INVESTIGATING SIGNIFICANT CHANGES IN USERS' INTEREST ON WEB TRAVERSAL PATTERNS

Thilagu M[1] and Nadarajan R[2]

[1]Department of Applied Mathematics and Computational Sciences,
PSG College of Technology, Coimbatore, India
[1]mthilagu@gmail.com, [2]nadarajan_psg@yahoo.co.in

ABSTRACT

In recent times, web usage mining has become a promising solution for discovering interesting patterns from web data. The discovered knowledge is useful for web developers or designers to understand users' navigation behaviour and help in improving website design, performance and e-business strategies. Still, analyzing the dynamic behaviour of users' interest on web traversal patterns from the underlying web usage data is an important task in order to analyze the significant changes in users' access to web pages and time spent on them. In this paper, an investigation on users' access behaviour of web traversal patterns in terms of support and utility is done and discussed. Algorithms have been proposed to discover significant web traversal patterns based on support and utility constraints in Phase-I. And, the proposed algorithms apply transitional pattern mining models to detect significant milestones or time points at which users' interest on the discovered patterns increases or decreases dramatically, in Phase-II. The information obtained from the proposed algorithms helps in understanding users' dynamic preferences of web traversal patterns. Experimental studies are done on a real dataset and the results are discussed.

KEYWORDS

Web Usage Mining, Significant Web Traversal Pattern, Significant Milestone, Frequent Pattern, High Utility Pattern

1. INTRODUCTION

In the past few years, there was an increasing interest and a growing research work in web usage mining [1][2] that captures and models web users' behavioural patterns. The interesting patterns obtained from the web usage data could be directly used to efficiently deal with activities relating to e-business, e-services and so on. Moreover, the discovered information is useful to improve the design and structure of a web site to facilitate business organizations function better [3]. Because of this, web usage mining has become a business intelligence solution for most of the organizations. Web usage mining systems like WebPersonalizer[4], Web Usage Miner(WUM), Mining Internet Data for Associative Sequences(Midas)[5], Web Site Information Filter System (WebSIFT)[6]and SiteHelper[7] have been developed based on data mining techniques such as association rule mining, sequential pattern mining clustering, classification etc and discussed in the literature.

Most web traversal pattern algorithms proposed are based on measure support and find frequently occurring patterns, considering whether a web page is present in a traversal path or not. However, frequency of a web traversal pattern does not provide sufficient information on users' impact of the pattern, since all the pages are treated with same significance. To address this limitation in traditional path traversal pattern mining, utility-based web path traversal pattern mining algorithms have been developed. Here, utility is measured in terms of browsing time in seconds

DOI: 10.5121/ijci.2013.2405

representing user preference or interest on web pages in a website. The discovered patterns with utility reveal the significance of them better when compared to measure support. However, in reality, users' interests or preferences are dynamic and varying from time to time. As a result, there may be a dramatic increase or decrease in users' interest at any time point, while accessing a website. Moreover, changes in users' access to web pages reflect in their occurrence frequency and utility and they cannot be the same at all time points in the database. Thus, it is essential for web developers to investigate the dynamic preferences or demands of users, so that the clicking rate of their users could be increased.

With this idea, the research problem considered in this paper has been proposed and the problem is discussed with an illustration as follows. Consider a sample database with twelve transactions at different timestamps, as given in Table 1. Each transaction in the database represents a user web browsing sequence associated with timestamp. The quantity after each webpage represents the browsing time spent by the user. For example, transaction a(1),b(1),c(2),d(2) is having 1 time unit for web page 'a', 1 time unit for 'b' and so on. The column Tran-Utility (Transaction Utility) represents total utility value of each transaction at some time point.

Table 1. A Sample Database

Tid	Web Traversal Sequence	Time Stamp	Tran-Utility
1	a(1), b(1), c(2), d(2)	Jan 2012	6
2	a(1), b(2), c(2), d(3), f (3)	Feb 2012	11
3	a(1), b(1), c(2), d (3)	Mar 2012	7
4	a(2), b(2), c(2), d(3), e(3)	Apr 2012	12
5	b(2), c(4), d(4), e(1), f(1)	May 2012	12
6	c(4), d (2), f(2), g(1)	Jun 2012	9
7	b(1), d(3), e(2), f(2)	Jul 2012	8
8	c(2), e(1), g(1)	Aug 2012	4
9	a(4), b(4), e(2), f(1), g(1)	Sep 2012	12
10	a(6), b(3), e(2), f(1)	Oct 2012	12
11	a(1), b(2), e(2) ,f(2)	Nov 2012	7
12	a(2), b(3), e(3)	Dec 2012	8

As mentioned earlier, the significance of web traversal patterns could be determined either with measure support or utility. Assume that *min-sup* ξ= 25% and *min-util* δ =25%. Here, the minimum support is measured as the percentage of the database size and it is 3(25% of 12) and the minimum utility is measured as the percentage of the total transaction utility in the database and it is 27(25% of 108). The support of a pattern X is defined as the number of occurrences of the pattern in the database and the utility of a pattern X is defined as the sum of utilities of pages that belong to X in all transactions in which X exists in the database. Considering web traversal patterns <abcd> and <abef> in Table 1, their support percentages are computed as 33.33% (4/12*100) and 25% (3/12*100). And, then their utilities in percentage are found as 30.55% (33/108*100) and 27.78% (30/108*100). These patterns are identified as frequent and high utility patterns satisfying the user given thresholds. Moreover, if they are found to be with the same support and utility percentage, then they will be treated equally with the same significance in the existing approaches.

