

International Conference on Communication, Computing, and Security [ICCCS-2012]

Efficiently Mining of Effective Web Traversal Patterns With Average Utility

M. Thilagu^a, R. Nadarajan^b^a Sri Krishna College of Technology, Coimbatore, India^b PSG College of Technology, Coimbatore, India

Abstract

Mining of frequent web traversal patterns finds pages that co-occur in an order in a transaction. Patterns only with frequency would not provide sufficient information about the preferences or interestingness of users. By adopting utility-based mining model in web traversal pattern mining, users' interestingness of patterns could be found by considering time spent on the web pages as utility or user preferences. However, an issue identified in the existing utility-based web traversal pattern mining algorithms is that transaction weighted utility of a pattern is computed considering utilities of all transactions in which it exists and thus includes utilities of all patterns occurring before the pattern, irrespective of its prefixes in those transactions. This leads to generation of more candidate patterns unnecessarily and affects the efficiency of the algorithm. Another issue is that, if the length of a pattern increases, automatically its utility will also increase. With the effect of pattern length, really good utility patterns cannot be identified among all high utility patterns. Also, longer patterns with less page utility in a transaction may result in higher values and it will be treated same as shorter patterns with more page utility. By considering high average utility patterns rather patterns with only actual utility, we could distinguish high utility patterns with respect to their lengths. Moreover, a high average utility pattern would be more effective than a high utility pattern and we could reveal better results by finding effective traversal patterns in a database. With this motivation, an efficient algorithm addressing the above said issues has been proposed to discover effective web traversal patterns in a transaction database. Experiments results on real datasets show that the proposed algorithm is more efficient than an existing approach.

© 2012 The Authors. Published by Elsevier Ltd. Selection and/or peer-review under responsibility of the Department of Computer Science & Engineering, National Institute of Technology Rourkela. Open access under [CC BY-NC-ND license](#).

Keywords: Web Usage Mining, Web Traversal Sequence, Utility-Based Mining, High Utility Pattern, Effective Web Traversal Pattern, Projected Transaction Weighted Utility

1. Introduction

Web traversal pattern mining is one of the applications in web usage mining. In web usage mining, the mining process is performed on users web access sequences stored in web logs to find the web user navigation or traversal patterns. These patterns are useful to improve web site design, web recommendations and system performance etc [B. Mobasher et al., 1996, J. Srivastava et al., 2000]. A web access sequence or traversal sequence is a list of web pages associated with timestamp in order and mostly forward references of pages are considered for mining purpose. Algorithms used for mining traversal patterns treat all the web pages equally in a database by only considering if a web page exists in a transaction or not. However, if web developers want to know how a traversal sequence is important or interesting to end users or whether users have any preferences on a traversal sequence, then

it cannot be answered with frequency of a pattern alone. To address this problem, utility-based mining model in web traversal patterns has been introduced, in which utility of a web page is the browsing time that a user spent on it. Frequent and utility-based web traversal pattern mining algorithms discussed in the literature, discover patterns either based on Apriori or pattern-growth approach. Methods adopting Apriori approach [R. Agrawal and R. Srikant, 1995], suffer from level-wise candidate generation-and-test methodology and require more database scans. To avoid candidate generation and multiple scans, a tree based pattern-growth approach has been proposed to represent web traversal sequences in a compressed form called WAP-tree and patterns are recursively mined from the tree [J. Pei et al., 2000]. In this approach, patterns are mined by constructing conditional trees recursively and it becomes a tedious task compared to the previous approach. Algorithms based on PrefixSpan [J. Pei et al., 2004], a pattern-growth approach reduces the number of candidate patterns by generating them in a projected or reduced database. However, the construction cost for projected databases is high and also requires multiple scans.

