Received: July 9, 2017

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Jaccard Similarity based Mining for High Utility Webpage Sets from Weblog Database

Vinod Kumar[1*] **Ramjeevan Singh Thakur[1]**

[1,2]*Department of Computer Applications, Maulana Azad National Institute of Technology, Bhopal, India*
* Corresponding author's Email: vinodkumarfbkp@gmail.com

Abstract: High utility webpage set states to those set of webpages which have high utility value in a weblog database. To find high utility item sets from transactional databases, there exist various algorithms. However, these existing algorithms mostly deals with data like- categorical, time series, binary etc. The weblog data is different from other types of data. The utility webpages extracted from the log data can be used for knowing the user's behaviour. In this research paper, two algorithms named HUWSM (high utility webpage sets mining) and HUWP-FP (high utility webpage sets - frequent pattern) Tree have been developed and used for efficiently mining high utility webpage sets from web log database. Along with this, a pattern generation technique based on the 'Jaccard Similarity' is also included in this method. The HUWSM method has also been compared with various other existing methods. This algorithm has shown a much better performance and more effectiveness over other algorithms like FHM, HUI-MINER and IHUP methods in terms of memory consumption and execution time.

Keywords: Web usage mining, Frequent pattern, High utility webpage sets, Jaccard similarity, Weblog.

1. Introduction

The web mining [1] procedure is like data mining process. The difference is generally in the data collection. In traditional data mining, the data is often already collected and stored in a data warehouse. For Web mining, data collection can be a substantial task, especially for Web structure and content mining, which involves going through a large number of target Web pages. Web Content Mining: Web content mining extracts or mines useful information or knowledge from web page contents. For example, we can automatically classify and cluster web pages according to their topics. Web structure mining: Web structure mining discovers useful information from hyperlinks, which represent the structure of the web. For example, the communities of users, who share common interest, can be discovered. Web usage mining [2] refers to the automatic discovery and analysis of patterns in click stream and associated data collected or generated as a result of user interactions with web resources on one or more web sites [3,4]. Conceptually, its main objective is to capture, model and analyse the behavioural patterns [5] and profiles of user's interactions with a website. The overall web usage mining process can be divided into three interdependent stages: data collection, pre-processing, pattern discovery, and pattern analysis. Various popular techniques are available in data mining for extracting information from weblog data. According to [6], association rule mining finds sets of data items that occur together frequently. Sequential patterns mining find sets of data items that occur together frequently in some sequences. Clearly, they can be used to find the patterns in the web log data [7] using support and confidence value. For example, in web usage mining, association rule mining can be used to find user's visit and purchase patterns, and sequential pattern can be used to find user's navigation patterns. Since, traditional frequent item sets mining algorithms cannot evaluate the utility information about item sets. Thus, the item sets extracted may be frequent but not profitable or high utility [8] item set. This problem

can be solved by taking into account the utility value of each item for appropriate result. The utility of webpage item refers to profitability value and interestingness [9] of the webpage [10]. In a web transactional database, the utility of a webpage rests on two types of utility: internal utility (IU) and external utility (EU). Internal utility (IU) [11] indicates importance to some webpage in every transaction (i.e. number of times the webpage is referenced in a session) and external utility consists of impor¬tance of distinct item [12, 13]. A webpage is of high utility [14, 15] if its utility is greater than the minimum specified utility value.

The proposed method HUWSM focuses on magnitude of utility value rather than quantity of webpage sets. High utility frequent pattern (HUW-FP) tree [16] data structure helps to store frequent high utility webpages in a compact form. Thus, patterns of high utility webpages can be obtained with only two database scans and it avoids the frequent web database scan. This applies MTU value and jaccard similarity (JS) approaches and takes into account only the high utility webpage sets and discard the low utility webpage sets lower than the given threshold value. Consequently, it gives better result as compared to existing methods FHM, IHUP, HUI-Miner. The proposed method performs well in terms of time and space complexity. Overall, the proposed method is better than the previous state of the art methods for mining high utility webpage set. The rest of this paper is organised as follows - Section 2 presents related works. Section 3 presents preliminary terms and related definitions. Section 4 demonstrates about utility quantization for items in web transactions database. Section 5 shows methodology for high utility webpage sets mining. Section 6 presents the experimental evaluation of method HUWSM. Section 7 presents a conclusion and future works.