However, users' access to these patterns and the browsing time spent on them are not the same in the entire span time of the database. For instance, it is found that pattern <abcd> appears only in the first four months from Jan 2012 to Apr 2012 (T_1-T_4), while the pattern <abef> appears only in the months from Sep 2012 to Nov 2012 (T_9-T_{11}) in the database. And, their browsing timings are also varying from time to time. Thus, interesting differences between the patterns <abcd> and <abef> could be found by analyzing the surfing behaviour of users considering transactions along with timestamp information. The above said issue is restricted in the existing frequent and utility based pattern mining algorithms, since they discover patterns considering either occurrence or utility of a web page and disregard timestamp information in a transaction. To resolve this problem, algorithms need to be developed to detect transitional patterns in a transaction database. Transitional patterns are the patterns whose users' preferences change over time and they reflect the dynamic behaviour of users' interest better. Transitional patterns may be of either positive or negative, wherein positive patterns have dramatic increase in support or utility and negative patterns have dramatic decrease in support or utility at some points. In this work, algorithms have been designed based on transitional pattern mining model with respect to support and utility measures. Actually, the proposed algorithms first discover frequent and utility patterns and then they identify transitional patterns among them. The algorithms also detect significant milestones, which are the time points with most significant changes in support and utilities of the transitional patterns.

The rest of the paper is organized as follows: Section 2 discusses related work of the problem considered in this work. In Section 3, terms and definitions of the frequent, utility and transitional pattern mining models are presented. In Section 4, the working principle of the proposed algorithms is described. The results of experimental studies are reported in Section 5. The last section gives the summary on the proposed work.

2. RELATED WORK

2.1. Frequent Pattern Mining

A number of web access pattern mining algorithms have been discussed in the literature, to find interesting patterns from web data. Chen et al. (1996) [8] developed two algorithms that determine large reference sequences for web traversal paths and these maximal sequences with forward references are mined based on Apriori approach[9]. The first algorithm full-scan (FS) finds frequently occurring web traversal patterns with multiple database scans. To reduce database scans and the disk I/O cost involved, a second algorithm called selective-scan (SS) or improved full-scan algorithm has been developed. To avoid the level-wise candidate generation-and-test in the existing algorithms, web access pattern mine (WAP-mine) algorithm with a tree structure called WAP-tree proposed by Pei et al. (2000)[10]. To improve WAP-mine algorithm, a position coded pre-ordered linked WAP-tree approach was developed to perform efficient mining of web access sequences [11]. Y.S. Lee, and S.J. Yen, (2008) designed IncWTP and WssWTP algorithms for incremental and interactive mining of web traversal patterns [12]. A variant algorithm based on Prefix-Span has been proposed to discover frequent subsequences in web logs by generating maximal potential patterns in order to minimize the number of scans [13]. All the above algorithms consider only the presence or absence of a web page in a transaction.

2.2. Utility Pattern Mining

Mining of high utility patterns considers non-binary values called utilities of web pages in a sequence, rather binary values (0/1). Zhou et al. [14] introduced the concept of utility in web path traversal mining model to express the significance of web pages in terms of browsing time spent by the user. This algorithm adopts the definitions in Two-Phase [15][16] high utility mining algorithm and mines patterns based on Apriori [9]. However, it suffers from the level-wise candidate generation-and-test methodology. To overcome this problem, an algorithm called EUWPTM (Efficient Utility-based Web Path Traversal Mining) [17] based on pattern growth approach [18] was proposed. The above two utility based algorithms consider only forward references of web pages and their internal utilities in a web sequence. A framework for mining high utility web access sequences algorithm [19] discovers 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 utility-based web access sequence tree (UWAS-tree) and incremental UWAS-tree (IUWAS-tree) for mining web access sequences in static and incremental databases respectively. An improved algorithm of EUWPTM has been proposed considering projected transaction utility of prefix based patterns to reduce number of unnecessary candidate patterns and pattern length while computing actual utility to generate effective patterns [20].

2.3. Transitional Pattern Mining

Transitional pattern mining frameworks are applied in different domains in the literature [21][22][23][24][25]. However, the proposed work is based on mining of transitional patterns with measure support in a transaction database. Thus, algorithms that address the significance of frequency change of patterns at different time points in a transaction database are discussed here. Algorithms proposed by Qian Wan and Aijun [26][27] are refereed as Transitional Pattern-Mine (TP-Mine) algorithms. These algorithms introduce a new type of pattern called transitional pattern that reflects the dynamic behaviour of frequent pattern in a transaction database. Discovery of transitional patterns and their significant milestones for the given period range are discussed in these algorithms. The major difference between old TP-Mine [26] and new TP-mine [27] algorithms is that the former algorithm considers time points in which a pattern does not occur, where as the latter algorithm considers time points in which a pattern occurs, while computing two supports of a pattern X namely support$^{i}_{-}$(X) and support$^{i}_{+}$(X) and the transitional ratio at ith time point of the pattern X. Thus, the latter algorithm is faster and efficient than former one. TP-Mine algorithms contain two major phases as follows. During Phase-I, frequent patterns with their supports are derived in the transaction database, using the efficient and scalable frequent pattern mining algorithm FP-growth [28]. In Phase-II, transitional patterns and their significant milestones are found based on these frequent patterns. An improved TP-mine algorithm was introduced by B. Kiran Kumar and A. Bhaskar [29] to minimize the number of computations, by finding frequent patterns for the given range of period of milestones in a database during Phase-I and identify transitional patterns from frequent patterns satisfying the user given transitional pattern threshold. Here, the existing approaches discussed above consider transitions or changes only in occurrence frequency of non-sequential patterns in transaction databases, whereas the proposed work considers significant changes both in frequency and utilities of web traversal patterns, which are also sequential patterns.

3. Problem Definition

Let P = {$p_1, \dots, p_m$} be a set of '*m*' web pages. A set X = {$p_j, ..., p_k$} $\subseteq$ P, where $j \leq k$ and $j, k \in$ [1,*m*], is called a web traversal or access pattern. A web access sequential database WASD over P is a set of sequence or transactions T ={T_1, ..., T_n}, n = |WASD|, where |WASD| is the total number of transactions in WASD or the size of WASD. Each page p_i is associated with utility information in a transaction. A web traversal sequence T_i = (tid, Y) is a transaction where tid represents a transaction-id (or timestamp), $i \in$ [1, *n*] and Y is a pattern. If X$\subseteq$ Y, then it is said that X is contained in T_i or X occurs in $T_{i.}$ Here, sequences are with only forward references of pages and gap between pages is not allowed.