Here, we focus on utility-based web traversal pattern mining approaches and define high utility web traversal patterns as follows. Let $P = \{P_1, P_2, \dots, P_n\}$ be a set of web pages in a website. A web access or traversal sequence $S = \langle P_1, P_2, \dots, P_m \rangle$ ($P_i \in P$, $1 \leq i \leq m$) or a set of web pages ordered based on timestamp in a sequence. A web traversal sequence may be a simple traversal pattern with only forward references or non-simple traversal pattern with both forward and backward references. A backward reference is revisiting of a page by the same user in a session. A web log database D is a collection of web traversal sequences, that is $D = \{S_1, S_2, \dots, S_n\}$, where each web access or traversal sequence $S_i \in D$ is a subset of P . In utility-based mining model, a web traversal sequence is represented as $S = \langle P_1(u_1), P_2(u_2), \dots, P_n(u_n) \rangle$, where each page P_i is associated with its utility u_i . The utility of a traversal sequence α is sum of utilities of web user sequences containing α in the database. A traversal sequence α is a high utility web traversal pattern if actual utility (α) \geq min-utility ϕ , in which min-utility is a user-specified minimum utility threshold. By applying utility-based mining model with transaction weighted utilization concept, the existing utility-based web traversal pattern mining methods [Zhou et al., 2007, C.F. Ahmed et al., 2009, C. F. Ahmed, S. K. Tanbeer, B. S. Jeong, 2011] maintain downward closure property and discover high utility patterns. However, we identified an issue relating to computation of transaction weighted utility of patterns and it is to be addressed for effective computation. Here, the transaction weighted utility is defined as sum of utilities of items in a transaction and it is uniformly shared by all items in that transaction. Because of this, transaction weighted utility of a prefix based pattern includes the utilities of patterns which are not part of their prefixes. For instance, consider a web traversal sequence $\langle a(2), b(3), c(2), d(1) \rangle$ with transaction utility 8. From this sequence, if we generate patterns prefixed with 'a' 'b' 'c' 'd' as $\langle abcd \rangle$ $\langle bcd \rangle$ $\langle cd \rangle$ $\langle d \rangle$ and then all patterns will be having the same transaction utility as 8, irrespective of their prefixes. Thus, they all would become candidate patterns if the given min-utility δ is say 5. To overcome this problem, we consider utilities of pages only in the projected sequences, so that the utility of a pattern will not include utilities of all its prefix patterns. Moreover, the utility value of the pattern will get minimized. That is, for each pattern prefixed with page P_i , we compute its transaction utility in the projected sequences prefixed with P_i . Considering the same above illustration, if we compute transaction utilities of patterns $\langle abcd \rangle$ $\langle bcd \rangle$ $\langle cd \rangle$ $\langle d \rangle$ in their projected sequences, then they would be 8, 6, 3 and 1. Now, the patterns $\langle abcd \rangle$ and $\langle bcd \rangle$ would only satisfy the given min-utility $\delta=5$. As a result, it is efficient to compute transaction utilities of a pattern in the projected sequence rather considering utilities in an entire sequence and leads to reduction in the number of candidate patterns. Another issue with the existing approaches is that all patterns satisfying the user specified minimum utility threshold (min-utility) would become high utility patterns, regardless of their pattern length. Thus, if the length of a pattern increases, automatically utility of the pattern will also increase. Moreover, longer patterns with less page utility in a transaction may result in higher values and it will be treated equally with shorter patterns with more page utility. Hence, we could not reveal better results with the effect of pattern length. To resolve this problem, we use high average utility concept introduced by Tzung-Pei et al., 2009 in our work. The average utility (avg-utility) of a pattern α could be defined as the actual utility of the pattern divided by its length. The length of a web traversal pattern $|\alpha|$ is the number of pages in the pattern if it has only forward references; otherwise it is the number of distinct pages in the pattern if the pattern has both forward and backward references. A traversal sequence α is a high average utility pattern if avg-utility (α) \geq min-avg-utility δ , in which min-avg-utility is a user-specified minimum average utility threshold. To accomplish the above discussed solutions, an efficient algorithm has been proposed and discussed below. The rest of the paper is organized as follows: In Section 2, we introduce the related work of the proposed algorithm. In Section 3, we give out the definitions of utility-based mining model and define the problem statement. Section 4 describes the working principle of the proposed

algorithm with an illustration. Section 5 presents the experimental results and performance evaluation of the proposed algorithm. We conclude and summarize our work in Section 6.