2. Related works

This section covers related research work done over the high utility item set extraction. There exists huge number of research literatures on this particular topic. [6] used the a priori to mine the association rule between sets of items in large databases but it requires the large number of database and computation time. [17] Proposed a two-phase algorithm for fast discovery of high utility item sets. But it performs multiple scans of database and generates many candidate Item sets. In paper [18] developed ‘U mining’, an algorithm for mining item set utilities from Transactional databases. This algorithm applies the pruning strategy to reduce the search space for finding the high utility item sets. Due to the pruning process, it misses some utility item sets. ‘U mining’ still suffers from this above shortcoming in the algorithm. Moreover, according to [14], a novel algorithm for mining high utility item sets has been proposed. This algorithm is still time and memory consuming. Authors in [17], have given HUI_Miner (high utility item sets miner) and the proposed algorithm for mining high utility item sets without candidate generation to reduce the costly execution time. This performs costly join operations on each pattern search. In [18], proposed an IHUP, the efficient tree structure for high utility pattern mining [19, 20] in case of incremental database. This algorithm generates huge sets of PHUI when threshold is kept low for the long transactions. In [21] proposed two algorithms UP_Growth and UP_Growth++, both are efficient algorithms for mining high utility Item sets from transactional database, but both the algorithms are complex for solution due to tree data structure. [21] Proposed a method FHM-fast high utility mining for large memory overhead. This is still suffering from large memory overhead. The FHM [22] algorithm occupies a large amount of memory space and it is time consuming too, because it computes all the candidate sets which makes it difficult to handle larger databases. In [23], IHUP recommends three novel data structures to efficiently perform incremental and interactive HUP mining. 1. (IHUPL-Tree) - incremental HUP lexicographic tree is arranged with the items in lexicographic order and introduced as a branch within the tree. 2. IHUPTF-Tree - The IHUP transaction frequency tree; here, tree nodes are organized in descending order as per their transaction frequency (TF) so as to items coming from several transactions can be placed in top of the tree, and consequently, higher prefix-sharing can be done. 3. IHUP transaction weighted utilization tree (IHUPTWU-Tree) is created in the transaction weighted utilization (TWU) value of items in descending order. IHUP generates a frequent pattern for all candidate item sets (favourable and unfavourable). [24] proposed a kHMC, an efficient algorithm for mining the top k high utility item sets, using novel threshold and pruning strategies, pruning strategies - RIU, CUD, COV, efficient in terms of memory and execution time to state of art algorithms. This requires proper trade off among high utility item sets, memory and running time. [26] Proposed HUIM, an efficient algorithm for extracting the high utility item sets from weblog data sets. They applied the cosine similarity method and high utility item sets tree structure to find

similar item sets from the web log data. It requires only two scans in the database; this still requires memory and running time optimization. The authors in [27] have performed efficient mining of high utility item sets from large database by their proposed algorithm CTU-PROL. Again, this algorithm also suffers from large computation time and it requires appropriate improvement in the algorithm to reduce the computation duration.

3. Preliminary terms and related definition

In this section, we describe the basic representations used for high utility webpage sets mining, including the concepts of the utility of an item set in the transaction of web log datasets.

Let *WP* = {wp1, wp2, wp3, wp4 ……. wpn} be a set of web pages. Each web page wpj ∈ WP has its own external utility, represented as EU. The external utility (EU) of a web page can be profit, cost, and other user-defined factors. A set $X \subseteq WP$ is called K-webpage set if X contains k webpages. A transactional weblog database *D*= {t1, t2, t3, t4…. tm} contains m transactions in which each transaction $tm \subseteq D$ is tuple containing: (1) a distinct identifier for transaction Tid; (2) a webpage set $Y \subseteq WP$ where each $wpj \in Y$ links with internal utility that means number of occurrences of wpj in tp and it is represented by *IU* (wpj, tp), Further, *if* $X \subseteq Y$, it is obvious to say that X occurs in transaction tp, i.e. transaction *tp* contains *X*.

Let's suppose *I*= {wp1, wp2, wp3,…, wpn} : a set of items (web pages) . Each transaction (T) has a unique identifier (Tid).

Def.1. *EU* (wpj): the external utility [4, 9, 15, 25] of a web page wp_j is the utility value in the transaction tp it may be profit, cost or any other user defined factor.

Def.2. *IU* (wpj): internal utility [4, 9, 15, 25] is the count value (quantity) associated with a web page (wp_j) in the transaction *tp*.

Def.3. *U* (wp_j,T): utility of a webpage *wpj* in transaction tp is defined as the product of *IU* (wp_j, t_p) and *EU* (wp_j) i.e.

$$U(wpj, tp) = IU(wpj, p) \times EU(wpj) \quad (1)$$

Def.4. The utility value of webpage set WP in transaction tp is the summation of utilities of all webpage sets $wpj \in WP$ that occurs in transaction tp, defined by

$$U(WP, tp) = \sum wpj \in WP \quad (2)$$

Where WP ∈ t_p U (wp_j, t_p).

Def.5. The utility value of webpage set WP in database D is the summation of utilities of WP that occur in transactions of D, and it is calculated as

$$U(WP) = \sum wpj \in tp \quad (3)$$

Where $t_p \in$ D U (WP, tp).

Def.6. The utility of a transaction TU [4, 9, 15, 25] (Td) is defined as the sum of the utilities of each webpage in transaction Td. For example, in Table 3 *TU* (T1) =18, *TU* (T2) =32 and the *TU* (T7) =8.

Def.7. The transaction weighted utilization (TWU) [4, 9, 15] of a webpage set X in a database D is defined as

$$TWU(X) = \sum_{X \subseteq Td \in D} TU(Td) \quad (4)$$

Def.8. Minimum threshold utility value (MTUV) is a value which is fixed by the user and it is dependent upon the total transaction utility [15].

$$MTUV = UDP \sum_{Td \in D} TU(Td) \quad (5)$$

Where UDP is a user defined percentile.

Def.9. X is a high transaction weighted utility webpage set only when *TWU*(X) > = *MTUV* else, it will be a low transaction weighted webpage set.