3.1 Frequent Pattern Mining Model

Definition 1: The support of a pattern X denoted as *sup*(X) and is defined as the number of transactions in which X occurs in the database WASD. The support is measured in terms of ratio as *sup*(X)/|WASD|, where |WASD| is the number of transactions in WASD.

Definition 2: If *sup*(X) is not less than the user given minimum support threshold ξ (*min-sup*), then X is said to be frequent web traversal pattern, where *min-sup* is given by the percentage of number of transactions of the database.

3.2 Utility Based Pattern Mining Model

The terms and definitions of the traditional utility-based mining model are first defined as given in previous approaches [14][17].

Definition 3: The quantitative measure of utility for page P_i in transaction T_j, denoted as $u(P_i, T_j)$ is defined as

$$u(P_i,T_j)=iu(P_i ,T_j)\times eu(P_i) \qquad (1)$$

where $iu(P_i,T_j)$ represents internal utility of page P_i in transaction T_j and $eu(P_i)$ represents external utility or the unit profit value of page P_i in a website.

Definition 4: The utility of a pattern X in transaction T_i, denoted as $u(X, T_i)$ is defined as

$$u(X, T_i) = \sum_{Pi \in X} u(P_i, T_i) \qquad (2)$$

where X= {$P_1,P_2,\ldots P_k$} is a pattern with length *k*, X$\subseteq$ T_i and $1\leq k \leq m$.

Definition 5: The utility of a pattern X denoted as *u*(X) is defined as

$$u(X) = \sum_{Ti \in WASD} \sum_{Pi \in X} u(P_i, T_i) \qquad (3)$$

Definition 6: The transaction utility of a transaction T_i denoted as $tu(T_i)$ is defined as

$$tu(T_i) = \sum_{Pi \in Ti} u(P_i, T_i) \qquad (4)$$

A major problem identified in utility based pattern mining model is that it does not support downward closure property straight like frequent pattern mining. For example, a super sequence <a(1),b(2),c(2)> with utility 5 has sub sequences <a>, <b> and <c> with low utilities 1,2 and 2. Assuming that *min-util*= 4, the super sequence is a high utility pattern, but its sub sequences are low utility patterns. To overcome this problem, transaction weighted utilization concept introduced in [15] is adopted in existing utility based pattern mining algorithms. The transaction weighted utility of a pattern X is the maximum possible value or upper bound, so that any super-sequence of pattern X cannot be a high transaction weighted utility sequence if X is not a high transaction weighted utility sequence satisfying the given *min-util*.

Definition 7: The transaction weighted utility of a pattern X in WASD is the sum of the transaction utilities of all transactions containing X and it is defined as

$$twu(\mathrm{X},\mathrm{WASD}) = \sum_{\mathrm{X}\subseteq T_i \in WASD} tu(\mathrm{T_i}) \qquad (5)$$

Definition 8: The minimum utility threshold δ (*min-util*), is given by the percentage of the total transaction utility value of the database WASD and it is defined as

$$min\text{-}util = \delta \text{ x } \sum_{\mathrm{T_i} \in WASD} tu(\mathrm{T_i}) \qquad (6)$$

Definition 9: If *u*(X) is not less than the user given minimum utility threshold δ (*min-util*), then X is said to be high utility web traversal pattern, where *min-util* is given by the percentage of number of total transaction utility of the database.

3.3. Transitional Pattern Mining Model

The terms and definitions of transitional pattern mining model based on support are adopted from [27], however transitional pattern mining model based on utility is newly introduced in this proposed work.

Assuming that the transactions in database WASD are ordered by their time-stamps and timestamp of ith transaction is greater than to that of transactions occurring before it in the database WASD. The position of a transaction T in the database WASD is denoted as *p*(T).The ith transaction of pattern X is denoted as $\mathrm{T^i(X)}$. The proposed research problem addresses transitional pattern mining model with respect to support and utility measures as follows.

3.3.1. Transitional Frequent Pattern Mining Model

Definition 10: The ith milestone of pattern X in the database WASD is represented as the relative position of ith transaction in the database. The ith milestone of pattern X is given as $m^{\mathrm{i}}(\mathrm{X})$ and defined as

$$m^{\mathrm{i}}(\mathrm{X}) = \frac{p(\mathrm{T^i(X)})}{|\,\mathrm{WASD}\,|\,\mathrm{x}100} \qquad (7)$$

where $\mathrm{T^i(X)}$ is the ith transaction in which pattern X occurs.

Definition 11: The support of frequent pattern X before its ith milestone denoted as $sup^{\mathrm{i}}_{-}(\mathrm{X})$ is defined as

$$sup^{\mathrm{i}}_{-}(\mathrm{X}) = \frac{i}{p(\mathrm{T}^{\mathrm{i}}(\mathrm{X}))} \tag{8}$$

Definition 12: The support of frequent pattern X after its i^th milestone denoted as $sup^{\mathrm{i}}_{+}(\mathrm{X})$ and is defined as

$$sup^{\mathrm{i}}_{+}(\mathrm{X}) = \frac{sup(\mathrm{X}) - i}{|\,\mathrm{WASD}\,| - p(\mathrm{T}^{\mathrm{i}}(\mathrm{X}))} \tag{9}$$

3.3.1.1. Transitional Support Ratio

A measure called transitional support ratio of pattern X is defined to determine the difference between support before and after at i^th milestone of a transitional frequent pattern.