2. Related Work

In the past decade, several researchers have contributed in the area of web usage mining, to find interesting patterns from web access sequences. Most of the existing algorithms [M.S. Chen et al., 1998, J. Pei et al., 2000, Y.S. Lee and S.J. Yen, 2008] discuss on mining of frequent web traversal patterns with pages that occur together. Recently, algorithms based on utility mining model have been proposed to mine high utility web traversal patterns that reveal the user interestingness or preferences better compared to patterns with frequency alone. Web path traversal mining model with utility concept was first introduced by Zhou et al., 2007. The algorithm uses the definitions of utility from the HUP mining model [H. Yao, H. J. Hamilton, and C. J. Butz, 2004] and the browsing time of a user is used as an internal utility of a web page. However, this work is based on the Two-Phase HUP mining algorithm proposed by Y.Liu et al., 2005 which suffers in the level-wise candidate generation-and-test methodology of the Apriori algorithm [R. Agrawal and R. Srikant, 1995]. The Two-Phase algorithm was developed based on the definitions of Yao et al., 2004 and 2006, to find high utility itemsets using the downward closure property of Apriori. Therefore, it generates too many candidate patterns, and needs several database scans to discover the resultant web traversal sequences. An algorithm EUWPTM (Efficient Utility-based Web Path Traversal Mining) [C.F. Ahmed et al., 2009] based on the pattern-growth sequential pattern mining approach [J. Pei et al., 2004] was proposed to discover the web traversal patterns very efficiently by recursively dividing the search space using divide and conquer technique. It reduces the number of candidates by a huge amount and avoids the several database scans needed by the previous algorithm. However, this algorithm computes the transaction weighted utility of patterns by considering the utilities of all pages in a transaction even if the transaction is a projected one. The above discussed algorithms consider only internal utility of web pages in a sequence and sequences with only forward references. A framework for mining high utility web access sequences algorithm [C. F. Ahmed, S. K. Tanbeer, B. S. Jeong, 2011] was proposed to discover high utility sequence patterns with internal and external utilities of pages. This algorithm considers web access sequences with both forward and backward sequences and allows gap between pages. It uses two tree structures called UWAS-tree and IUWAS-tree for the mining process in order to support incremental and interactive mining. However, in all the above discussed methods, transaction utility of a pattern shares utilities of all its prefix patterns in a transaction and if the length of a pattern increases, automatically its utility also increase and thus leading to generation of unnecessary candidate patterns.

3. Problem Definition and Illustration

In this section, we first discuss the terms and definitions of utility-based mining model as given in previous works [L. Zhou et al., 2007, C.F. Ahmed et al., 2009]. To maintain downward closure property, utility-based web traversal mining adopts transaction weighted utilization concept introduced in [Y. Liu et al., 2005]. Here, the transaction weighted utility of a sequence is the maximum possible value or upper bound, so that any super-sequence of pattern X cannot be a high utility sequence if X is not a high utility sequence satisfying the given min-utility.

3.1 Basic Terms and Definitions

Definition 1: The quantitative measure of utility for event P_i in sequence S_j , denoted as $u(P_i, S_j)$ and it is defined by

$$u(P_i, S_j) = iu(P_i, S_j) \times eu(P_i) \quad (1)$$

In utility-based mining, internal utility $iu(P_i, S_j)$, represents the utility value of event P_i in sequence S_j and external utility $eu(P_i)$ is the unit profit value of event P_i . In our work, we consider only internal utility of pages by treating all pages with same significance.

Definition 2: A sequence $X = \{P_1, P_2, \dots, P_k\}$ is called a k -sequence, where $X \subset S_j$, $P_j \subset P$, and $1 \leq i \leq k$ and its utility in a sequence S_i is defined by

$$u(X, S_i) = \sum_{P_i \in X} u(P_i, S_i) \quad (2)$$

Definition 3: The utility of a sequence X in a web log database D is defined by

$$u(X, D) = \sum_{X \in S_i, S_i \in D} u(X, S_i) \quad (3)$$

Definition 4: The average utility of a sequence X is the total actual utility of X in the database divided by length of the sequence X , denoted as $|X|$ and defined as

$$au(X) = u(X, D) / |X| \quad (4)$$

Definition 5: The utility of a sequence S_i is defined by

$$u(S_i) = \sum_{P_i \in S_i \in D} u(P_i, S_i) \quad (5)$$

Definition 6: The transaction weighted utility of a sequence X in a web log database D is the sum of the transaction utilities of all sequences containing X and it is defined as

$$twu(X, D) = \sum_{X \subseteq S_i \in D} u(S_i) \quad (6)$$

Definition 7: The average transaction weighted utility ($atwu$) of a sequence X is the total transaction utility of X in the database D divided by the length of sequence X , denoted as $|X|$ and it is defined as

$$atwu(X) = twu(X, D) / |X| \quad (7)$$

Definition 8: The minimum average utility(min-avg-utility) threshold δ is given by the percentage of the total utility value of the web log database, similar to min-utility.

$$\text{min-avg-utility} = \delta \times \sum_{S_i \in D} u(S_i) \quad (8)$$

Definition 9: A sequence X is an effective traversal pattern if $au(X)$ is greater than or equal to min-avg-utility.