Def.10. Webpage set X is a high utility webpage set if *U* (X) ≥*MTUV*. High utility webpage set means, determining webpage set which satisfies the conditions of *U*(X) ≥*MTUV* [26].

Def.11. For some webpage set *X* = {wp_1, wp_2 ..... wpn} is a webpage set where L is number of webpage set and it should be greater than or equal to *2, for all* $2 \le n \le L$.

The Jaccard coefficient [27], which measures similarity between Patterns of webpage set P1 and P2 (say) can be calculated from the formula as given below –

$$J(P1, P2) = \frac{P1 \cap P2}{P1 \cup P2} \quad (6)$$

Here, the Jaccard Index (JI) / Jaccard Similarity (JS) coefficient [Jaccard]/ (initially termed as coefficient

Table 1. Horizontal slice of a typical pre-processed web log data

IP address	Timestamp	Access Request	Result status code	Bytes Transferred	Referrer	User agent
192.115.78.2	25/Apr/1998:03: 04:41--0300	GET wp1.html http/1.0	200	2077	wp_1	Mozilla/4.05
10.152.78.9	25/Apr/1998:03: 05:20--0300	GET wp2.html http/1.0	200	1234	wp_2	Safari/5.1.1
192.125.78.9	25/Apr/1998:03: 05:28--0300	GET wp.html http/1.0	200	1956	wp_1	Chrome/29.0.1547
10.116.178.50	25/Apr/1998:03: 05:41--0300	GET wp3.html http/1.0	200	2798	wp_3	Internet Explorer/10

de communaute by Paul Jaccard), is a statistical measure used for comparing the similarity and dissimilarity among various sample sets.

4. Utility quantization for items in web transaction database

Web log file where the web server automatically writes information each time a user requests a web site from that particular web site. This log file is located in places like web servers, web proxy servers, and client browsers. It is the primary source of data in Web usage mining [4]. Each hit against the server, corresponding to an HTTP request, generates a single entry in the server access logs. Each log entry (depending on the log format) may contain fields identifying the time and date of the request, the IP address of the client, the resource requested, possible parameters used in invoking a web application, status of the request, HTTP method used, the user agent (browser and operating system type and version), the referring web resource, and, if available, client-side cookies which uniquely identify a repeat visitor. A sample example of a server access log is shown in Table 1.

In web log data [28] set, there is no explicit transaction entry where we can directly fire queries to get the result out of the web log databases. It has to be pre-processed for the sessionization. The sessionization [29] is the process of segmenting the user activity record of each user into sessions, each representing a single visit to the site. In case of web log data, the sessionization is equivalent to the transaction as it occurs in transactional databases. Sessionization which uses heuristics fall into two basic categories: time-oriented or structure-oriented. Time-oriented heuristics apply either global or local time-out estimates to distinguish between consecutive sessions, while structure-oriented heuristics use either the static site structure or the implicit linkage structure captured in the referrer fields of the server logs. Time-oriented heuristics

Table 1. External utility of web pages

Items	wp1	wp2	wp3	wp4	wp5	wp6	wp7
Unit Profit	5	3	2	1	3	2	4

Table 3. Transactions table

Tid / Webpage	Internal Utility of Webpages in Transactions						
	T1	T2	T3	T4	T5	T6	T7
wp_1	1	1	2	1	1	1	0
Wp_2	0	0	1	2	1	2	1
Wp_3	2	1	1	2	2	0	0
Wp_4	2	1	2	3	2	0	1
Wp_5	1	6	3	1	2	4	0
Wp_6	0	0	1	5	0	0	0
Wp_7	1	0	0	0	0	3	1

apply either global or local time-out estimates to distinguish between consecutive sessions, while structure-oriented heuristics use either the static site structure or the implicit linkage structure captured in the referrer fields of the server logs. The utility of an item set depends not only on the support of the item set but also on the prices or weight of items in that item set [10]. Here, we will have to assign a predefined profitability and interestingness value to each and every item. Here, in Table 2, the predefined value of webpages wp1, wp2, wp3, wp4, wp5, wp6, wp7 for each web page is given. For calculating utility of more than one webpage, the sum value of that number of pages is calculated.

By Eq. (1), U (wpi, Td) =IU (wpi) x EU (pi, Td);

U (wp5, T3) = 3 x 3 =9.

By Eq. (2), U (WP) = $\sum$wpj $\in$tp, tp $\in$ D U (WP, tp)

WP= {wp1, wp2}.

U ({wp1, wp2}, T3) = U (wp1, T3) + U (wp2, T3)

=2 x 5 +1 x 3 = 13.

Suppose, Utility of webpage set WP in the given web database is WP = {wp1, wp2}

U (WP) = U ({WP, T3) + U (WP, T4) + U ({WP, T5) + U ({WP, T6) = 13+11+8+11=43.

Table 4. Transactions utility

Transaction	T1	T2	T3	T4	T5	T6	T7
TU	18	32	28	31	20	35	8

Table 2. Transactions weighted utilization

Items	wp_1	wp_2	wp_3	wp_4	wp_5	wp_6	wp_7
TWU	164	122	129	137	164	59	61

Different possible number of high utility item set can be obtained from the transaction Table 3.