Definition 13: The transitional support ratio (*tsr*) of frequent pattern X at its i^th milestone in the database WASD is denoted as $tsr^{\mathrm{i}}(\mathrm{X})$ and defined as

$$tsr^{\mathrm{i}}(\mathrm{X}) = \frac{sup^{i}_{+}(\mathrm{X}) - sup^{\mathrm{i}}_{-}(\mathrm{X})}{\mathrm{Max}(sup^{i}_{+}(\mathrm{X}), sup^{i}_{-}(\mathrm{X}))} \tag{10}$$

3.3.1.2. Transitional Frequent Pattern

A transitional frequent pattern with respect to measure support is defined with constraints as follows.

Definition 14: A frequent pattern X is transitional frequent pattern (TFP) in WASD if there exists at least one milestone of X, say $m^{\mathrm{k}}(\mathrm{X}) \in \mathrm{M}^{\mathrm{X}}$ satisfying the following constraints, where $k \in [1,n]$ and M^{X} is the set of milestones of pattern X and $n=|\mathrm{M}^{\mathrm{X}}|$.

a. $sup^{\mathrm{k}}_{-}(\mathrm{X}) \geq$ *min-sup* and $sup^{\mathrm{k}}_{+}(\mathrm{X}) \geq$ *min-sup*
b. $tsr^{\mathrm{k}}(\mathrm{X})| \geq \sigma$ [σ is the transitional support threshold]

3.3.2. Transitional Utility Pattern Mining Model

Definition 15: The utility of high utility pattern X before its i^th milestone denoted as $u^{\mathrm{i}}_{-}(\mathrm{X})$ is defined as

$$u^{\mathrm{i}}_{-}(\mathrm{X}) = \frac{\sum_{j=1}^{k} u(\mathrm{X}, \mathrm{T_j})}{\sum_{j=1}^{k} tu(\mathrm{T_j})} \tag{11}$$

where '*k*' is the relative position of i^th transaction representing i^th milestone of high utility pattern X in the database.

Definition 16: The utility of high utility pattern X after its i^th milestone denoted as $u^{\mathrm{i}}_{+}(\mathrm{X})$ and is defined as

$$u^{\mathrm{i}}_{+}(\mathrm{X}) = \frac{\sum_{j=k+1}^{n} u(\mathrm{X},\mathrm{T_j})}{\sum_{j=1}^{n} tu(\mathrm{T_j}) - \sum_{j=1}^{k} tu(\mathrm{T_j})} \qquad (12)$$

where '*k*' is the relative position of ith transaction representing ith milestone of high utility pattern X in the database and '*n*' is the number be of transactions in the database.

3.3.2.1. Transitional Utility Ratio

A measure called transitional utility ratio of pattern X is defined to determine the difference between utility before and after at ith milestone of a transitional utility pattern.

Definition 17: The transitional utility ratio (*tur*) of high utility pattern X at its ith milestone in the database D is denoted as $tur^{\mathrm{i}}(\mathrm{X})$ and defined as

$$tur^{\mathrm{i}}(\mathrm{X}) = \frac{u^{\mathrm{i}}_{+}(\mathrm{X}) - u^{\mathrm{i}}_{-}(\mathrm{X})}{\mathrm{Max}(u^{\mathrm{i}}_{+}(\mathrm{X}), u^{\mathrm{i}}_{-}(\mathrm{X}))} \qquad (13)$$

3.3.2.2. Transitional Utility Pattern

Definition 18: A high utility pattern X is transitional utility pattern (TUP) in WASD if there exists at least one milestone of X, say $m^{\mathrm{k}}(\mathrm{X}) \in \mathrm{M}^{\mathrm{X}}$, satisfying the following constraints, where k $\in [1,n]$ and M^{X} is the set of milestones of pattern X.

a. $u^{\mathrm{k}}_{-}(\mathrm{X}) \geq$ *min-util* and $u^{\mathrm{k}}_{+}(\mathrm{X}) \geq$ *min-util*
b. $tur^{\mathrm{k}}(\mathrm{X})| \geq \mu$ [μ is the transitional utility threshold]

3.3.3. Transitional Support and Utility Thresholds

Transitional support and utility thresholds are used as parameters in constraint (b) of Definitions 14 and 18 to identify transitional frequent and utility patterns. That is, transitional support and utility ratio values $tsr^{\mathrm{i}}(\mathrm{X})$ and $tur^{\mathrm{i}}(\mathrm{X})$ are tested with these thresholds. Here, both of the ratios would lie in the range between -1 and 1 according to Definitions 13 and 17. However, considering the absolute transitional support or utility ratio of a pattern at its ith milestone, the higher absolute value tells that there is a greater difference in support or utility before and after its ith milestone and vice versa. Here, transitional support and utility thresholds should be absolute values greater than 0 and less than 1. That is, the value zero (0) indicates that there is no difference between before and after support or utilities at ith milestone and the value one (1) indicates that there is no after support or utility at ith milestone occurring as the last milestone of a pattern X. Hence, both values (zero and one) can be disregarded while setting up transitional support and utility thresholds. Moreover, the threshold values are set to be up with values greater than 0.5 in order to find milestones with significant differences between their before and after support or utility. To achieve this, condition (a) in Definitions 14 and 18 is used in finding transitional patterns whose support or utility changes dramatically before and after its milestones in a transaction database.

3.3.4. Positive and Negative Transitional Patterns

As mentioned earlier, a transitional pattern may be of either positive or negative. That is, a transitional pattern may reflect changes as either dramatic increase or decrease in its support or utility.

Definition 19: A frequent or high utility transitional pattern X is called positive transitional pattern (PTP) when $tsr^{k}(X) > 0$ or $tur^{k}(X) > 0$.

Definition 20: A frequent or high utility transitional pattern X is called a negative transitional pattern (NTP) when $tsr^{k}(X) < 0$ or $tur^{k}(X) < 0$, where '*k*' denotes the k^{th} milestone of pattern X.

