WebPUM: A Web-based recommendation system to predict user future movements

Norwati Mustapha

Expert Systems With Applications

Expert Systems with Applications 37 (2010) 6201–6212

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

WebPUM: A Web-based recommendation system to predict user future movements

Mehrdad Jalali [a,b,*], Norwati Mustapha [b], Md. Nasir Sulaiman [b], Ali Mamat [b]

[a] Department of Software Engineering, Faculty of Engineering, Islamic Azad University of Mashhad Mashhad, Iran
[b] Department of Computer Science, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Selangor, Malaysia

ARTICLE INFO

Keywords:
Web usage mining
Web-based recommendation systems
Navigation pattern mining

ABSTRACT

Web usage mining has become the subject of exhaustive research, as its potential for Web-based personalized services, prediction of user near future intentions, adaptive Web sites, and customer profiling are recognized. Recently, a variety of recommendation systems to predict user future movements through Web usage mining have been proposed. However, the quality of recommendations in the current systems to predict user future requests in a particular Web site is below satisfaction. To effectively provide online prediction, we have developed a recommendation system called WebPUM, an online prediction using Web usage mining system and propose a novel approach for classifying user navigation patterns to predict users' future intentions. The approach is based on the new graph partitioning algorithm to model user navigation patterns for the navigation patterns mining phase. Furthermore, longest common subsequence algorithm is used for classifying current user activities to predict user next movement. The proposed system has been tested on CTI and MSNBC datasets. The results show an improvement in the quality of recommendations. Furthermore, experiments on scalability prove that the size of dataset and the number of the users in dataset do not significantly contribute to the percentage of accuracy.

1. Introduction

Given the rapid growth rate of the Web, proliferation of e-commerce, Web services, and Web-based information systems, the volumes of click streams and user data collected by Web-based organizations in their daily operations have reached huge proportions. Substantial increase in the number of Web sites presents a challenging task for Webmasters to organize the content of the Web sites to cater user needs. Modeling and analyzing Web navigation behavior are helpful in understanding the kind of information in demand by online users.

The analyzed results from Web navigation behaviors are indispensable knowledge to intelligent online applications and Web-based personalization system in improving searching accuracy during information seeking. Nevertheless, as online navigation behaviors revolutionize in each passing day, intelligent information extraction is an equally challenging issue.

Web usage mining (WUM) refers to automatic discovery and pattern analysis in click streams and associated data collected or generated that from user interactions with Web resources on one or more Web sites (Cooley, Mobasher, & Srivastava, 1997; Srivastava, Cooley, Deshpande, & Tan, 2000; Wang & NetLibrary, 2006). Web usage mining has been used effectively as an approach for automatic personalization and as a way to overcome deficiencies in traditional approaches such as collaborative filtering. The goal of personalization based on Web usage mining is to recommend a set of objects to the active user, possibly consisting of links, ads, text, or products, tailored to user perceived preferences. This task is accomplished by matching the active user session with usage patterns discovered through Web usage mining.

In this paper, to effectively provide online prediction, we have developed a Web-based recommendation system known as WebPUM for online prediction through Web usage mining system. We also proposed a novel approach for classifying user navigation patterns to predict user future intentions. The approach is based on graph partitioning to model user navigation patterns during the mining phase. Next, this approach is utilized with the longest common subsequence algorithm in classifying current user activities to predict user next movement.

The remainder of this paper is organized as follows: Section 2 covers the existing Web usage mining (WUM) recommendation systems and investigates other recommendation systems that utilize WUM for prediction of user next request. Section 3 describes the system design while Section 4 focuses on system evaluation. Results of experimental evaluations are reported in Section 5. Finally, Section 6 concludes the current study and sheds light on some directions in the future works.

* Corresponding author. Address: Department of Software Engineering, Faculty of Engineering, Islamic Azad University of Mashhad, Mashhad, Iran.
E-mail addresses: mehrdadjalali@ieee.org, mehrjalali@gmail.com (M. Jalali), norwati@fsktm.upm.edu.my (N. Mustapha), nasir@fsktm.upm.edu.my (Md. Nasir Sulaiman), ali@fsktm.upm.edu.my (A. Mamat).

0957-4174/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2010.02.105

2. Background and related works

Huan and Kamber (2000) propose Web mining, a new unifying are for all methods to apply data mining to Web data. However, Web mining tools aim to extract knowledge from the Web, rather than retrieving information. Research on Web mining is classified into three categories, which are Web structure mining that identifies authoritative Web pages, Web content mining that classifies Web documents automatically or constructs a multilayered Web information base, and Web usage mining that discover user access patterns in navigating Web pages (Mobasher, Cooley, & Srivastava, 1999). From the data-source perspective, both Web structure and Web content mining target the Web content, while Web usage mining targets the Web access logs.

Web usage mining (WUM) comprises of three major processes: data pretreatment, data mining, and pattern analysis (Cooley, Mobasher, & Srivastava, 1999). Pretreatment performs a series of processing on Web log files, which are data conversion, data cleaning, user identification, session identification, path completion, and transaction identification. Next, mining algorithms are applied to extract user navigation patterns. A navigation pattern represents the relationships among Web pages in a particular Web site. Some pattern analyzing algorithm is applied to extract data from data mining part for the recommendation system. Recently, a number of Web usage mining (WUM) systems have been proposed to predict user's preferences and their navigation behaviors.

Analog (Yan, Jacobsen, Garcia-Molina, & Dayal, 1996) is one of the first WUM systems. It is structured according to off-line and online component. The off-line component builds session clusters by analyzing past user activities recorded in server log files. The online component builds active user sessions, which are then classified according to the generated model. The classification allows to identify pages related to the ones in the active session and to return the requested page with a list of suggestions. The geometrical approach used for clustering is affected by several limitations, related to scalability and to the effectiveness of the results found. Nevertheless, the architectural solution introduced was maintained in several other more recent projects.

Mobasher, Cooley, and Srivastava (2000) and Nakagawa and Mobasher (2003) present WebPersonalizer, a system that provides dynamic recommendations as a list of hypertext links to users. The analysis is based on anonymous data usage combined with the structure formed by the site hyperlinks. Data mining techniques (i.e., clustering, association rules, and sequential pattern discovery) are used in the preprocessing phase in order to obtain aggregate usage profiles. In this phase, Web server logs are converted into clusters that are made up from sequences of visited pages, and set of pages with common usage characteristics. The online phase considers the active user session in order to find matches among the user activities and the discovered usage profiles. Matching entries are then used to compute a set of recommendations that will be inserted into the last requested page as a list of hypertext links. WebPersonalizer is a good example of two-tier architecture for personalization systems.

A partitioning graph-theoretic approach is proposed to develop adaptive Web sites that are able to automatically improve organization and presentation of the personalization systems by mining usage logs (Perkowitz & Etzioni, 2000b). The core element of this system is a new clustering method, called cluster mining, which is implemented in the PageGather algorithm. Clusters are defined either in terms of cliques, or connected components, which are proven to be more coherent. While connected clusters components are larger, they are faster to compute and easier to find. A new index page is created from each cluster with hyperlinks to all the pages in the cluster.

PageGather receives user sessions, represented as sets of pages that have been visited, as input. Using these data, the algorithm creates a graph, as signing pages to nodes. An edge is added between two nodes if the corresponding pages co-occur in more than a certain number of sessions. The main advantage of PageGather is that it can create overlapping clusters. Furthermore, in contrast to the other clustering methods, the clusters generated by this method are able to directly group together all characteristic features of the users. This means each cluster represents a unique behavioral pattern, associating pages in a particular Web site. However, being a graph-based algorithm, it is rather computationally expensive, especially in the case where cliques are computed.