3.2 Problem Statement

Given a web log database D , the problem is to find effective web traversal patterns satisfying the user specified min-avg-utility δ . In our problem, we considered traversal patterns with forward references and maintaining contiguous property between pages.

4. Mining of Effective Web Traversal Patterns

The proposed algorithm is based on the pattern-growth approach applied in [C.F. Ahmed et al., 2009]. The existing algorithm is an efficient one by searching for patterns in the minimized search space. However, it considers the entire transaction utility for computing transaction weighted utility of any pattern in a transaction. This leads to generation of more candidate patterns and affects the efficiency the algorithm. To avoid this, we compute projected transaction weighted utility of each pattern in the divided search space and minimize the number of candidate patterns. Now, we explain the working principle of our algorithm in detail as follows. Initially, the algorithm finds transaction weighted utility (twu) and projected twu of each web page (1-sequence) in the database by scanning it once. Then, the algorithm discovers high $atwu$ 1-sequence which is same as high twu 1-sequence, because the length of the 1-sequence is one. For each high $atwu$ 1-sequence α with its average projected twu (same as projected twu) satisfying the given min-avg-utility, the algorithm divides the search space prefixed with α and performs the following steps recursively. At first, it finds all high average twu 1-sequence β in the α -projected database and appends them with α to generate candidate patterns. The projected database is further divided and projected with the newly generated candidate patterns treated as α . For instance, the database is firstly partitioned into a search space prefixed with web page 'a'. And, all candidate patterns prefixed with 'a' are recursively generated in the projected databases. In our algorithm, a pattern with twu not satisfying the given min-avg-utility is completed pruned to become a part of a high utility pattern and a pattern with projected twu not satisfying the given min-avg-utility is pruned in generating prefix based patterns. Like the above said process, we repeat the steps recursively to generate all candidate patterns prefixed with each high twu and high projected twu web page. Finally the database is scanned once to compute the actual utilities of all the generated candidate patterns. Then, the actual average utility of the

patterns is computed and those patterns with average utility satisfying the given min-avg-utility δ would become effective patterns. We give out the mining process steps in Section 4.1 and candidate generation algorithm in Section 4.2.

4.1 Steps of the Mining Process

Input: A Weblog database D , min-avg-utility δ

Output : Effective Web Traversal Patterns

```
//  $\alpha$  : a sequential pattern,  $l$ : the length of  $\alpha$  ,  $D | \alpha$ : the  $\alpha$ -projected database, if  $\alpha \neq \langle \rangle$ ;
otherwise, the web log database  $D$ 
1. Scan the database  $D$  once, compute  $twu$  and projected  $twu$  of 1-sequence
2. Find all high average  $twu$  1-sequence
3. For each high average  $twu$  1-sequence  $\alpha$ 
   If average projected  $twu$  of  $\alpha \geq$  min-avg-utility  $\delta$ 
   call candidate-gen( $\alpha$  ,  $l$ ,  $D | \alpha$  )
4. Scan the database  $D$  once again to compute actual and average utilities of all candidate patterns
5. Return all patterns with actual average utility satisfying the given min-avg-utility  $\delta$  as effective patterns
```

4.2 Candidate generation algorithm

Input: Projected Database D , min-avg-utility δ

Output : Candidate Patterns

Procedure candidate-gen(α , l , $D | \alpha$)

Method:

1. Scan the projected database $D | \alpha$ and compute twu of the set of high average sequences of length-1 that could be appended with α to form a sequential pattern set (SP)
2. For each sequential pattern $\beta \in SP$
 - a. compute average twu of β
 - b. If average twu of $\beta \geq$ min-avg-utility δ , then output β
3. For each candidate pattern β , construct β -projected database $D | \beta$ and call candidate-gen(β , $l+1$, $D | \beta$)

4.3 An Illustration

Consider a web log database containing traversal sequence of web pages associated with their utility as given in Table 1. Each transaction represents the browsing behaviour or web access pattern of a user. Assume that the user specified min-avg-utility = 6 (10%).