Some of them are given following: {wp_1, wp_2}:48; {wp_2, wp_3}:25; {wp_1, wp_3, wp_3}:18; {wp_3, wp_4, wp_5}: 81; {wp_1, wp_2, wp_3, wp_4}: 67; {wp_1, wp_2, wp_3, wp_4, wp_5, wp_6}: 59; {wp_1, wp_2, wp_3}: 50; {wp_4, wp_5}:49, {wp_5, wp_6}:24;

Taking the minimum transaction utility (MTU) =30, we obtain the webpage sets of high utilities as given here-

{wp_1, wp_2}:48; {wp_3, wp_4, wp_5}: 81; {wp_1, wp_2, wp_3, wp_4}: 67; {wp_1, wp_2, wp_3, wp_4, wp_5, wp_6}: 59; {wp_1, wp_2, wp_3}: 50; {wp_4, wp_5}:49.

By Def. (6), the transaction utility (TU) for each transaction is calculated. In Table 3 and Table 4, the *TU* (T1) =18, *TU* (T2) =32. Similarly, for T3, T4, T5, and T6 the transaction utility is calculated. The table also contains the TWU for each webpage available in web databases.

According to Eq.(4). The calculated value of *TWU* (wp1) =*TU* (T1) +*TU* (2) +*TU* (3) +*TU* (4) +*TU* (5) +*TU* (6) =18 +32+28+31+20+35=164. The TU value of T7 is not included because the wp1 has not occurred in T7. Likewise, *TWU* (wp2) =122, *TWU* (wp3) =129, *TWU* (wp4) = 137, *TWU* (wp5) = 164, *TWU* (wp6) =59, *TWU* (wp7) = 61; This TWU value is helpful in arranging the web transaction in logically sorted order to prune the search space and segregating the high utility transaction.

5. Methodology for high utility webpage sets mining

Terminology used in algorithm high utility webpage sets mining (HUWSM): Transaction Utility: TU, Transactional Weighted Utilization: TWU, High Utility Webpage Set- Frequent Pattern: HUW-FP, Frequent Pattern: FP, Jaccard Similarity Pattern: JSPT, Transaction Table: Td, Database: DB, Minimum Threshold Utility: (MTU), Minimum Jaccard Similarity Value: MJSV.

The procedure for high utility webpage sets mining form weblog data proceeds as following shown in the Fig. 1. The very first step is the

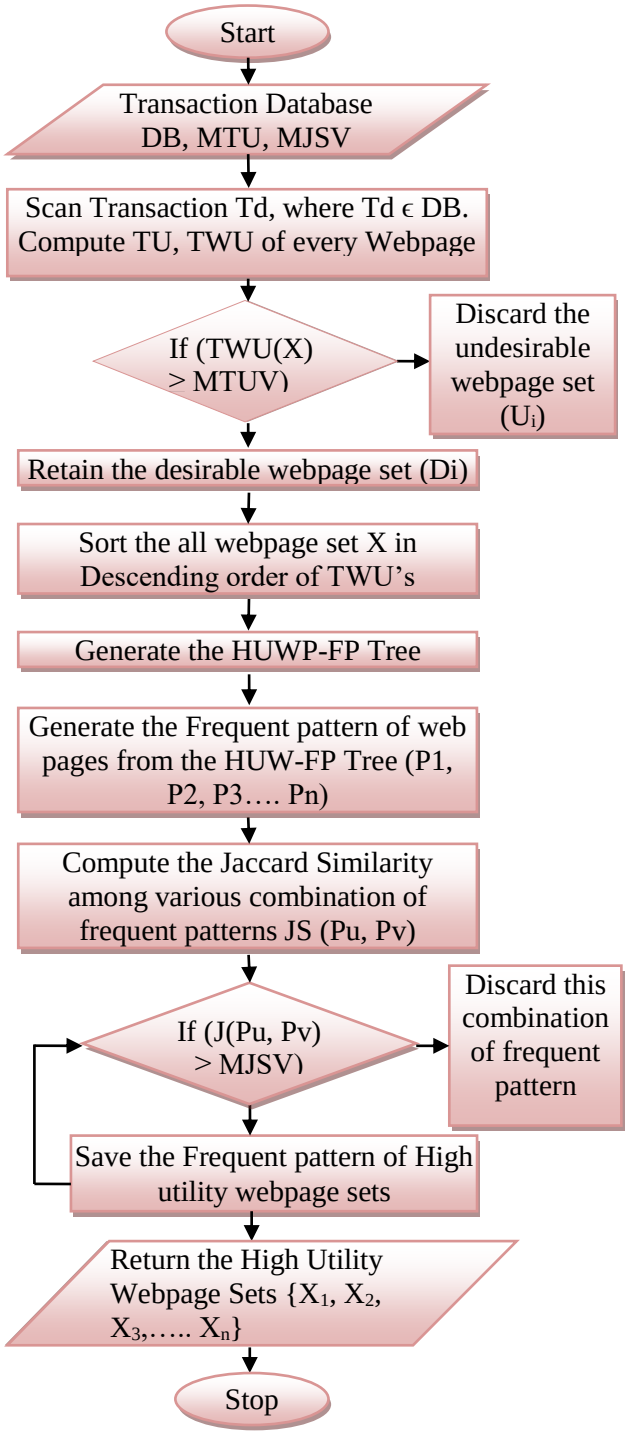

Figure.1 Methodology: high utility webpage sets mining (HUWSM)

collection of web log data sets form web servers to form a web log database. The second steps is the pre-processing of web log data in desired form so that the mining process can be applied efficiently and effectively over log data and it has been discussed in previous section.