Another partitioning–clustering method is employed in WebCANVAS, which visualizes user navigation paths in each cluster (Cadez, Heckerman, Meek, Smyth, & White, 2000). In this system, user sessions are represented by categories of general topics for Web pages. A number of predefined categories are used as a bias and URLs from the Web server log files are assigned to them, constructing the user sessions.

In the context of Web usage mining, the discovery of association rules usually aims at finding associations between Web pages and their co-occurrence in user sessions (Mobasher et al., 1999). This is particularly interesting for personalization, as the ultimate objective is to use itemsets to dynamically recommend Web pages to the users. One interesting conclusion of this work is that, although itemsets can be used directly as input to the recommendation engine in order to provide dynamic Web pages, it is not a very accurate method. To improve this, the itemsets are clustered, using the *k*-means algorithm in order to produce transaction clusters. A transaction cluster represents a group of users with similar browsing behaviors. Nonetheless, transaction clusters were known to be inappropriate for managing data with a large number of dimensions, i.e., Web pages recorded in Web log files. Therefore, Web pages were also grouped into usage clusters according to their frequency of co-occurrence in user transactions.

In Yoda, the recommendation system is designed to support large-scale Web-based applications that require highly accurate recommendations in real-time (Shahabi, Banaei-Kashani, Chen, & McLeod, 2001). Yoda employed a hybrid approach that combines collaborative filtering (CF) and content-based querying to achieve higher accuracy. Yoda is structured as a tunable model that is trained online with and offers real-time online recommendations. The online process benefits from an optimized aggregation function with low complexity that allows real-time weighted aggregation of the soft classification to predefined recommendation sets.

Sequential Web Access-based Recommender System (SWARS) uses sequential access pattern mining (Zhou, Hui, & Chang, 2004). In the off-line phases of the proposed system, CS-Mine is an efficient sequential pattern mining algorithm used to identify sequential Web access patterns with high frequencies. The access patterns are then stored in a compact tree structure, called Pattern-tree, which is then used for matching and generating Web links for recommendations during online phases. In this system, when the number of recommended pages is more than 5, then the precision and satisfaction are not significant.

Liu and Kešelj (2007) propose an automatic classification of Web user navigation patterns and a novel approach to classifying user navigation patterns and predicting user future requests. The approach is based on the combined mining of Web server logs and the content of the retrieved Web pages. Character *N*-grams is used to represent the content of Web pages. A collection of *N*-grams, combined with user navigation patterns represents the user navigation profiles. In this system, current off-line mining system is being integrated into an online Web recommendation system to observe and to calculate the degree of user satisfaction on the

generated recommendations derived from the predicted requests by the system.

Baraglia and Palmerini proposed a WUM system called SUGGEST that provides useful information to aid in Web navigation and to optimize the Web server performance (Baraglia & Silvestri, 2004, 2007). SUGGEST adopts a two-level architecture composed by an off-line creation of historical knowledge and an online engine that understands user behaviors. As the requests arrive at the system, it will incrementally update a graph representation of the Web site based on the active user sessions. Next, the active sessions are classified using a graph partitioning algorithm.

However, there are potential limitations of this architecture. First the memory required to store Web server pages is quadratic in the number of pages, which will severely affect large Web sites that are made up from millions of pages. Second, it does not permit us to manage Web sites made up from dynamically generated pages.

In summary, all of these works attempt to find reference architecture and algorithm to improve quality of the personalized recommendation systems, but alas the recommendations are still below satisfaction. In our work we advance a model and propose novel approach to predict user intention in the near future request.

3. System design

This paper proposes a new architecture called WebPUM, which is to predict user future requests. The model is being partitioned into two interleaved phases; off-line and online. In spite of being separated in the model, the off-line phase strongly affects the online phase. Fig. 1 illustrates the architecture of WebPUM.

3.1. Offline phase of WebPUM

This phase consists two main modules, which are data pretreatment and navigation pattern mining. Data pretreatment module is designed to extract user navigation sessions from the original Web user log files. A new clustering algorithm based on graph partitioning is introduced for navigation patterns mining.

3.1.1. Data pretreatment

Generally, data pretreatment in Web usage mining systems aims to reformat the original Web user log files to identify all user sessions. The Web server registers all user navigations as part of the Web user log files. There are many types of Web logs depending on parameter settings in each server, but typically the log files share the same basic information such as: client IP address, request time, requested URL, HTTP status code, or referrer.

Moreover, several pretreatment tasks need to be done before applying navigation pattern mining algorithms on the Web user log files. The tasks include data collection, data cleaning, data structuration, and data summarization

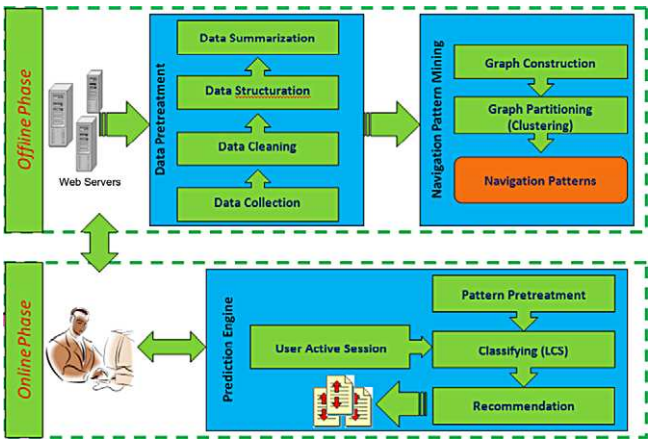

Fig. 1. Architecture of WebPUM.

- Data collection.
  At the beginning of data pretreatment, Web server log files are collected from several Web servers to be utilized in various tasks.
- Data cleaning.
  The data in the original Web user log files are raw; hence, not all the log entries are valid for Web usage mining. The system should only keep the entries that carry relevant information. Therefore, data cleaning is performed to eliminate the irrelevant entries from the log files, which includes Web robots, spiders and crawlers, picture files associated with requests for particular pages, CGI files, and other irrelevant files.
- Data structuration.
  After cleaning the Web user log files, a series of tasks will be applied on the cleaned dataset to identify users and sessions. The behavior of an individual user in a particular Web session is recognized in the current module during the off-line phase. Generally, this module groups a log file of unstructured requests into user and user session through the tasks of user identification and session identification.
  (i) User identification.
  In any Web usage mining system, a mechanism to distinguish different users is required for analyzing user access behavior (Berendt, Mobasher, Spiliopoulou, & Wiltshire, 2001; Spiliopoulou, Mobasher, Berendt, & Nakagawa, 2003). Since users are treated as anonymous in most Web servers, two heuristic strategies have been proposed to help on discriminating among of users, which are the proactive strategy and the reactive strategy (Spiliopoulou et al., 2003).The proactive strategy tries to unambiguously associate each request to a site visitor before or during the visitor's interaction with the site, while the reactive strategy attempts to associate requests with the visitor after the interaction with the site based on the existing incomplete records. As a rule, a cookie-based identifier is a must for applications of proactive strategy (Spiliopoulou et al., 2003). However, the use of cookie needs to comply with existing laws, to the least that users are clearly aware of its presence. Because of this, the proactive strategy is not always a feasible option.The reactive strategy tries to approximate users in terms of IP address, type of operating system, and browsing software. In other words, requests are treated as they are from the same user and are collected into the same group under a particular user only if these requests possess the same IP address, operating system and browsing software. WebPUM applies the reactive strategy for user differentiation.
  (ii) Session identification.
  A session can be described as the group of activities performed by a user from the moment he/she entered the site to the moment he/she left. Therefore, session identification is the process of segmenting the access log of each user into individual access sessions (Cooley et al., 1999). Two time-oriented heuristic methods have been proposed for session identification, which are session-duration-based method and page-stay-time-based method (Berendt et al., 2001; Cooley et al., 1999; Spiliopoulou et al., 2003).
  The session-duration-based method aims to set a session duration threshold. If the duration of a session exceeds a certain limit, it could be considered that there is another access session of the user. Discovered from empirical findings, a 30-min threshold for total session duration has

been recommended (Berendt et al., 2001; Spiliopoulou et al., 2003). For the page-stay-time-based method, the time spent on a page must not exceed the threshold. If the difference between the request of most recently assigned to a session and the next request from the user is greater than the threshold, it is assumed that a new access session has started. A conservative threshold for 10 min page-stay time has been proposed to capture loading time and studying the content of a page (Berendt et al., 2001; Spiliopoulou et al., 2003). In this research, current techniques and algorithms will be applied to recognize user session.