Table 1. Web Traversal Sequences

TID	Traversal Sequence	Transaction Utility
1	a(2), b(3), c(5), d(3), e(3)	16
2	a(1), c(4), e(3), d(2), f(2)	12
3	b(3), c(2), e(1), f(1)	7
4	a(5), c(3), f(3), g(1)	12
5	a(4), c(5), e(2), d(2)	13

As per the algorithm steps, we first scan the database once and arrive twu with projected twu of 1-sequence as $\langle a:53,53 \rangle$, $\langle b:23,21 \rangle$, $\langle c:60,42 \rangle$, $\langle d:41,12 \rangle$, $\langle e:48,16 \rangle$, $\langle f:31,7 \rangle$ and $\langle g:12,1 \rangle$. In this case, all the above 1-sequences with twu satisfy the given min-avg-utility and become high average transaction weighted utility 1-sequence. However, high average projected transaction weighted utility 1-sequences are $\langle a \rangle$, $\langle b \rangle$, $\langle c \rangle$, $\langle d \rangle$, $\langle e \rangle$ and $\langle f \rangle$ only. Now, we discuss how patterns prefixed with page 'a' are mined using the proposed algorithm. As mentioned earlier, we divide the search space with sequences prefixed with 'a' and generate candidate patterns twu and $atwu$ and corresponding effective patterns recursively as shown in Table 2.

Table 2. Candidate and Effective Patterns Prefixed with Page ‘a’

Prefix	Patterns with <i>twu</i>	Patterns with <i>atwu</i>	Pruned Patterns	Candidate Patterns	Effective Patterns
A	<ab:16> <ac:37>	<ab:8> <ac:18.5>	-	<ab> <ac>	<ac>
Ab	<abc:16>	<abc:5.3>	<abc>	-	
Ac	<ace:25>	<ace:8.3>	-	<ace>	<ace>
Ac	<acf:12>	<acf:4>	<acf>	-	
ace	<aced:25>	<aced:6.3>	-	<aced>	<aced>
aced	<acedf:12>	<acedf:2.4>	<acedf>	-	

Patterns satisfying the criteria $atwu \geq \text{min-avg-utility}$ would become candidate patterns and in this case, there are four candidate patterns generated as given in Table 2. While generating candidate patterns at level k , a pattern not satisfying the given criterion is pruned and it will not be participating in generating its super pattern at level $k+1$. For example, when a pattern <abc> is found to be a pattern not satisfying the given criteria, then it is pruned and would not become a candidate pattern to generate its super patterns. Once all the candidate patterns are generated, their actual and average utilities are computed by scanning the database once. The actual utility of a pattern is computed by summing up its utilities when the pattern appears as a subset of sequences in the database. For example, the actual utility of pattern <ace> is the sum of utilities in sequences S_2 and S_5 in which it exists and thus the actual utility is $8(S_2) + 11(S_5) = 19$. By dividing the actual utility-sum (19) with number of pages in pattern <ace>, we get the actual average utility as 6.3. Similarly, we compute the actual and average utilities of other candidate patterns <ab> <ac> and <ced> as <5, 22, 23> and <2.5, 11, 7.6>. Patterns satisfying the given criterion would be effective web traversal patterns and shown in Table 2.

5. Experimental Results

To study and evaluate the performance of our proposed algorithm and compared with an existing approached proposed by C.F. Ahmed et al.,2009. We have used two real datasets *CTI* and *kosarak* obtained from repositories available in <http://www.cs.depaul.edu> and UCI Machine Learning repository. Each data set consists of a collection of sessions where each session has a sequence of page references. However, both datasets do not provide the utility value of each web click. Similar to the previous utility-based pattern mining algorithms[8,9], we have generated random numbers for the utility values ranging from 1 to 10 measured in terms of seconds for web clicks or pages. *CTI* Dataset contains the preprocessed and filtered sessionized data for the main DePaul CTI Web server (<http://www.cs.depaul.edu>). Here, the file used for our experiment is the *cti.tra* file which contains the filtered sessionized data in transaction format. Each line in this file corresponds to the sequence of pages visited during one session. While the order of occurrence of pageview in each session represents the order in which these pageviews were visited, the transactions do not contain repeated visited to the same pageview in the same session. Thus, only the first access to a pageview is recorded as part of the transaction. To make the mining process easier, some preprocessing is performed on *cti.tra* file to translate pageviews into page ids using *cti.cod* file. The *kosarak* dataset contains web click-stream data of a Hungarian on-line news portal. It is a huge sparse dataset containing almost one million transactions (990,002) and 41,270 distinct items. Our programs were written in Microsoft Visual Studio (2005) C# and run with the Windows XP operating system on an Intel dual core 2.18 GHz CPU with 2 GB main memory.

