vs MinimumUtility

3.2 Analysis based on Memory usage:

The memory usage is represented in Megabytes (MB). The memory usage for identifying the High Utility itemsets by varying the minimum utility value is shown in Figure 3. From the figure it is evident that the minimum utility value and the memory usage is directly proportional to each other. If the value of minimum utility is high, then the memory usage is also high.

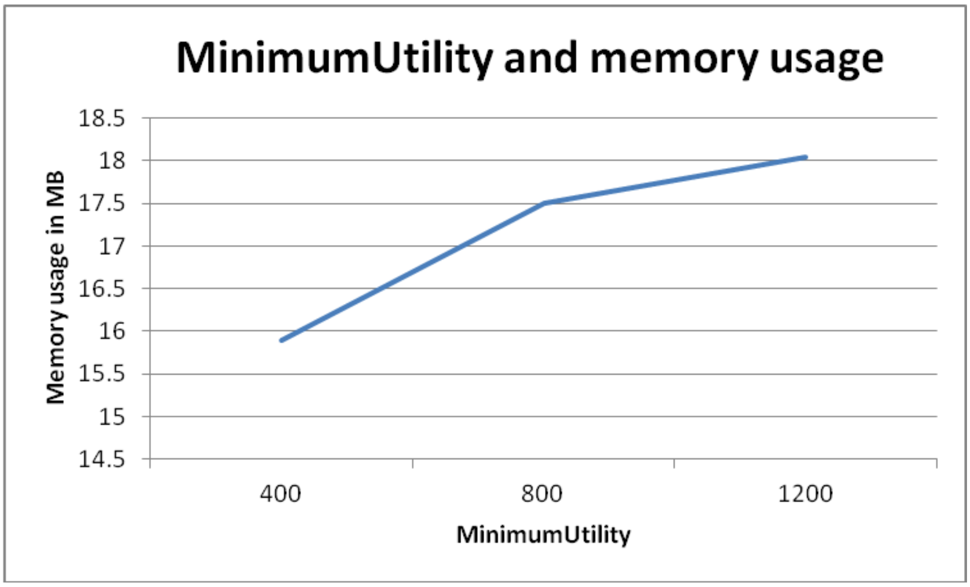

Figure 3. Memory usage vs MinimumUtility

3.3 Analysis based on HUI count:

The HUI count for identifying the High Utility itemsets by varying the minimum utility value is shown in Figure 4. From the figure it is evident that the Minimum Utility value and HUI count are inversely proportional to each other.

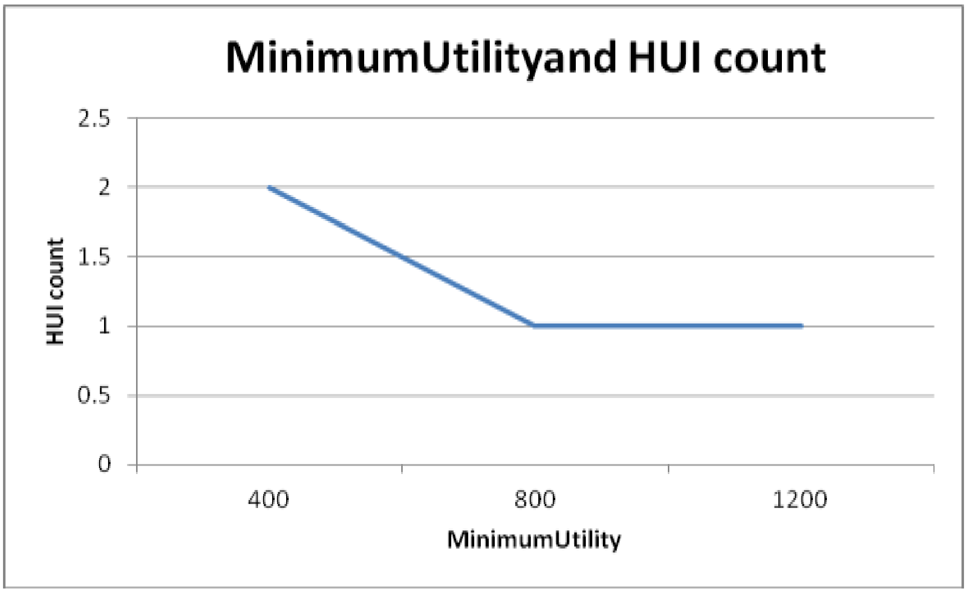

Figure 4. HUI count vs MinimumUtility.

4. Conclusion

The proposed work aims at bringing all sets of products that customers buy together that produce a high profit. Bio-inspired algorithm is extremely efficient for HUI mining, but not all HUI will be found in the database and the efficiency is low in the number of HUIs discovered. To rectify this problem, a replacement framework is proposed using BA algorithm. This algorithm is more efficient in terms of quality and convergence speed when put next to other algorithms. The major impact of the proposed HUIBA are: it prevents the expensive generation of candidates, Utility computation can be reduced, it is easy to classify itemsets with high utility income

Conflict of interest: The authors declare that there is no conflict of interest

REFERENCES

[1] Liu Y., Liao W., Choudhary A. (2005) A Two-Phase Algorithm for Fast Discovery of High Utility Itemsets. In: Ho T.B., Cheung D., Liu H. (eds) Advances in Knowledge Discovery and Data Mining. PAKDD 2005. Lecture Notes in Computer Science, vol. 3518. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11430919_79.

[2] Yang, Xin-She & Gandomi, Amir. (2012). Bat Algorithm: A Novel Approach for Global Engineering Optimization. Vol.29. https://doi.org/10.1108/02644401211235834.

[3] Wu, Xindong & Zhu, Xingquan & Wu, Gongqing & Ding, Wei. (2014). Data Mining with Big Data. Knowledge and Data Engineering, IEEE Transactions on. https://doi.org/10.1109/TKDE.2013.109.

[4] Benites, Fernando & Sapozhnikova, Elena. (2014). Using Semantic Data Mining for Classification Improvement and Knowledge Extraction. https://doi.org/10.13140/2.1.3706.0803.

[5] Wu, Cheng-Wei & Fournier Viger, Philippe & Yu, Philip & Tseng, Vincent. (2011). Efficient Mining of a Concise and Lossless Representation of High Utility Itemsets. Proceedings - IEEE International Conference on Data Mining, ICDM. pp. 824-833. https://doi.org/10.1109/ICDM.2011.60.

[6] Guo, Shi-Ming. (2016). HUITWU: An Efficient Algorithm for High-Utility Itemset Mining in Transaction Databases. Journal of Computer Science and Technology. vol. 31, pp. 776-786. https://doi.org/10.1007/s11390-016-1662-2.

[7] Tran, Tram & Vo, Bay & Le, Tho & Nguyen, Ngoc Thanh. (2017). Text Clustering Using Frequent Weighted Utility Itemsets. Cybernetics and Systems. vol. 48. pp. 193-209. https://doi.org/10.1080/01969722.2016.1276774.

[8] Djenouri, Youcef & Habbas, Zineb & Djenouri, Djamel & Comuzzi, Marco. (2017). Diversification Heuristics in Bees Swarm Optimization for Association Rules Mining. pp. 68-78. https://doi.org/10.1007/978-3-319-67274-8_7.

[9] Zhicong Kou and Lifeng Xi(2018) "Binary Particle Swarm Optimization-Based Association Rule Mining for Discovering Relationships between machine capabilities and product features", Mathematical Problems in Engineering (Hindawi),https://doi.org/10.1155/2018/2456010.