- Data summarization.
  After structurizing the data, data file is transferred to a relational database. It then applies data transformation and aggregated data computation for user sessions.
  (i) Data transformation.
  Data transformation transforms the set of URLs into tables in the relational database. In this process, the numeric values are assigned to URL addresses in the URL field. This is to reduce complexity of the experimental process before data mining techniques are applied. Relational database will also be used for creating training and testing sets for system evaluation.
  (ii) Aggregated data computation.
  While the previous process in data summarization is to create relational database, the aggregated data computation is to focus on processing new parameters for user session and user navigation. These parameters represent the statistical values that will be used in the proposed method. For instance, for each user session in the database, we are able to compute following parameters:
  (1) Number of visit for individual session.
  (2) Session length in time (the difference between the date of last visit date and the date of first visit).
  (3) Time length of two individual visits in a session.

Also, we are able to compute the following parameters for the whole database:

(1) Number of users and sessions.
(2) Number of repeated Web pages in the sessions.
(3) Percentage of the session's length

3.1.2. Navigation pattern mining

In the proposed system, user navigation patterns are described as the common browsing characteristics among a group of users. Since many users may have common interests at any point during their navigation, navigation patterns should capture the overlapping interests or the information needs of the users. In addition, navigation patterns should also be capable to discriminate Web pages based on the significance in each pattern.

Following the data pretreatment step, navigation pattern mining is performed on the derived user access sessions. The representative user navigation pattern can be obtained by clustering algorithms. Clustering of user navigation pattern aims to group sessions into clusters based on their common properties. Access sessions that are obtained by the clustering process are actual patterns of Web user activities. These patterns will be used to further classify current user activities in online phase of the WebPUM. In this study, user navigation patterns are defined as follows:

Definition 1. A user navigation pattern *np* captures an aggregate view of the behavior of a group of users based on their common interests or information needs. As the results of session clustering, $NP = \{np_1, np_2, \ldots, np_k\}$ is used to represent the set of user navigation patterns, in which each np_i is a subset of P, the set of Web pages (Liu & Kešelj, 2007).

In this study, a clustering model is used for navigation pattern mining. The model exploits graph partitioning algorithm by applying new method for creating undirected graph. The clustering model is build to find collection of related pages at a particular Web site, relying on the visit-coherence assumption (Perkowitz & Etzioni, 2000a). The pages that a user visits during one interaction with the site tend to be conceptually related. The process of the clustering takes three steps: are elaborated as follows:

(1) Compute the degree of connectivity between Web pages and create an adjacency matrix.
(2) Create an undirected graph corresponding to the adjacency matrix.
(3) Find connected component in the graph based on graph search algorithm.

Step 1: Compute the degree of connectivity between Web pages and create an adjacency matrix.

For each pair of pages a and b, we compute $W(a, b)$, which is the degree of connectivity between Web pages. A new measurement is proposed for approximating the degree of connectivity for each pair of Web pages in a session, which are *Time Connectivity* and *Frequency*.

Time Connectivity measures the degree of visit ordering for each two pages in a session. Using novel formula (1):

$$TC_{a,b} = \frac{\sum_{i=1}^{N} \frac{T_i}{T_{ab}} \times \frac{f_a(k)}{f_b(k)}}{\sum_{i=1}^{N} \frac{T_i}{T_{ab}}} \quad (1)$$

where T_i is time duration in ith session that contain both pages a and b and T_{ab} is the difference between requested time of page a and page b in the session. We consider $f(k) = k$ if a Web page appears in position k. However, a different form of f could be chosen. For instance, it is also possible to increase the importance of the position in each page during a session by taking $f(k) = k^2$. The formula is also normalized so all values for time connectivity are between 0 and 1.

Frequency measures the occurrence of two pages in each session as shown in (2):

$$FC_{a,b} = \frac{N_{ab}}{\max\{N_a, N_b\}} \quad (2)$$

where N_{ab} is the number of sessions containing both page a and page b. N_a and N_b are the number of session containing only page a and page b. This formula also holds values between 0 and 1.

In WebPUM *Time Connectivity* and *Frequency* are two strong indicators of the degree of connectivity for each pair of Web pages. Therefore, in the weight measure we devised, *Time Connectivity* and *Frequency* are valued equally. We use the harmonic mean of *Time Connectivity* and *Frequency* to represent the connectivity between two pages, shown as (3). We take this formula for weight of each edge in the undirected graph

$$W_{a,b} = \frac{2 \times TC_{ab} \times FC_{ab}}{TC_{ab} + FC_{ab}} \quad (3)$$

Step 2: Create an undirected graph corresponding to the adjacency matrix.

The graph structure can be used to store the weights as an adjacency matrix M where each entry M_{ab} contains the value W_{ab} computed according to formula in (3). To limit the number of edge in such graph, element of M_{ab} whose value is less than the threshold value and is small correlated will be thus discarded. In this study, this threshold is named as *MinFreq*.

Step 3: Find connected component in the graph based on graph search algorithm.

The graph partitioning algorithms divide a graph into k disjoint partitions, such that the partitions are connected and there are a small number of connections between the partitions. Graph partitioning algorithm is utilized to search for groups of strongly correlated Web pages by partitioning the graph according to its connected components. Depth-first search (DFS) is an algorithm for traversing or searching a graph. Starting from a vertex a, DFS induced by M is applied to search for the connected component reachable from this vertex. Once the component has been found, the algorithm checks if there are any nodes that are not considered in the visit. If so, it means that a previously connected component has been split, and therefore, it needs to be identified. To do this, DFS is applied again by starting from one of the nodes that is not yet visited. In the worst case, when all the URLs are in the same cluster, the cost of this algorithm will be linear in terms of the number of edges in the complete graph G.

Two main parameters must be accounted for while the algorithm is applied to the undirected graph. Minimum frequency and minimum cluster size are two parameters that significantly affect mining of navigation patterns. *MinFreq* is a minimum frequency parameter for filtering weights that are below a constant value. The edges of the graph whose values are less than *MinFreq* are inadequately correlated and are thus not considered by the DFS graph search algorithm. DFS also considers all the connected components that posses the number of nodes greater than a fixed size. Otherwise the rest of components will be considered as insignificant. In this paper, the minimum cluster size is termed as *MinClusterSize*.

In this study, connected components that have been created based on graph partitioning algorithm are considered as a set of navigation patterns. At the end of this step, the algorithm shows $NP = \{np_1, np_2, \ldots, np_k\}$, whereby NP is a set of navigation patterns. NP can also be considered as a set of clusters that will further be utilized during the online phase.

Fig. 2 illustrates an example of the clustering process. There are a set of Web pages in each session, and we consider each page as a graph vertex (Fig. 2(a)). An undirected graph is created based on the degree of connectivity between the Web pages (Fig. 2(b)). If we take *MinFreq* = 2 and *MinClusterSize* = 3, the edges lower than this value will be consequently eliminated before graph search algorithm (Fig. 2(c)) is applied. In the last part, DFS algorithm is applied to the undirected graph to find the connected components in the graph (Fig. 2(d)). The clusters $C2$ and $C5$ are eliminated due to *MinClusterSize* being lower than three in these clusters. The result of navigation pattern mining (Clustering) is shown as follows:

$$C1 = NP1 = \langle P_1, P_2, P_5, P_6, P_9, P_{11}\rangle$$
$$C3 = NP3 = \langle P_4, P_8, P_{14}\rangle$$
$$C4 = NP4 = \langle P_{13}, P_{15}\rangle$$

The algorithm for navigation pattern mining (clustering) based on graph partitioning algorithm is shown in Fig. 3.