The step – III calculates numerical value of TU

ITEM	TWU	LINK
Wp1	164	
Wp3	164	
Wp5	139	
Wp4	137	
Wp2	122	

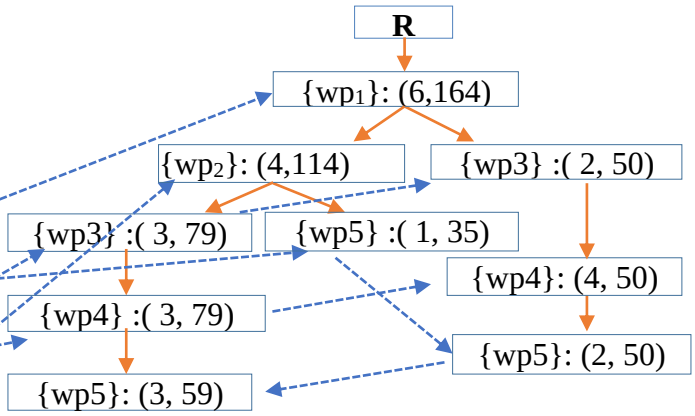

Figure.2 HUW-FP tree for webpages

and TWU using the IU and assigned EU [16] values in the web log database. The job of the next step –IV is to eliminate the undesirable webpages that has TWU [11, 20] less than the MTUV. Now, it results in reduced weblog database space having only desirable webpages. Further, the step-V builds the high utility webpage frequent pattern tree from the favourable webpages obtained in the previous step IV. To construct the HUW-FP [25] Tree, firstly, all the webpages are arranged in the descending order of their TWU values and the webpages of the lowest TWU [26] value is excluded from the computation work. The tree is helpful in finding the information about the webpages and their utilities. The HUW-FP Tree starts with a node called R which references to the root node of the tree. Each node of the HUW-FP holds the three piece of information webpage name; support count value and TWU value [24]. The tree spawns entire possible patterns with help of support count value and TWU value [25]. On contrary to the single parameter, it uses two parameters for generating more favourable item. The benefit of this tree, it can generate complete traversal pattern starting from the root of HUW-FP tree to the leaf node. Moreover, it can also pop up the common navigation patterns among webpages. The tree also slashes the search space and time complexity because in this approach the candidate webpage sets can been proficiently produced with merely two scans of the web database. Moreover, the step –VI, refines the frequent pattern results obtained in previous step –V, the task is carried out by calculating the jaccard similarity among frequent pattern webpage sets. Here, the patterns with jaccard similarity more than or equal to MJSV=40% is retained besides this discarded from the webpage set. The last step-VII, which finally, extracts the high utility webpage sets.

5.1 Jaccard coefficient / Jaccard similarity (JS)

Patterns: P1= {wp1, wp2, wp3, wp4, wp5}, P2= {wp1, wp2, wp5}, P3= {wp1, wp3, wp4, wp5}, P4= {wp1, wp2, wp6, wp7, wp8}
Patterns - P1 and P3
JS (P1, P2) = (P1 ∩ P2)/ (P1 ∪ P2) =3/5
JS (P1, P2) =0.60 = 60% (Pattern accepted).
Similarly, JS for various other patterns
Patterns - P1 and P3
JS (P1, P3) = (P1 ∩ P3)/ (P1 ∪ P3) =4/5 = 0.80 = 80% (Pattern accepted).
Patterns- P2 and P3
JS (P2, P3) = (P2 ∩ P3)/ (P2 ∪ P3) =2/5 = 0.40 = 40% (Pattern accepted).
Patterns- P2 and P4
JS (P2, P4) = (P2 ∩ P4)/ (P2 ∪ P4) =2/6 = 0.33 = 33% (Pattern rejected).

6. Experimental evaluation

The comprehensive experimental evaluation of the proposed algorithm is performed on a 3.00 GHz Intel Core i5-7400T Processor, 6M Cache with 4 GB RAM and Microsoft Windows 8.1 operating system. The algorithm is implemented in Java programming language and on software's - JDK 1.8.0_60 and NetBeans IDE 8.0.2. The web log dataset [30] used for this experiment is taken from the NASA-HTTP Kennedy Space Centre World Wide Web server in Florida. The weblog data is pre-processed and arranged as shown in Table 1. For the sake of convenience and fast execution of algorithm, for each URL of web log the integer value is chosen as a token value. The reason is that the computation over integer is much easier than other data type. The experiment is conducted by changing the minimum utility threshold value in percentage on ten thousand of the web transactions. The proposed algorithm