A frequent or high utility pattern X may be both a positive and negative transitional pattern in the same transaction database if there exists two milestones say m^1 and m^2 satisfying the above mentioned conditions, where $tsr^1(X) > 0$ or $tur^1(X) > 0$ and $tsr^2(X) < 0$ or $tur^2(X) < 0$.

3.3.5. Significant Milestones

Generally, a transitional pattern may satisfy conditions (a) and (b) mentioned in Definition 14 or 18 with respect to measure support or utility, at multiple milestones. However, milestones where the support or utility of a transitional pattern changes most significantly are the interesting ones. These milestones are called as significant milestones, which may be either support/utility-incrementing or support/utility-decrementing milestones.

Definition 21: The significant incrementing milestone of a positive transitional pattern X with respect to set M^X is defined as a tuple, $(m^p(X),tsr^p(X))$ or $(m^p(X),tur^p(X))$, where $m^p(X) \in M^X$ is the p^{th} milestone of X such that $\forall\, m^i(X) \in M^X$, $tsr^p(X) \geq tsr^i(X)$ or $tur^p(X) \geq tur^i(X)$.

Definition 22: The significant decrementing milestone of a negative transitional pattern X with respect to set M^X is defined as a tuple, $(m^q(X),tur^q(X))$ or $(m^q(X),tur^q(X))$, where $m^q(X) \in M^X$ is the q^{th} milestone of X such that $\forall\, m^i(X) \in M^X$, $tsr^q(X) \leq tsr^i(X)$ or $tur^q(X) \leq tur^i(X)$.

4. PROPOSED METHODOLOGY

In this section, the two algorithms proposed in this work are described as follows. The first algorithm called TFPM (Transitional Frequent Pattern Mining) is based on frequent and transitional frequent pattern mining models, where as the second algorithm called TUPM (Transitional Utility Pattern Mining) is based on utility and transitional utility pattern mining models. These two algorithms perform their tasks in two phases. In Phase I, frequent pattern mining model or utility-based mining model is applied to detect frequent or high utility patterns in the database. During Phase II, support or utility based transitional pattern mining model is used to detect transitional frequent and utility patterns and their significant milestones.

4.1. Phase I: Mining of Frequent and High Utility Web Traversal Patterns

To mine frequent or high utility patterns in the database, the proposed algorithms apply divide-and-conquer technique introduced in PrefixSpan algorithm [18], a fast and efficient sequential pattern mining technique. The algorithm projects the database recursively and searches for patterns in the minimized search space.

Firstly, the algorithm TFPM initially scans the database once to compute support of each 1-sequence. For each frequent 1-sequence α, the database is projected prefixed with α. It then generates patterns prefixed with α by performing the following steps recursively. At first, it finds all frequent 1-sequence β in the α-projected database and appends them with α to generate patterns at the next level. The projected database is further divided and projected with the newly generated pattern αβ. At each level, a pattern with support not satisfying the given min-sup is pruned from generating its super patterns. The above steps are repeated for each frequent 1-sequence in order to generate a complete set of frequent patterns in the database.

The algorithm TUPM mines high utility patterns similar to algorithm TFPM. In this case, the significance of a web access pattern is determined with measure utility instead of support. As discussed earlier, the problem with utility pattern mining model is that high utility patterns may consist of low utility patterns and downward closure property cannot be directly supported in this model. To overcome this problem, transaction weighted utilization concept is applied in the proposed algorithm like existing utility based pattern mining approaches. Hence, transaction weighted utility of a pattern is considered rather its actual utility while generating candidate patterns. Because of this, the database is scanned once again to compute the actual utilities of the high transaction weighted utility candidate patterns. Finally, patterns with actual utility satisfying the given *min-util* are returned as high utility patterns.

4.2. Phase II: Detecting Transitional Patterns and Their Significant Milestones

Once frequent or high utility patterns are discovered during Phase-I, the next step is to detect transitional patterns and their significant milestones in the database. As mentioned earlier, the input for Phase-II would be frequent or high utility patterns with their transaction-ids or time points. To find the milestones of the discovered frequent or high utility patterns, the database is scanned once.

With frequent patterns and their transaction-ids as input, the algorithm TFPM applies transitional frequent pattern mining model to detect support-based transitional patterns and their significant milestones. That is, frequent patterns satisfying constraints in Definition 14 are identified as transitional frequent patterns or support-based transitional patterns. These patterns are classified into either positive or negative transitional patterns based on transitional ratios at each of their milestones. Moreover, milestones or time points with maximum transitional ratio of positive transitional patterns become significant incrementing milestones and time points with maximum transitional ratio of negative transitional patterns become significant decrementing milestones, using Definitions in Section 3.3.5.

Similarly, the algorithm TUPM applies transitional utility pattern mining model to detect transitional utility patterns and their significant milestones. That is, high utility patterns satisfying constraints in Definition 18 are identified as transitional utility patterns or utility-based transitional patterns. Also, positive and negative transitional utility patterns are found with their significant milestones, according to Definitions in Section 3.3.5.

4.3. Algorithm Steps

The steps of the TFPM algorithm are given below.

```
Step 1: Scan the database once and find support of each 1-sequence
Step 2: Find the frequent 1-sequence satisfying the user given min-sup
Step 3: For each frequent 1-sequence $\alpha$
        Repeat
          a. Divide the search space prefixed with $\alpha$
          b. Find the frequent 1-sequence $\beta$ that could be merged with $\alpha$
          c. Output $\alpha\beta$ as frequent pattern
Step 4: Scan the database once and find the transaction-ids of all the discovered
        frequent patterns
Step 5: For each frequent pattern X
          a. For each i^th milestone of X
               a1. Read the database and compute $sup^{i}_{-}(X)$, $sup^{i}_{+}(X)$ and
                   $tsr^{i}(X)$
               a2. If both $sup^{i}_{-}(X)$ and $sup^{i}_{+}(X)$ are not less than min-sup
                     If $tsr^{i}(X)$ is not less than transitional support threshold $\sigma$
                       Output X is a transitional frequent pattern
                     If $tsr^{i}(X) > 0$
                       Output X is a positive transitional pattern
                     If $tsr^{i}(X) < 0$
                       Output X is a negative transitional pattern
          b. For each transitional frequent pattern X
               Output milestones with most changes in support as
               Significant milestones of X
```