3.2. Online phase of WebPUM

According to the different phases of the proposed system, the system generates navigation patterns in the off-line phase by utilizing graph partitioning algorithm. An online component within the prediction engine has the task to predict user future requests. Classifying the user current activities based on navigation patterns in a particular Web site is the main objective of this phase. In addition, creating a list of recommended Web pages as prediction of user future movement is another objective in this phase. The main online component is the prediction engine.

Prediction engine is used to classify user navigation patterns and to predict user future requests. For this purpose, we propose the longest common subsequence (LCS) algorithm to classify current user activities. In order to classify user's active session, we look for navigation patterns that contain the larger number of similar Web pages in each session. Pattern search approaches can be utilized to find similar Web pages between current active session and navigation patterns. LCS algorithm is to find the longest subsequences that are common to all sequences in the set of sequences (often just two).

The second objective of this component is computing a recommendation set for the current session, consisting of links to pages that the user may want to visit based on similar usage patterns. The recommendation set essentially represents a "short-term" view of potentially useful links based on the user's navigational activity through the site. These recommended links are then added to the last page in the session accessed by the user before that page is sent to the user browser.

3.2.1. Longest common subsequences (LCS)

The problem of comparing two sequences $\vec{\alpha}$ and $\vec{\beta}$ in determining their similarity is one of the fundamental problems in pattern matching. One of the basic form of the problem is to determine the longest common subsequence (LCS) of $\vec{\alpha}$ and $\vec{\beta}$. The LCS string comparison metric measures the subsequence of maximal length common to both sequences (Apostolico, 1997).

Formally, given a sequence $\vec{\alpha} = \langle \alpha_1, \alpha_2, \ldots, \alpha_n\rangle$, a sequence $\vec{\gamma} = \langle \Gamma_1, \Gamma_2, \ldots, \Gamma_n\rangle$ is a subsequence of $\vec{\alpha}$ if there exists a strictly increasing sequence $\langle j_1, j_2, \ldots, j_n\rangle$ of indices of $\vec{\alpha}$ such that for all $i = 1, 2, \ldots, l$, we have $\alpha_{ji} = \Gamma_i$. Given two sequences of $\vec{\alpha}$ and $\vec{\beta}$, we say that $\vec{\gamma}$ is common subsequence of $\vec{\alpha}$ and $\vec{\beta}$ if $\vec{\gamma}$ is a subsequence of both $\vec{\alpha}$ and $\vec{\beta}$. We are interested in finding the maximum-length or longest common subsequences given the two paths or a sequence of page-visits $\vec{\alpha} = \langle \alpha_1, \alpha_2, \ldots, \alpha_n\rangle$ and $\vec{\beta} = \langle \beta_1, \beta_2, \ldots, \beta_m\rangle$. LCS has a well-studied optimal sub-structure property as given by the following:

Theorem 1. *Let* $\vec{\alpha} = \langle \alpha_1, \alpha_2, \ldots, \alpha_n\rangle$ *and* $\vec{\beta} = \langle \beta_1, \beta_2, \ldots, \beta_m\rangle$ *be sequences, and let* $\vec{\gamma} = \langle \Gamma_1\Gamma_2, \ldots, \Gamma_n\rangle$ *be any LCS of* $\vec{\alpha}$ and $\vec{\beta}$.

(1) *If* $\alpha_n = \beta_m$, *then* $\Gamma_1 = \alpha_n = \beta_m$ *and* $\vec{\gamma}_{l-1}$ *is a LCS of* $\vec{\alpha}_{n-1}$ *and* $\vec{\beta}_{m-1}$.
(2) *If* $\alpha_n \neq \beta_m$, *then* $\Gamma_1 \neq \alpha_n$ *implies* $\vec{\gamma}$ *is a LCS of* $\vec{\alpha}_{n-1}$ *and* $\vec{\beta}$.
(3) If $\alpha_n \neq \beta_m$, *then* $\Gamma_1 \neq \beta_m$ *implies* $\vec{\gamma}$ *is a LCS of* $\vec{\alpha}$ *and* $\vec{\beta}_{m-1}$ *where* $\vec{\alpha}_{n-1} = \langle \alpha_1, \alpha_2, \ldots, \alpha_{n-1}\rangle$, $\vec{\beta}_{m-1} = \langle \beta_1, \beta_2, \ldots, \beta_{m-1}\rangle$ and $\vec{\gamma}_{l-1} = \langle \Gamma_1\Gamma_2, \ldots, \Gamma_{n-1}\rangle$.

The proof of this theorem can be found in Gormen, Leiserson, Rivest, and Stein (1990). Efficient recursive algorithms to compute the LCS exist using this property of the LCS (Dancik, 1994). We reserve the discussion on details of the algorithms in Aho, Hirschberg, and Ullman (1974), Dancik (1994) and Hirschberg (1977).

Definition 2. Let S_1 and S_2 be two sequences. $|LCS(S_1, S_2)|$ is the size of the longest common subsequence between S_1 and S_2. The degree of similarity between S_1 and S_2 is defined as (4):

$$Sim_{LCS} = \frac{2 \times |LCS(S_1, S_2)|}{|S_1| + |S_2|} \tag{4}$$

3.2.2. Recommendation algorithm with LCS

Pattern search algorithm is employed to find navigation patterns based on the current user activities in order to predict and recommend user future request. We applied LCS, a pattern search in the recommendation part of the system. In this paper, there are several steps to create recommendation set based on the current user session in the online phase of the system.

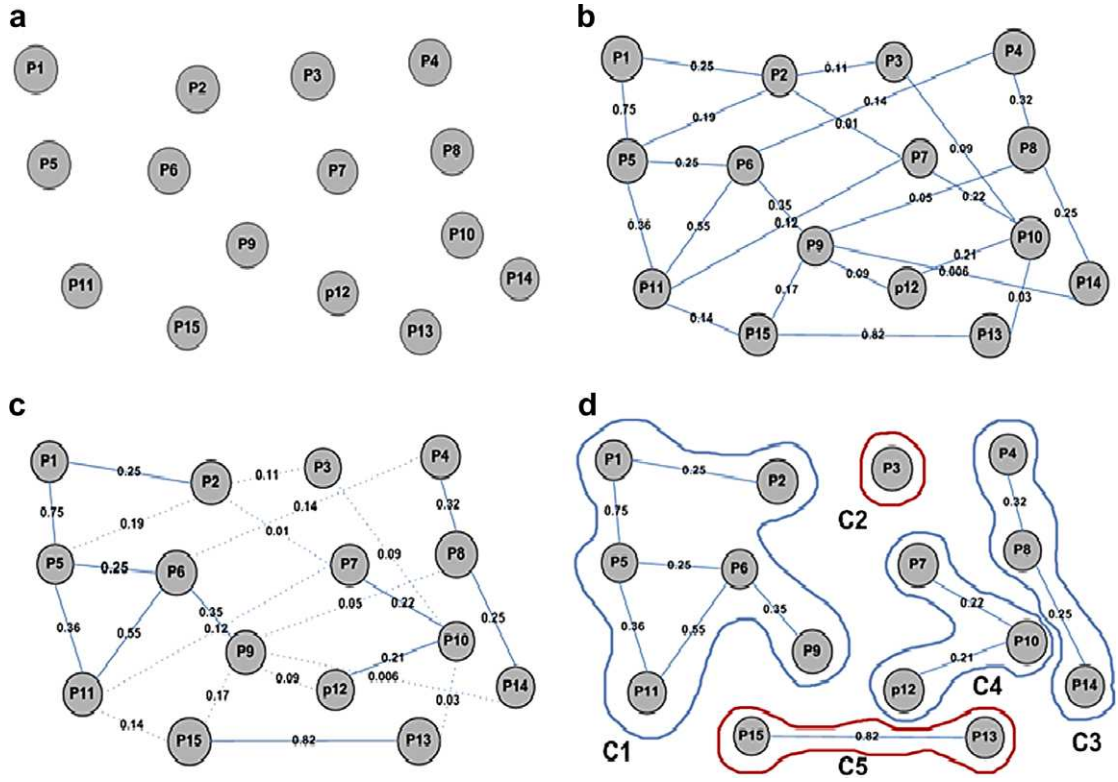

Fig. 2. An example of clustering process.