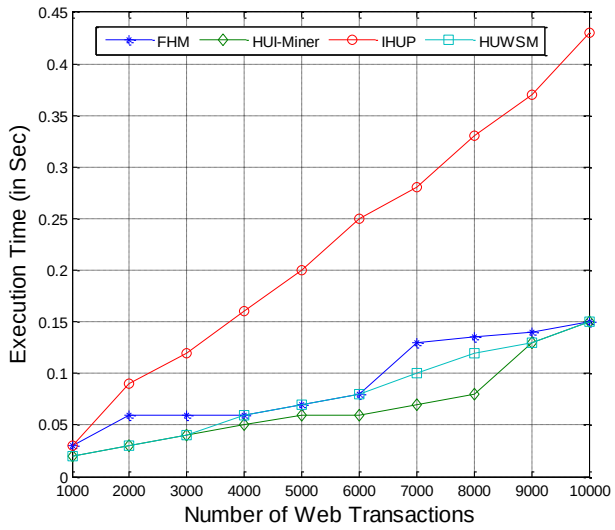

Figure 3. Execution time (Sec) with respect to no. of web transactions

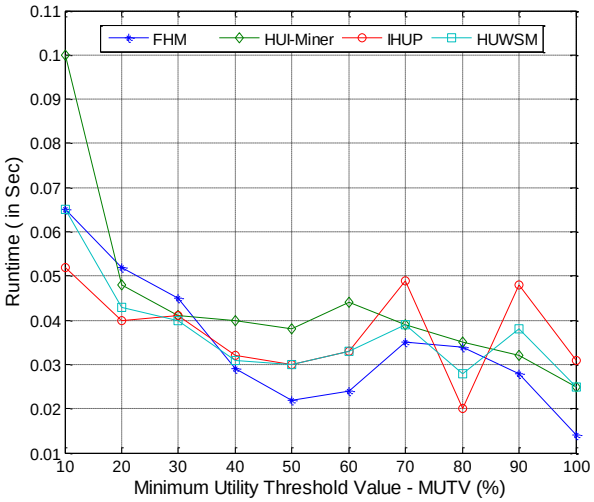

Figure 4. Execution time comparisons on different minimum utility threshold values

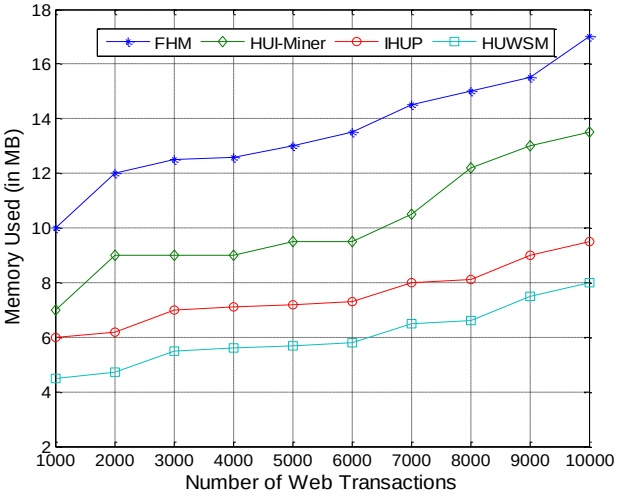

Figure.5 Memory utilization v/s no. of web transactions

HUWSM is executed and compared with different existing algorithms FHM [22], IHUP [17], HUI-Miner [25] algorithms. Fig. 3 illustrates an execution time for different number of web transactions. To maintain the uniformity in evaluation criteria, the experimental evaluation is done keeping the minimum threshold value at 30. Under the given hardware and software environment, the rate of growth of execution time for IHUP algorithm is highest among all FHM, HUI-Miner, and HUWSM. IHUP also takes the highest execution time. Other algorithms have less growth rate of execution time. It is obvious from the Fig. 3 that the running time of HUWSM is comparatively lesser than other two algorithms IHUP and FHM. The HUI-Miner and HUWSM show almost similar behaviour in terms of execution time. This decrement in execution time is due to removal of unfavourable webpage sets. Fig. 4 shows the comparison of execution time taken by these algorithms when different minimum threshold utility values are set. It is pretty clear from the figure that the minimum threshold value decreases the runtime is increased but more number of web transactions is processed. Runtime decreases as the MTUV increases [26]. Here, HUWSM algorithm shows significant improvement over other algorithms.

The algorithms HUWSM outperforms the IHUP, FHM, and HUI-Miner, in memory consumption it avoids dealing over a large number of web data. Fig. 5 demonstrates memory consumption with different number of web transactions while keeping minimum utility threshold value at 30. Among the existing algorithm FHM, IHUP, HUI-Miner algorithms, the new technique HUWSM has the lowest memory consumption. This algorithm consumes 6 MB of storage when dealing with 6000 of web transaction. Whereas IHUP, HUI-Miner and FHM algorithm takes 7.0 MB, 9.5 MB, 13.5 MB of memory space respectively. This clearly shows that the memory usage is more in these algorithms. Hence, this is efficient in memory usage in dealing with web log database. Fig. 6 demonstrations a number of high utility webpage sets produced on different number of web transactions. For this purpose, this section of exercise is done to find how many numbers of high utility webpages will be obtained when minimum threshold value is kept at 30 and results were obtained on different number of web transactions. Here, the IHUP generates the highest number of high utility webpages; comparatively other three algorithms generate small number of high utility webpage sets and give almost similar number of high utility webpage sets. These generated high