The steps of the TUPM algorithm are given below.

```
Step 1: Scan the database once and find transaction weighted utility (twu) of each 1-sequence
Step 2: Find the high twu1-sequence satisfying the user given min-util
Step 3: For each high twu 1-sequence $\alpha$
            Repeat
              a. Divide the search space prefixed with $\alpha$
              b. Find the high twu 1-sequence $\beta$ that could be merged with $\alpha$
              c. Output $\alpha\beta$ as high twu pattern
Step 4: Scan the database once and find the actual utilities of all high twu patterns
Step 5: Identify patterns with actual utility satisfying the user given min-util as high
        utility patterns
Step 6: Scan the database once and find the transaction-ids of all the discovered
        utility patterns
Step 7: For each high utility pattern X
        For each $i^{th}$ milestone of X
                a1. Read the database and compute $u^i_-(X)$ and $u^i_+(X)$ and $tur^i(X)$
                a2. If both $u^i_-(X)$ and $u^i_+(X)$ are not less than min-util
                        If $tur^i(X)$ is not less than transitional utility threshold $\mu$
                            Output X is a transitional utility pattern
                        If $tur^i(X) > 0$
                            Output X is a positive transitional pattern
                        If $tur^i(X) < 0$
                            Output X is a negative transitional pattern
            a. For each transitional utility pattern X
                   Output milestones with most changes in utility as
                   Significant milestones of X
```

5. Experimental Studies

To test and analyze the TFPM and TUPM algorithms of the proposed work, experimental studies are pursued on a real dataset. The proposed algorithms are written in Visual C# 2008 and tested using an IBM machine with dual core processor with 2 GB RAM and Windows XP Operating System.

5.1. Dataset

The real dataset BMS-WebView-1 is obtained from KDDCUP [30][31]. It contains 59,602 transactions with 497 distinct pages or items, with the average transaction size as 2.5. They represent several months' worth of click-stream data or user traversing web pages of an e-commerce web site. Here, a transaction is considered as a web access sequence with only forward page references and however pages are not provided with utility values. The utility values of pages in BMS-WebView-1 dataset are assigned with generated random numbers ranging from 1 to 10 and measured in terms of seconds, like existing utility-based pattern mining algorithms [14][15].

5.2. Results

In this work, extensive experiments have been conducted on the BMS-WebView-1 dataset, by setting up minimum support or utility threshold with low values and transitional support or utility

threshold with value higher than or equal to 0.5 to find transitional patterns with great difference between before and after support or utility at ith milestone, so that significant milestones or time points are detected with dramatic changes in support or utility.

Transitional Pattern (X)	m^i**(X)**	sup^i_-**(X) (%)**	sup^i_+**(X) (%)**	tsr^i**(X)**
10295,10299	11.24%	1.21	0.46	-61.98%
10331,10335	50.12 %	0.20	0.58	65.52 %
10315,10331	45.14 %	0.22	0.51	56.86 %
34885,34889	82.72%	0.23	0.69	66.66%
33449,33469	61.50 %	2.43	0.26	-89.30%

Table 2. Transitional Frequent Patterns and Their Significant Milestones

The investigations on the dynamic behaviour of users' interest on web traversal patterns are obtained by testing the TFPM and TUPM algorithms. The results of TFPM algorithm are first discussed. The algorithm is tested with constraints as minimum support threshold ξ =0.1% and transitional support threshold σ =0.5. Here, the number of frequent patterns generated is 672 and out of which 42 patterns are found to be transitional frequent patterns. In this case, the number of positive and negative transitional patterns is 11, the number of positive transitional patterns is 20 and the number of negative transitional patterns is 11. Table 2 shows a subset of transitional frequent patterns with length 2 and their significant milestones ranging from 11.24% to 82.72%.

The first pattern <10295,10299> is a negative transitional pattern, since its significant milestone with transitional support ratio is (11.24%,-61.98%). Generally, the significant milestones occurring at the beginning or end of the database can be ignored, since the number of transactions before and after of these milestones may not be uniformly distributed in the database. Because of this, dramatic changes in support before and after cannot be derived accurately. Due to the above said reason, the last pattern <34885,34889> can also be ignored. To avoid this, an appropriate period can be set up to find interesting transitional patterns with significant milestones having dramatic changes in support. However, it is understood that that the occurrence of the pattern <10295,10299> is more at the beginning of the database, whereas the occurrence of pattern <34885,34889> is more at the end of the database, using their significant milestones and corresponding transitional ratios such as (11.24%,-61.98%) and (82.72%,66.66%) respectively.

Considering the positive transitional patterns <10331,10335> and <10315,10331>, it is found that pattern <10331,10335> has significant change in users' interest at milestone 50.12 %, since its after support (0.58) is more than twice of before support (0.20) and the difference is represented in terms of transitional ratio as 65.52%. Similarly, pattern <10315,10331> is also having significant change in users' interest at milestone 45.14%, since its after support (0.51) is more than twice of before support (0.22) and it is represented as 56.86%. Moreover, pattern <33449,33469> is a negative transitional pattern with significant decrease in users' interest at milestone 61.50 % with high transitional support ratio -89.30%. Hence, the reason for dramatic increase and decrease in users' interestingness of these patterns are to be examined.

The algorithm TUPM is tested for mining utility patterns and their significant milestones, with constraints as minimum utility threshold δ=0.1% and transitional utility threshold μ=0.5. The number of high utility patterns generated is 748 and the number of transitional patterns found out of high utility patterns is only 29. In this case, the number of positive and negative transitional patterns is 4, the number of positive transitional patterns is 10 and the number of negative transitional patterns is 15. Here also, only a subset of transitional utility patterns with length 2 and their significant milestones ranging from 10.95% to 88.65% shown in Table 3 are considered for discussion purpose.