(1) Data pretreatment for recommendation.
(2) User classifying based on LCS algorithm.
(3) Predict next user's activities and create a recommendation set.

Step 1: Data pretreatment for recommendation.

Preprocessing for both current active session and navigation patterns are performed as the first step of the prediction engine. In this step, data is prepared by applying LCS algorithm to take into account the efficiency of the algorithm. In this study, the current active session S is represented as a vector:

$$\vec{S} = \langle P_1, P_2, \ldots, P_m \rangle$$

where $P_i = n$, $1 \leqslant i \leqslant m$, and n is a unique numeric value that is assigned to each Web page in the off-line phase. When user visits a Web page, the system replaces the Web page with a predefined unique numeric value.

A fixed size sliding window is utilized over current active session to capture the ongoing user activities. The mean of Web pages in each session of dataset is considered as one user active session window. Consequently, in the proposed algorithm, Web pages inside the user's active session window will be arranged in order according to the numeric values.

In this research each cluster that has been created in the off-line phase is regarded as set of navigation patterns, $n\vec{p} = \langle n\vec{p}_1, n\vec{p}_2, \ldots, n\vec{p}_m \rangle$ where $n\vec{p}_i$ is a set of k Web pages as a navigation pattern. This is shown as $n\vec{p}_i = \langle P_1, P_2, \ldots, P_k \rangle$ where $1 \leqslant i \leqslant n$, and p_i is a Web page in a navigation pattern. Moreover, $n\vec{p}_i$ has to be ordered as the same to user's active session window.

Step 2: User classifying based on LCS algorithm.

There are two sets, navigation patterns $n\vec{p}_i$ and active session window $\vec{S}$ as input of this step. Classifying algorithm attempts to find a navigation pattern (Cluster) by utilizing longest common subsequences algorithm. Navigation pattern with the highest degree of similarity is found according to the LCS algorithm to predict next user's activities and create a recommendation set.

Step 3: Predict user's next intention.

In this step, a set of Web pages is returned to the user as a recommendation set. Recommendation engine attempts to show a set of Web pages to the current user after the system finds a navigation pattern with highest degree of similarity. Meanwhile, Web pages in the recommendation set are ranked in terms of degree of connectivity between Web pages in the form of adjacency matrix M created during the off-line phase. Moreover, for improving the efficiency of recommendation, the recommendation engine shows only the Web pages with the highest degree of connectivity, while the rest of Web pages are not considered in the recommendation set. The new recommendation set is created by the next user's movement in the Web site. In this case, a new user session window will be created after each user activity.

The algorithm shown in Fig. 4 illustrates the steps performed in this phase. The cost of the recommendation algorithm in the recommendation phase is constant, which is $O(1)$.

To illustrate this process, consider the example of navigational pattern set given in Table 1. For this example, we consider 3 active session windows. Sequence $\vec{\omega} = \langle P_{17}, P_1, P_{27} \rangle$ is a set of Web pages in active session window that is ranked based on matrix M. The sequence $n\vec{p}_2 = \langle P_5, P_{17}, P_{32}, P_{40}, P_{13}, P_1, P_{27}, P_{43} \rangle$ is the cluster discovered by prediction engine based on LCS algorithm. For this example, it discovered the sequence $\vec{\gamma} = \langle P_5, P_{32}, P_{40}, P_{13}, P_{43} \rangle$ as a prediction list.

The user may choose a page from prediction list or may follow previews of activities. According to the new active session window, prediction engine will rebuild the predictions. For example, sequence $\vec{\omega} = \langle P_{27}, P_{37}, P_{18} \rangle$ is the new active session window and prediction engine searches for more related cluster based on LCS algorithm. The discovered cluster is probably different from the previous cluster found by LCS. For example, if $n\vec{p}_7 = \langle P_7, P_{37}, P_{31}, P_{29}, P_{18}, P_{26} \rangle$ is a new cluster discovered by LCS

```
Input:
   • Cleaned, filtered, and sessionized Log file.
   • MinFreq.
   • MinClusterSize.

Output:
   • A list of Clusters C

L[p] = P ;    // Assign all URLs to a list of web pages
for each (Pi, Pj) € L[p] do     // for all pair of web pages
  M (i,j)=WeightFormula (Pi, Pj);   //computing the weight based on
formula(3)
  Edge (i,j)=M (i,j);
end for

//There is an undirected Graph (E, V)

for all Edge (u, v) € Graph (E, V) do  // removing all edges that its
weight is below than MinFreq
  if Edge  (u ,v) < MinFreq then
    remove (Edge (u, v));
  end if
end for

for all vertices(u) € Graph (E, V)  do
      Cluster [i]=DFS (u);      //doing the DFS algorithm
      if cluster[i] <MinClusterSize  // remove the cluster that its size
is below than MinClusterSize
       remove (Cluster[i]);
      end if
      i=i+1
end for

return (Cluster);
```

Fig. 3. The clustering algorithm.

algorithm, then the new prediction could be $\vec{\gamma} = \langle P_7, P_{31}, P_{29}, P_{26} \rangle$. Before this prediction is suggested to the user, they will be ranked based on the values stored in the co-occurrence matrix *M*. This prediction engine module sorts the pages in the prediction list that is more strictly related to those in the session as compared to that determined classification.

4. System evaluation

A variety of techniques are used to measure the performance of the recommendation systems. Some experimentation has been done for characterizing the quality of the recommendations. The proposed recommendation system, WebPUM, is evaluated based on effectiveness measure. A system effectiveness is evaluated by using some parameters utilized during both off-line and online phase. In order to evaluate effectiveness of the WebPUM, several tests should be conducted for both online and off-line phase.

The quality of the clusters produced by navigation pattern mining module is evaluated in the off-line phase by a visit-coherence parameter introduced by Perkowitz and Etzioni (1999). The hypothesis is that users behave coherently during their navigation, i.e., pages within the same session are in general conceptually related. In this study, we use this concept to obtain a measure of quality for the proposed system.

Definition 3. Visit-coherence measures the percentage of the Web pages inside a user session, which belongs to the cluster that represents the session being considered.

Visit-coherence is utilized to evaluate the quality of the clusters (navigation patterns) produced during the off-line phase. Furthermore, visit-coherence quantifies a session intrinsic coherence. As in the PageGather system (Perkowitz & Etzioni, 1999), the basic assumption here is that the coherence hypotheses hold for every session.