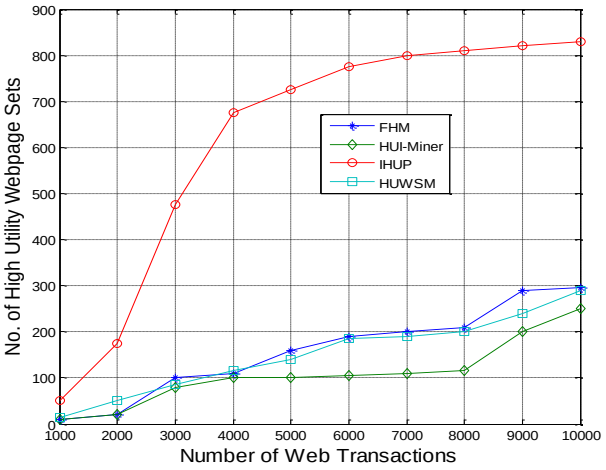

Figure 6. Number of high utility webpage sets v/s the no. of web transactions

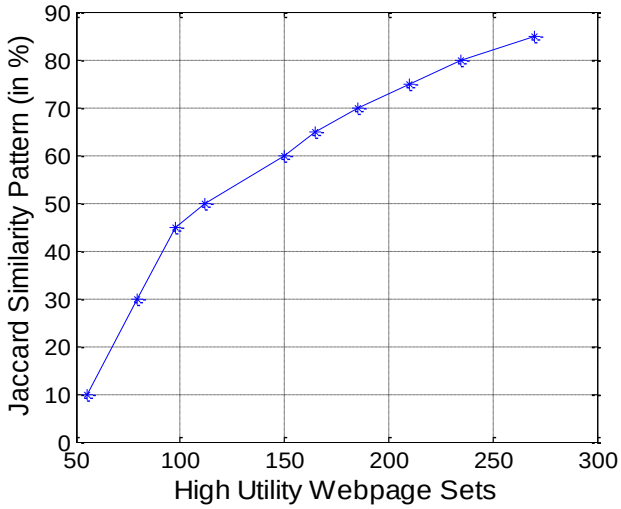

Figure.7 % Jaccard similarity (JS) v/s no. of high utility webpage sets

utility webpages may or may not be similar among them. Now further, it is intended to find the favourable and similar high utility webpage sets. So, Jacaard Similarity (JS) process is applied here, it keeps the Minimum Jaccard Similarity Value (MJSV) at 60%. The Fig. 7 displays numbers of high utility webpage sets and their jaccard similarity of HUWSM algorithm. Thus, the high utility webpage sets obtained are much more similar among them all. The experiment is conducted by changing the minimum utility threshold value in percentage on ten thousand of the web transactions. The proposed algorithm HUWSM is executed and compared with different existing algorithms-FHM, IHUP, and HUI-Miner.

7. Conclusion

The method proposed in this paper i.e. HUWSM applies the concept of internal utility and external utility, transaction utility (TU) and transaction weighted utilization (TWU) to identify the high utility webpage sets. Moreover, HUW-FP tree is proposed to store the frequent patterns of webpages due to this, potentially high utility webpages can be generated with only two scans in the database. In addition to that, pattern generation technique with 'Jaccard Similarity' finds more accurate frequent and high utility item sets. Experimental results have demonstrated that HUWSM outperforms the FHM, IHUP, HUI-Miner algorithms in terms of both running time and memory usage, and number of high utility webpages. These discovered high utility webpage sets are very helpful in various commercial outlook of e-business, valuable for advertisement management, web site optimization for better service, tourist guidance, and value analysis of a product and useful for exploring user behaviour similarity and their activity performed over the website. For future work, we would like to develop algorithm based on soft cosine similarity and soft Jaccard similarity method to examine further improvement in result and performance of HUWSM.

References

[1] O. Etzioni, "The World Wide Web: Quagmire or Gold Mine." *Communications of the ACM*, Vol. 39, pp. 65–68, 1996.

[2] M. Zdravko and T. L. Daniel, *"Data Mining the Web, Uncovering Patterns In Web Content, Structure and Usage",* John Wiley & sons Inc., New Jersey, USA, pp. 115–132, 2007.

[3] R. Cooley, B. Mobasher, and J. Srivastava, "Web Mining: Information and Pattern Discovery on the World Wide Web", In: *Proc. of the 9th IEEE Intelligent Conf. on Tools with Artificial Intelligence*, pp. 558–567, 1997.

[4] B. Mobasher, *Web Usage Mining*, In John Wang (eds.), Encyclopedia of Data Warehousing and Mining, Idea Group, 2006.

[5] C. F. Ahmed, S. K. Tanbeer, and B. S. Jeong, "A Framework for Mining High Utility Web Access Sequences", *IETE Technical Review*, Vol. 28 No.1, 2011.

[6] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules", In: *Proc. of the 20th Conf. on VLDB*, San Francisco, pp. 487-499, 1994.

[7] R. Cooley, P. N. Tan, and J. Srivastava, "Discovery of Interesting Usage Patterns from

web Data", *International Workshop on Web Usage Analysis and User Profiling*, ISBN 3-540-67818-2, pp. 163-182, 2000.

[8] R. Chan, Q. Yang, and Y. Shen, "Mining High Utility Itemsets", In: *Proc. of the 3rd IEEE International Conf. on Data Mining*, Washington, pp. 19-26, 2003.