5.1 Execution Time

We varied the minimum average utility and tested the performance of the proposed algorithm with the above datasets. Unlike previous mining algorithms, the number of candidate patterns generated in the proposed algorithm is greatly reduced due to computation of transaction weighted utility of the patterns in the projected sequences rather considering the entire transaction. From the experimental results given in Fig 1 and 3, we

understand that the number of candidate patterns is greatly reduced in the proposed algorithm compared to the existing approach. It is also observed that the number of resultant patterns gets decreased by increasing the minimum average utility and as an effect the execution time of the algorithm also decreases. Therefore, the number of resultant patterns and the runtime are reduced, when min-avg-utility is increased. Here, we are able to find much difference between the running time of the existing and proposed algorithms of the *kosarak* dataset with lengthy sequences and small difference in the case of *CTI* dataset with short sequences. As a result, the execution time shown in the graphs Fig 2 and Fig 4 exhibit that execution time of our proposed algorithm is better than the existing approach, with both datasets.

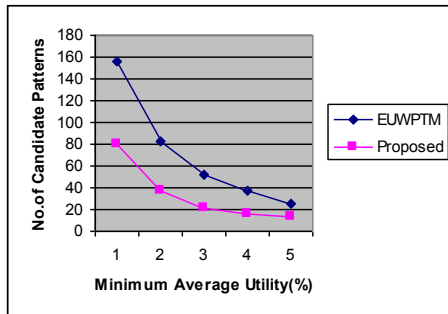


Fig 1. Candidate Pattern Generation on CTI Dataset

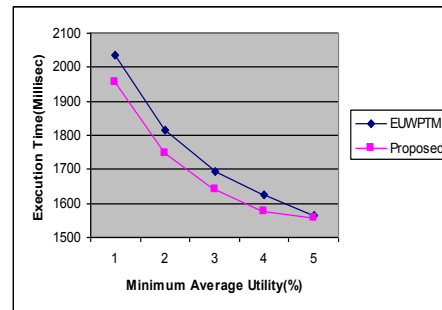


Fig 2. Performance Analysis on CTI Dataset

5.2 Scalability

We studied the scalability of our proposed algorithm on execution time and number of candidate patterns generated by varying the number of transactions in the dataset. We used real *kosarak* dataset for the scalability experiment, since it is a huge sparse dataset with a large number of distinct items and transactions as mentioned earlier. We divided the dataset into five portions from 0.05 to 0.09 million transactions in each part size. That is, the datsize in terms of transactions is varied from 50k to 90k and with min-avg-utility=1%. From the results, it is observed that as the database size increases, overall mining time and number of candidate patterns increase. However, there is a great reduction in the number of candidate patterns in the proposed algorithm, when compared to the existing algorithm. Hence, there is an improvement in the efficiency of the proposed algorithm and outperforms the existing algorithm. Fig 3 and Fig 4 shows the results by varying min-avg-utility on *kosarak* dataset with datsize=100KB.

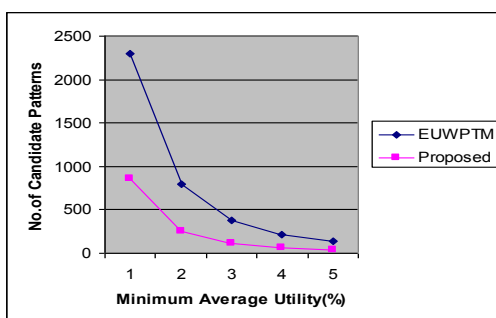


Fig 3. Candidate Pattern Generation on Kosarak Dataset with varying min-avg-utility