To evaluate the visit-coherence, we split dataset into two halves after the pretreatment phase. The clustering task is applied on the first half dataset and the recommendation engine is employed on the second half dataset to create recommendations. Visit-coherence is then evaluated based on the recommendations. The second half of the dataset is known as evaluation dataset. In this study, parameter β is defined to measure the number of Web pages in every session i that belongs to a navigation pattern (cluster) found for that session as in (5)

$$\beta_i = \frac{|\{p \in S_i | p \in np_i\}|}{N_i} \tag{5}$$

where p is a page, S_i is ith session, np_i is the cluster representing i, and N_i is the number of pages in ith session. The average value for β over all Ns sessions in the evaluation part of dataset is shown as:

$$\alpha = \frac{\sum_{i=1}^{Ns} \beta_i}{Ns} \tag{6}$$

where α is percentage of the visit-coherence that should be considered for various range of *MinFreq*.

```
Input:
   • User's active session window S⃗ .
   • A set of navigation patterns np.

Output:
   • A set of web pages as recommendation

Sort ( S⃗ ); // sort active session window

For each npi € np⃗ do
  Sort (npi); // sort navigation pattern
End for

For each npi € np⃗ do
  AnswerSet=|LCS (npi, S⃗ )|; // find the LCS between navigation patterns and
active session

End for
  Mach_String=NP_String (Max (AnswerSet));   // Put the navigation pattern
  with maximum LCS to the Mach_String

  Prediction_Set= S⃗ - Mach_String;   // Create prediction set based on the
  difference between active session and fund navigation pattern

  Recommendation_set= Rank (Prediction_Set);  // Rank the prediction set
and recommend it to the user

Return (Recommendation_Set);
```

Fig. 4. The recommendation algorithm.

Table 1
Navigational patterns generated by clustering algorithm.

NP no.	Navigational pattern
1	$\langle P_2, P_{10}, P_{15}, P_{20}, P_8 \rangle$
2	$\langle P_5, P_{17}, P_{32}, P_{40}, P_{13}, P_1, P_{27}, P_{43} \rangle$
3	$\langle P_6, P_3, P_{11} \rangle$
4	$\langle P_{33}, P_{36}, P_{16}, P_{12}, P_9, P_{24}, P_{44} \rangle$
5	$\langle P_7, P_{37}, P_{31}, P_{29}, P_{18}, P_{26} \rangle$
6	$\langle P_{50}, P_{42}, P_{13}, P_{52}, P_{49}, P_{38}, P_{14} \rangle$

Another well-known evaluation parameter that is considered in clustering-based systems is outlier. Clustering-based outlier detection algorithm is utilized to locate outliers.

Definition 4. The outlier is a percentage of Web pages that does not belong to any navigation pattern (cluster), therefore does not contribute to the online phase.

Outliers are being discovered according to different values of *MinFreq* through application of the clustering algorithm. A great deal Web page outliers during the clustering phase indicate that the method is not efficient. Fig. 5 illustrates definition of outliers in this study.

Measuring the prediction accuracy in the recommendation system requires characterization of the quality of results obtained. To measure the quality of recommendations during the off-line phase, we use second half of the dataset after the dataset is divided into two halves; training set and evaluation set.

Each navigational pattern np_i (a session in the dataset) in the evaluation set is divided into two parts. The first n pageviews in np_i, used for generating predictions. The remaining part of np_i is used to evaluate the generated predictions. The active session window is used by the prediction engine as part of the user navigational patterns in order to produce a prediction set. We refer this

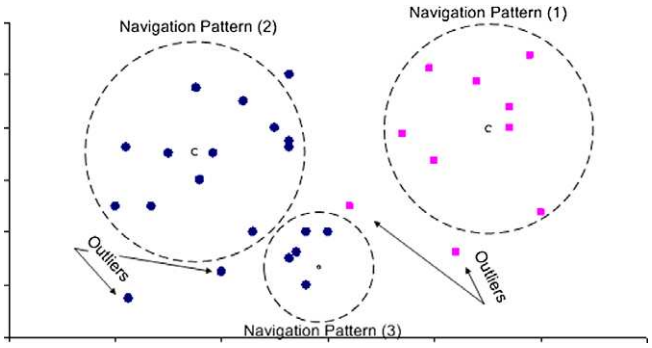

Fig. 5. Illustration of the outliers.

part as the navigational pattern *np*, the active session with respect to *np*, denoted by as_{np}. The prediction engine takes as_{np} and a recommendation threshold τ as inputs to produce a set of pageviews as prediction list. Recommendation threshold τ is the *MinFreq* and this prediction is denoted by $P(as_{np}, \tau)$. The set of pageviews $P(as_{np}, \tau)$ can now be compared with the remaining $|np| - n$, pageviews in *np*. This part is denoted as *np* by $eval_{np}$.

Fig. 6 illustrates the evaluation process. Our comparison of these sets is based on three different metrics, which are accuracy, coverage, and *F*1 measure.

The *Accuracy* of prediction set is defined as:

$$Accuracy(P(as_{np}, \tau)) = \frac{|P(as_{np}, \tau) \cap eval_{np}|}{P(as_{np}, \tau)} \tag{7}$$

where $|P(as_{np}, \tau) \cap eval_{np}|$ is a number of common Web pages in both prediction list and evaluation set. *Accuracy* is a number of

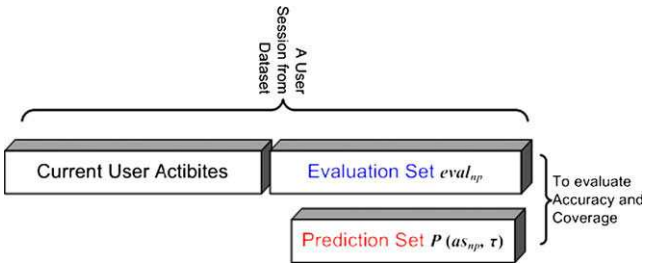

Fig. 6. Evaluation of the recommendations.

relevant Web pages retrieved and divided by the total number of Web pages in recommendations set.

Accuracy also measures the degree to which the prediction engine produces accurate recommendations. Another evaluation parameter in the online phase is *Coverage* that is defined as:

$$Coverage(P(as_{np},\tau)) = \frac{|P(as_{np},\tau) \cap eval_{np}|}{eval_{np}} \quad (8)$$

Coverage is the ratio between the number of relevant Web pages retrieved and the total number of Web pages that actually belongs to the user session. On the other hand, coverage measures the ability of the prediction engine to produce all of the pageviews that are likely to be visited by the user.

The *F1 measure* attains its maximum value when both accuracy and coverage are maximized. Finally, for a given prediction threshold τ, the mean over all navigational patterns in the evaluation set is computed as the overall evaluation score for each measure

$$F1 = \frac{2 \times Accuracy(P(as_{np},\tau)) \times Coverage(P(as_{np},\tau))}{Accuracy(P(as_{np},\tau)) + Coverage(P(as_{np},\tau))} \quad (9)$$

All parameters attempt to measure the quality of recommendation between 0 and 1, which is the range of *MinFreq*.

5. Experimental evaluation

In order to evaluate the performance of the proposed system, two main experiments have been conducted. In the first experiment, clustering algorithm is employed for navigation pattern mining. In the second experiment, prediction of the user next request has been performed by classification algorithm based on longest common subsequence (LCS).

5.1. System requirement for experimental evaluation

This section provides a summary of datasets used in the experimental evaluation as well as the hardware and software requirements to run the proposed system. All evaluation tests have been performed on a dual processor Intel® Core™ Duo CPU 2.4 GHz with 3.23 GBytes of RAM and Windows XP operating system. The implementations have been executed on .Net Framework 2 and VB.Net as well as C#.Net for coding purposes.

In our experiments, it is necessary to use such a dataset that allows us to analyze Web log data. Our experiments have been conducted on DePaul University CTI log file dataset (www.cs.depaul.edu) and MSNBC dataset.

5.2. Web-log pretreatment results

Well-known pretreatment algorithms have been done on the initial CTI dataset. The only cleaning step performed on this data was the removal of references to auxiliary files (e.g., image files). No other cleaning or preprocessing has been performed in the first phase. The data is in the original log format used by Microsoft IIS. The characteristics of the dataset we used are given in Table 2.