Transitional Pattern (X)	$\boldsymbol{m^{i}(X)}$	$\boldsymbol{u^{i}_{-}(X)}$ **(%)**	$\boldsymbol{\underline{u}^{i}_{+}(X)}$ **(%)**	$\boldsymbol{tur^{i}(X)}$
10295,10299	11.24%	0.94	0.38	-59.57%
10295,10311	10.95%	0.90	0.37	-58.89%
10331,10335	54.57%	0.19	0.48	60.42 %
12751,18863	88.65%	0.14	0.80	82.50%
10295,10307	70.76%	0.99	0.49	-50.50%

Table 3. Transitional Utility Patterns and Their Significant Milestones

Considering patterns with uniform distribution of transactions before and after their significant milestones, <10331,10335> is identified as a positive transitional pattern with significant milestone, transitional utility ratio as (54.57%,60.42%). Another pattern <10295,10307> is a negative transitional pattern with significant milestone, transitional utility ratio as (70.76%, -50.50%) in Table 3. Both of these patterns are to be given with more attention, since differences between their milestones' before and after utility are high. As a result, users' interest on frequent or high utility web access patterns may have dramatic increase at any time point in the database, due to dynamic nature of users' interest. Thus, the results obtained from the proposed algorithms would help web developers in decision making, so that they could take necessary actions to improve their e-services.

5.3. Performance Analysis

To test the performance of the proposed algorithms on BMS-WebView-1 dataset, minimum support and utility thresholds are varied from 0.1 to 0.5 in percentage. Here, the performances of algorithms TFPM and TUPM are measured in terms of execution time. The number of frequent or high utility patterns generated by varying threshold *min-sup* in TFPM or *min-util* in TUPM is plotted in Figure 1. Figure 2 exhibits the relationship between the execution timings and varying threshold *min-sup* in TFPM or *min-util* in TUPM. The execution timings of TUPM is more than that of TFPM even though the number of high utility patterns generated is less that of frequent patterns, since TUPM requires an additional database scan to compute the actual utility of the high *twu* utility candidate patterns during utility pattern mining process. From Figure 1 and Figure 2, it is observed that the number of resultant patterns gets decreased by increasing the minimum support or utility and as an effect the execution time of the algorithm also decreases. The scalability of the proposed algorithms is tested by varying the number of transactions in the BMS-WebView-1 dataset. Here, the time complexity of the proposed algorithms is linear, which implies that if the data size increases, the execution time of the algorithms also gets increased.

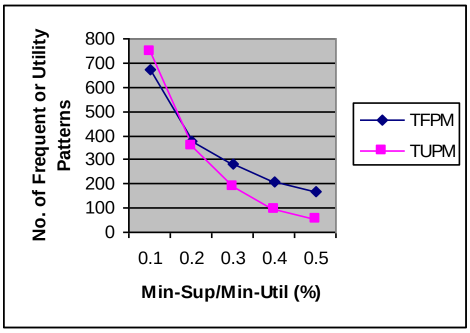

Figure 1. Number of Frequent or Utility Patterns varying *min-sup* or *min-util*

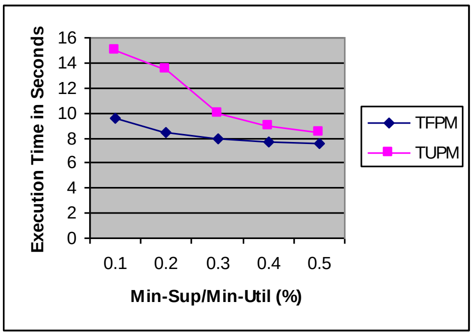

Figure 2. Performance Analysis varying *min-sup* or *min-util*

6. CONCLUSIONS

In this paper, a method to investigate the significant changes in users' interest on web traversal patterns is discussed. The discovery of transitional web traversal patterns whose user preferences change over time is restricted in the existing approaches. To resolve this problem, the proposed work detects those patterns to capture the dynamic behaviour of users' interest or preferences in terms of support and utility. Experimental results on a real dataset of an e-commerce website show that mining of positive and negative transitional patterns based on occurrence frequency or browsing time is useful task for web developers in understanding the dynamic preferences of users better. Moreover, information on transitional web traversal patterns helps in improving website design in order to provide better services to the users.

REFERENCES

[1] J. Srivastava, R. Cooley, M. Deshpande, and P.-N. Tan (2000), "Web usage mining: discovery and applications of usage patterns from web data", *SIGKDD Explorations*, Vol. 1(2), pp. 12-23.

[2] B. Mobasher, N. Jain, E.H. Han, and J. Srivastava. (1996), "Web mining: Pattern discovery from World Wide Web transactions," *Tech Rep*: TR96-050, pp.1-25.

[3] Ajith Abraham (2003), "Business Intelligence from Web Usage Mining", *Journal of Information & Knowledge Management (JIKM)*, World Scientific Publishing Co., Singapore, Volume 2, No. 4, pp.375-390.

[4] B. Mobasher, R. Colley and J. Srivastava (2000), Automatic personalization based on web usage mining, *Commun. ACM* 43(8), pp.142–151.

[5] M. Spiliopoulou, Web usage mining for web site evaluation (2000), *Commun. ACM* 43(8) pp. 127–134.

[6] J. Srivastava, R. Cooley, M. Deshpande and P. Tan (2000), Web usage mining: Discovery and applications of usage patterns from web data, *SIGKDD Explor.* 1(2), pp. 12–23.

[7] D. Pierrakos, G. Paliouras, C. Papatheodorou and C. Spyropoulos (2003), Web usage mining as a tool for personalization: A survey, *User Model. User-Adapt. Interact.* 13(4) pp. 311–372.