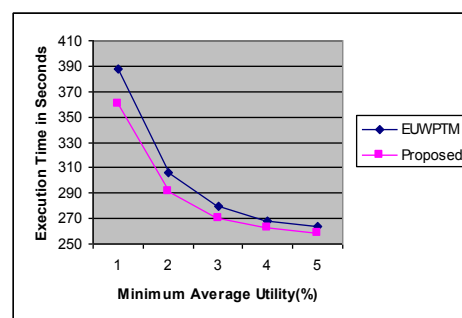


Fig 4. Performance Analysis on Kosarak Dataset with varying min-avg-utility

6. Conclusion

In this paper, the development of an efficient method to mine utility-based effective web path traversal patterns has been discussed. The algorithm is efficient and faster, since it searches for utility patterns in the reduced search space with projected sequences. Moreover, in the proposed algorithm, transaction weighted utility of a pattern is computed using the projected sequences without considering the utilities of items in the entire sequence. This leads to reduction in candidate pattern generation and improves the efficiency of the proposed algorithm. To

reveal better results and resolve the problem occurring due to pattern length, the algorithm mines high average utility patterns rather patterns with actual utility. Here, really good utility patterns out of high utility web traversal patterns are found to determine the impact of user interestingness on patterns better. As a result, the proposed algorithm efficiently finds utility-based effective web traversal patterns by greatly minimizing the number of candidate patterns generated during the mining process. Experiments results show that the proposed algorithm is more efficient than an existing approach. As a future work, sequences with both forward and backward references could be dealt with and impact or significance of web pages may be assigned with external utilities.

References

- B. Mobasher, N. Jain, E.H. Han, and J. Srivastava. "Web mining: Pattern discovery from World Wide Web transactions," Tech Rep: TR96-050, pp. 1-25.
- Chowdhury Farhan Ahmed, Syed Khairuzzaman Tanbeer, Byeong-Soo Jeong, and Young-Koo Lee "Efficient Mining of Utility-Based Web Path Traversal Patterns", ISBN 978-89-5519-139-4 ,Feb. 15-18, 2009 ICACT 2009.
- C.F.Ahmed, S. K. Tanbeer, B. S. Jeong, "A Framework for Mining High Utility Web Access Sequences IETE Technical Review, 2011 Vol: 28 Issue: 1 Pages/record No.: 3-16.
- H. Yao, H. J. Hamilton, and C. J. Butz. "A Foundational Approach to Mining Itemset Utilities from Databases", in: Proceedings of the Third SIAM International Conference on Data Mining (SDM), pp. 482-6, 2004.
- H. Yao, and H. J. Hamilton. "Mining itemset utilities from transaction databases", *Data and Knowledge Engineering*, vol. 59, pp. 603-26, 2006.
- J. Srivastava, R. Cooley, M. Deshpande, and P.-N. Tan, "Web usage mining: discovery and applications of usage patterns from web data", *SIGKDD Explorations*, vol. 1(2), pp. 12-23, 2000.
- J. Pei, J. Han, B. Mortazavi-Asl, and H. Zhu. "Mining access patterns efficiently from web logs," in: Proceedings of the 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), pp. 396-407, 2000.
- L. Zhou, Y. Liu, J. Wang, and Y. Shi. "Utility-based Web Path Traversal Pattern Mining", in: Proceedings of the 7th IEEE International conference on Data Mining Workshops, pp. 373-8, 2007.
- M.S. Chen, J.S. Park, and P.S. Yu. "Efficient data mining for path traversal patterns", *IEEE Transactions on Knowledge and Data Engineering*, pp. 209-21, 1998.
- Pei J., Han J., Mortazavi-Asl B., Wang J., Pinto H., Chen Q., Dayal U. and Hsu M. C. (2004) Mining Sequential Patterns by Pattern-Growth: The PrefixSpan Approach. *IEEE TKDE*, vol. 16, 1424-1440.
- R. Agrawal, and R. Srikant, "Mining sequential patterns," in: Proceedings of the 11th International Conference on Data Engineering, pp. 3-14, 1995.
- Tzung-Pei Hong, Cho-Han Lee, Shyue-Liang Wang, "Mining High Average-Utility Itemsets", Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics, San Antonio, TX, USA - October 2009.
- Y. Liu, W.-K. Liao, and A. Choudhary, "A fast high utility itemsets mining algorithm", *Proc. of the 1st International Conference on Utility-Based Data Mining*, pp. 90-99, 2005.
- Y. Liu, W.K. Liao, and A. Choudhary. "A Two Phase algorithm for fast discovery of High Utility of Itemsets," in: Proceedings of the 9th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), pp. 689-95, 2005.
- Y.S. Lee, and S.J. Yen. "Incremental and interactive mining of web traversal patterns," *Information Sciences*, vol. 178, no.2, pp. 287-306, 2008.