[9] U. Yun and J. J. Leggett, "WIP: Mining Weighted Interesting Patterns with a Strong Weight and/or Support Affinity", *Information Science*. Vol. 177, No.17, pp. 3477-3499, 2007.

[10] J. C. Lin, T. Li, P. Fournier-Viger, T. P. Hong, J. Zhan, and M. Voznak, "An Efficient Algorithm to Mine High Average-Utility Itemsets", *Advanced Engineering Informatics, Elsevier*, Vol. 30, No. 2, pp. 233-243, 2016.

[11] H. Ryang and U. Yun, "Top-K High Utility Pattern Mining With Effective Threshold Raising Strategies", *Knowledge-Based Systems* Vol. 76 pp. 109–126, 2015.

[12] S. Shankar, T. Purusothaman, S. Jayanthi, and N. Babu, "A Fast Algorithm For Mining High Utility Itemsets", In: *Proc. of IEEE International Advance Computing Conference*, Patiala, pp. 1459-1464, 2009.

[13] V. Tseng, B. Shie, C. Wu, and P. Yu, "Efficient Algorithms for Mining High Utility Itemsets from Transactional Databases", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 25, No. 8, pp. 177-186, 2013.

[14] V. Tseng, C. Wu, P. Fournier-Viger, and P. S. Yu, "Efficient Algorithms For Mining The Concise and Lossless Representation of Closed+ High Utility Itemsets", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 27, No. 3, pp. 726–739, 2015

[15] V. Tseng, C. Wu, P. Fournier-Viger, and P. S. Yu, "Efficient Algorithms for Mining Top-K High Utility Itemsets", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 28, No. 1, pp.54-67, 2016.

[16] S. Patel and B. Madhushree, "A Survey on Discovering High Utility Itemset Mining from Transactional Database", *Information and Knowledge Management*, Vol.5, No.12, 2015.

[17] M. Liu and J. Qu, "Mining High Utility Itemsets without Candidate Generation", In: *Proc. of the 21st ACM International Conference on Information and Knowledge Management*, New York, pp. 55-64, 2012.

[18] H. Yao and H. J. Hamilton, "Mining Itemset Utilities from Transaction Databases", *Data & Knowledge Engineering*, Vol. 59, No. 3, pp. 603- 626, 2006.

[19] A. Erwin, R. P. Gopalan, and N.R. Achuthan, "A Bottom-Up Projection Based Algorithm for Mining High Utility Itemsets", In: *Proc. of the 2nd international workshop on Integrating Artificial Intelligence and Data Mining*, Vol. 84, pp. 3-11, 2007.

[20] V. Tseng, C. Wu, P. Fournier-Viger, and P. S. Yu, "Efficient Algorithms for Mining Top-K High Utility Itemsets", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 28, No.1 pp. 54-67, 2016.

[21] V.S. Tseng, C.W. Wu, B.E. Shie, and P.S. Yu, "UP-Growth: An Efficient Algorithm for High Utility Itemset Mining", In: *Proc. of ACM-Knowledge Data Discovery*, Washington, DC, USA, pp. 253-262, 2010.

[22] P. Fournier-Viger, C. Wu, S. Zida, and V.S. Tseng, "FHM: Faster High-Utility Itemset Mining Using Estimated Utility Co-Occurrence Pruning", *Foundations Intelligent System*, Vol. 85, No. 2, pp. 83-92, 2014.

[23] C. F Ahmed, S. K.Tanbeer, J. Byeong-Soo, and L. Young-Koo, "Efficient Tree Structures For High Utility Pattern Mining In Incremental Databases", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 21, No. 12, pp. 1708-1721, 2009.

[24] Q. H. dong, B. Liao, P. Fournier-Viger, and T. L. Dam ,"An Efficient Algorithm For Mining The Top K High Utility Itemsets Using Novel Threshold Raising and Pruning Strategies", *Knowledge Based Systems*, Vol. 104, pp.106-122, 2016.

[25] S. Shankar, T. Purusothaman, and S. Jayanthi, "Novel Algorithm for Mining High Utility Itemsets", In: *Proc. of International Conf. On Computing, Communication and Networking, IEEE*, pp. 1-6, 2008.

[26] B. Bakariya, and G S Thakur, "An Efficient Algorithm For Extracting High Utility Itemsets From Web Log Data", *IETE Technical Review*, Vol. 32, No. 2, pp.151-160, 2015.

[27] S. Yen and Y. Lee, "Mining High Utility Quantitative Association Rules", In: *Proc. of the 9th International Conf. on Data Warehousing and Knowledge Discovery, Regensburg*, pp. 283-292, 2007.

[28] K. Joshila Grace, V. Maheswari, and D. Nagamlai, "Web Log Data Analysis and Mining", *Advanced Computing Communications in Computer and Information Science*, Vol. 133, pp. 459-469, 2011.

[29] D.A. Adeniyi, Z. Wei, and Y. Yangquan, "An Automated Web Usage Data Mining and Recommendation System Using K-Nearest

Neighbour (KNN) Classification Method", *Applied Computing and Informatics*, Vol. 12. pp. 90-108, 2016.
[30] Weblog dataset downloaded from *http://ita.ee.lbl.gov/html/contrib/NASA-HTTP.html*(Retrieved on March 12th, 2017)