The length of active session window is important to classify current active session in the proposed recommendation system. The average number of Web pages in a session can be used for considering the length of active session. As shown in Fig. 7, the percentage of sessions formed by a predefined number of pages quickly decreases when the minimum number of pages in a session increases. Moreover, for CTI and MSNBC datasets, the average length of a user session is about three pages. Since we still have almost half of all the sessions, we then choose this value as the minimum length for an active session to be classified.

5.3. Results on navigation pattern mining

We created a weighted undirected graph *M* based on new method to assign weights to the graph. The results of the graph partitioning algorithm on graph *M* is a set of navigation pattern known as clusters. The weights of the graph for each pair of Web pages represent the degree of the connectivity between the Web pages. DFS search algorithm has been applied to find clusters of the connected components.

To evaluate the clustering method, the system results are compared with method proposed by Baraglia and Silvestri (2007). Their methodology was repeated by using our datasets. Consequently, the results are utilized as methods comparison.

Fig. 8 plots the number of clusters based on *MinFreq*, for both previous work and the proposed method that run on the CTI dataset. In the proposed method, the number of clusters increased up to *MinFreq* = 0.4. Obviously, the system obtains less connected components of the graph if we discard more edges from the graph according to *MinFreq*. Moreover, higher values of *MinFreq* (*MinFreq* = 0.4) creates graph with highly disconnected components, thus the clusters found are smaller than *MinClusterSize* and these clusters are discarded. Therefore the total number of clusters found does not increase.

Fig. 9 plots the number of clusters created based on the previous work and the proposed method for different values of *MinFreq* for the MSNBC dataset. Similar reasoning can be used to describe the behavior of the curves. Specification of the MSNBC dataset shows that the number of Web pages categories is about 20 and the dataset consists of more than 120,000 sessions. In this experiment, we run the clustering algorithm on the sessions that consist of only the categories of Web pages in the MSNBC Web site. The details of each session had not been captured in the dataset.

The number of the clusters is insignificant enough to evaluate the performance of the clustering. Outliers and visit-coherence are two other main parameters that we measured to evaluate the quality of clustering.

Fig. 10 shows the percentage of outliers for the CTI and the MSNBC datasets. The percentage of outliers increases for higher values of *MinFreq*. Moreover, higher values of *MinFreq* creates graph with highly disconnected components, thus the size of clusters is decreased in this case. Therefore, the number of Web pages that do not belong to any cluster increase for higher values of the *MinFreq*.

To measure the quality of clustering, visit-coherence is an evaluation parameter that can be utilized to describe the accuracy of the clustering. To evaluate visit-coherence, we split the datasets obtained from the pretreatment phase into two halves; applied clustering based on graph partitioning algorithm on one half and measure the quality of the clusters based on the second half of the datasets. Clustering has also been performed in the previous works.

Fig. 11 plots the visit-coherence for two datasets. Mostly, the percentage of visit-coherence in the proposed system is higher

Table 2
Dataset used in the Experiment

Dataset	Initial dataset			Dataset after pretreatment			
	Size (MB)	Records (thousand)	Period (days)	Size (MB)	Num. of users	Num. of Web pages	Num. of sessions
CTI	260	1051	30	10	5446	800	20,950
MSNBC	–	–	–	12	989,818	20	121,197

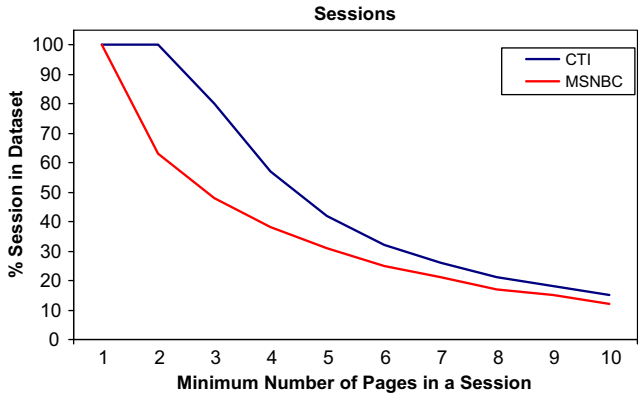

Fig. 7. Minimum number of pages in a session.

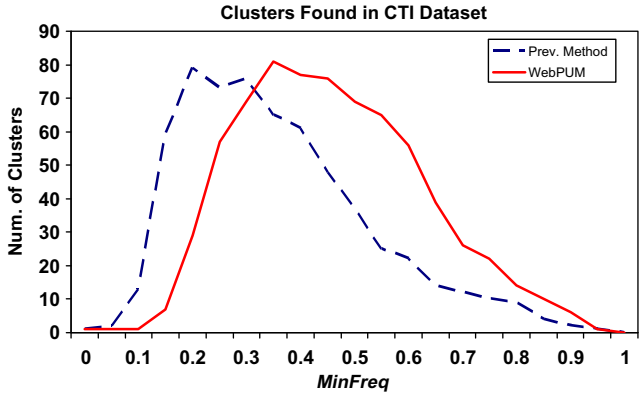

Fig. 8. Number of cluster found for CTI dataset.

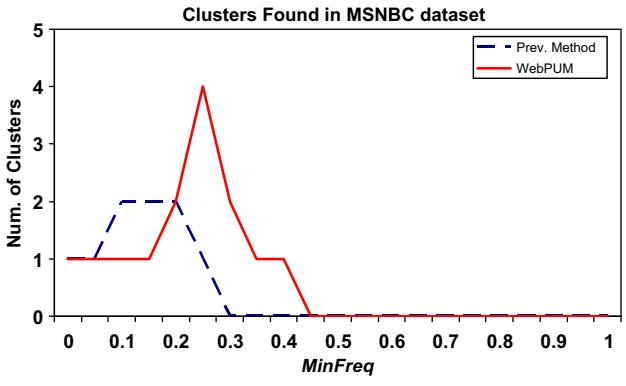

Fig. 9. Number of cluster found for MSNBC dataset.

than previous work according to the difference values of the *MinFreq*. Moreover, for the higher values of the *MinFreq* the quality of the clustering grow less in view of the fact that the system creates less number of the clusters for high values of the *MinFreq*.

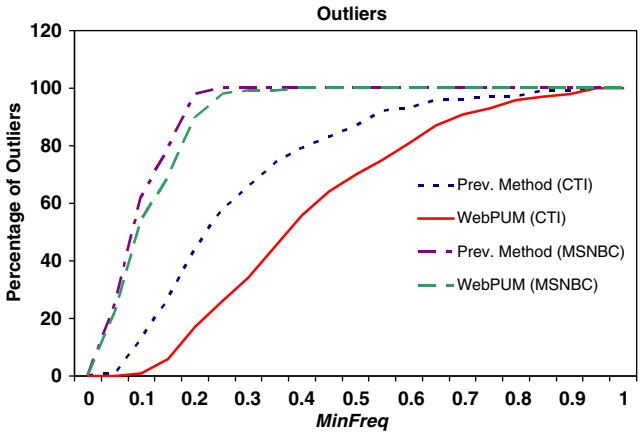

Fig. 10. Percentage of outliers.

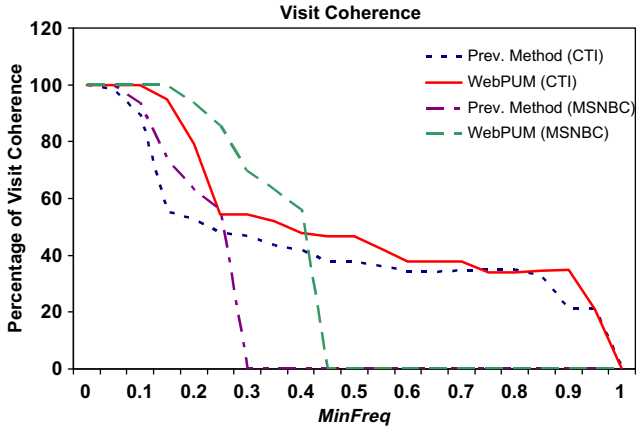

Fig. 11. Visit-coherence in two datasets.