[8] M.S. Chen, J.S. Park, and P.S. Yu. (1998), "Efficient data mining for path traversal patterns", *IEEE Transactions on Knowledge and Data Engineering*, pp. 209-21.

[9] R. Agrawal, and R. Srikant (1995), "Mining sequential patterns," in: Proceedings of the 11 th International Conference on Data Engineering, pp.3-14.

[10] J. Pei, J. Han, B. Mortazavi-Asl, and H. Zhu. (2000),"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.

[11] C. I. Ezeife, and Y. Lu (2003), "Position coded pre-order linked WAP-tree for web log sequential pattern mining" PAKDD'03 Proceedings of the *7th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining.* pp. 337-349.

[12] Y.S. Lee, and S.J. Yen. (2008), "Incremental and interactive mining of web traversal patterns," *Information Sciences*, Vol. 178(2), pp. 287- 306.

[13] M. Thilagu and R. Nadarajan (2010), "Discovery of Maximal Contiguous Sequence Patterns with Priority in Web Logs", *Asian Journal of Information Technology*, 9: pp 238-242.

[14] L. Zhou, Y. Liu, J. Wang, and Y. Shi.(2007), "Utility-based Web Path Traversal Pattern Mining", in: Proceedings of the *7th IEEE International conference on Data Mining Workshops*, pp. 373-8.

[15] Y. Liu, W.-K. Liao, and A. Choudhary (2005), "A fast high utility itemsets mining algorithm", *Proc. of the 1st International Conference on Utility-Based Data Mining*, pp. 90–99.

[16] Y. Liu, W.K. Liao, and A. Choudhary. (2005), "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.

[17] Chowdhury Farhan Ahmed, Syed Khairuzzaman Tanbeer, Byeong-Soo Jeong, and Young-Koo Lee (2009), "Efficient Mining of Utility-Based Web Path Traversal Patterns", ISBN 978-89-5519-139-4 ,Feb. 15-18, ICACT 2009, pp.2215-2218.

[18] 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, pp.1424-1440.

[19] C. F. Ahmed, S. K. Tanbeer, B. S. Jeong (2011), "A Framework for Mining High Utility Web Access Sequences, *IETE Technical Review*,Vol. 28(1) pp.3-16.

[20] M. Thilagu, R. Nadarajan (2012), "Efficiently Mining of Effective Web Traversal Patterns With Average Utility", *International Conference on Communication, Computing, and Security,ICCCS-2012*, *Procedia Technology* Vol.6, pp.444 – 451.

[21] Guo-Zhu Dong and Jin-Yan Li. (1999), "Efficient mining of emerging patterns: discovering trends and differences. In *KDD '99:Proc. of the fifth ACM SIGKDD Int. Conf. on Knowledge discovery and data mining*, pp.43–52.

[22] Stephen D. Bay and Michael J. Pazzani (1999) , "Detecting change in categorical data: mining contrast sets" In *KDD '99: Proc of the fifth ACM SIGKDD Int. Conf. on Knowledge discovery and data mining*, pp. 302–306.

[23] Ying-Jiu Li, Peng Ning, X. Sean Wang, and Sushil Jajodia (2001), “Discovering calendar-based temporal association rules”, In *TIME ’01: Proc. of the 8th Int. Symposium on Temporal Representation and Reasoning (TIME’01)*, pp.111, Washington,DC, USA, IEEE Computer Society.

[24] Marko Salmenkivi and Heikki Mannila (2005), “Using markov chain monte carlo and dynamic programming for event sequence data”, *Knowledge and Information Systems*, 7(3), pp.267–288.

[25] Charu C. Aggarwal (2003), “A framework for diagnosing changes in evolving data streams”, In *SIGMOD ’03: Proc. of the 2003 ACM SIGMOD Int. Conf. on Management of data*, pp.575–586.

[26] Qian Wan and Aijun An. (2007),“Transitional patterns and their significant milestones”, In *Proc. of the 7th IEEE Int. Conf. on Data Mining*, Omaha, NE, USA.

[27] Qian Wan and Aijun (2009),“Discovering Transitional Patterns and Their Significant Milestones in Transaction Databases”, *IEEE Trans. on Knowledge and Data Engg.*,vol.21,No.12 pp.1692-1707.

[28] Jia-Wei Han, Jian Pei, Yi-Wen Yin, and Run-Ying Mao (2004), “Mining frequent patterns without candidate generation: A frequentpattern tree approach”, *Data Mining and Knowledge Discovery*, 8(1), pp. 53–87.

[29] B. Kiran Kumar and A. Bhaskar (2010), “ETP-Mine: An Efficient Method for Mining Transitional Patterns” *International Journal of Database Management Systems*(IJDMS) Vol.2, No.3, pp. 21-29.

[30] Frequent itemset mining dataset repository. Available at:http://www.fimi.cs.helsinki.fi/data/

[31] Z. Zheng, R. Kohavi, and L. Mason (2001), "Real world performance of association rule algorithms," *Proc. of the 7th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pp. 401-6.

Authors

M.Thilagu received the MCA (Master of Computer Applications) in the year 1993, from PSG College of Technology, Coimbatore and M.Phil in Computer Science in the year 2003, from Bharathiar Univerisity,Coimbatore. Currently, she is pursuing PhD in Computer Science in the area of web log mining. Her research interests include Data Mining, Information Retrieval and Object Oriented Computing.

Dr.R.Nadarajan,Msc,,PhD., has 25 years of teaching experience and guided 8 PhDs. Currently he is supervising 20 research scholars in the fields of Computer Science. He published more than 54 research papers in refereed international journals. He is the member of ACM, ISTE, Computer Society of India, and OR Society of India. He is the architect of the Five year integrated M.Sc (Software Engineering) and Five year integrated M.Sc (Theoretical Computer Science) course. His research interests include Database Management System, Data Mining and Object Oriented Computing.