In addition, there are less numbers of Web pages in connected components for the higher value of *MinFreq*. Furthermore, the clusters consist of a great deal of Web pages for small values of the *MinFreq*, consequently, number of Web pages that belong to the particular cluster increases in this case.

5.4. Evaluation of the recommendations

In this section, we measure the quality of the recommendations generated by prediction engine during the online phase of WebPUM. Sessions found in one half of both datasets are submitted to the prediction engine to classify current user activities and to generate recommendations. Subsequently, overlapping between the generated prediction $P(as_{np}, \tau)$ and the session pages $eval_{np}$ is computed by using the formula (8) introduced in Section 3. Finally, the percentage of all predictions represents the quality of recommendations by WebPUM system.

In this study, we have evaluated the quality of the recommendations for WebPUM system versus the previous method introduced

in Baraglia and Silvestri (2007). Three parameters have been measured to verify the quality of the predictions, which are accuracy, coverage and *F*1.

Fig. 12 depicts the accuracy of the proposed system for *MinFreq* ranging from 0 to 1 for the CTI and the MSNBC datasets. The percentage of accuracy for the proposed system achieved the best results when we choose the value of *MinFreq* to be around 0.55 for CTI dataset. The results prove that the approach is able to improve the accuracy of recommendations in a great deal of different values of *MinFreq* as compared to the previous works.

Fig. 13 depicts the coverage of the proposed system for *MinFreq* ranging from 0 to 1 for both datasets. The coverage achieved high percentage for the lower values of *MinFreq*. In this case, for the lower values of *MinFreq* by classifying current user activities, the prediction engine is able to find clusters with large size with more Web pages inside the cluster belong to the actual user's activities. Moreover, similar reasoning can be used for the higher values of *MinFreq* which decreased the percentage of the coverage. The results illustrates that the approach is able to improve coverage of the recommendation systems in comparison with the previous works.

Fig. 14 depict the *F*1 measure for *MinFreq* ranging from 0 to 1 for the CTI and the MSNBC datasets. Most of the time, *F*1 achieves higher percentages for the proposed system than the previous works.

For all experiments, we run paired *t*-test with 95% confidence. Paired *t*-test is used to compare means on the same or related subject over time or in differing circumstances. On the other hand, the *t*-test assesses whether the means of two groups are statistically different from each other. This analysis is appropriate in comparing the means of two groups. The assumption for using the paired *t*-test is that the observed data are from the same subject or from a matched subject and are drawn from a population with a normal distribution. A paired *t*-test is carried out to compare the experimental results for *F*1 measure. The mean of *F*1 measure for the CTI dataset is 19.5 for the proposed approach and 7.9 for the previous work. The two-tail *p*-value for the paired *t*-test is 0.0006 with a significant value of (p-value $\leqslant 0.05$).

Meanwhile, the mean of *F*1 measure for the MSNBC dataset is 6 for the proposed approach and 2 for the previous work. The two-tail *p*-value for the paired *t*-test is 0.009 with a significant value of (p-value $\leqslant 0.05$). The implication of this result is that the proposed approach is more accurate than the previous works.

In Web-based recommendation systems, scalability of the proposed method is also taken into account. In this paper, we run two experiments based on *MinFreq* that achieves maximum accuracy in both the proposed system and the previous method. In the first experiment, we divide dataset into some parts, whereby each part is a percentage of the original dataset. We started the experiment from 10% of the dataset for both dataset. It was observed from Fig. 15 that both dataset were not affected by varying the percentage values.

In the second experiment, we tested the scalability based on the number of users in the dataset. We began the experiment with 10% users in the dataset. In the each trial, we then raised 10% of the user to the initial dataset. Fig. 16 plots the accuracy based on the percentage of users in the dataset. The percentage of the users is not affected in the results of accuracy.

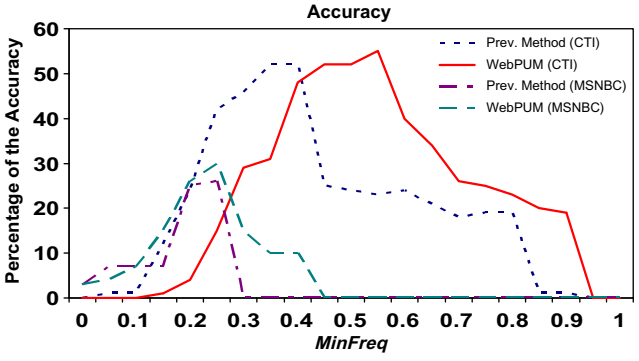

Fig. 12. Accuracy of the recommendations.

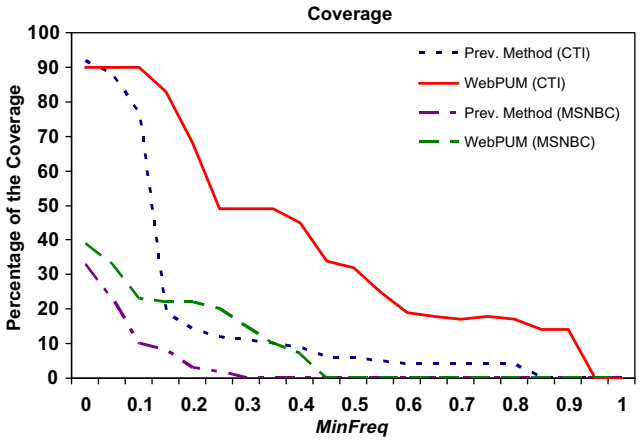

Fig. 13. Coverage of the recommendations.

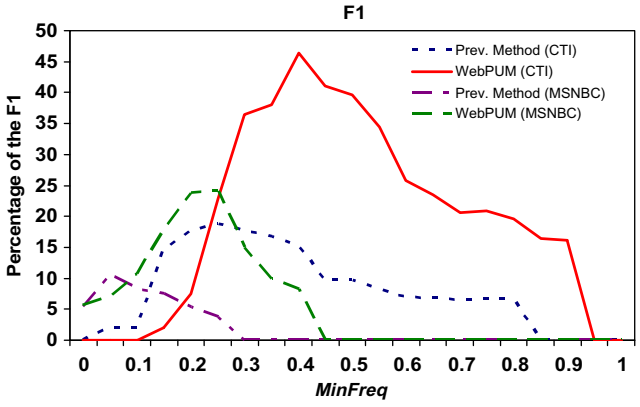

Fig. 14. *F*1 measure of the recommendations.

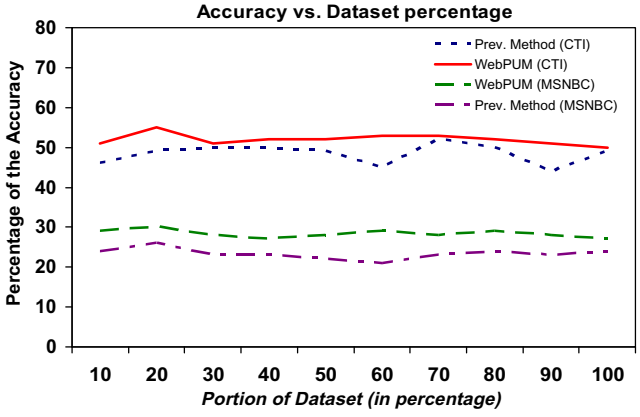

Fig. 15. Accuracy vs. dataset percentage.

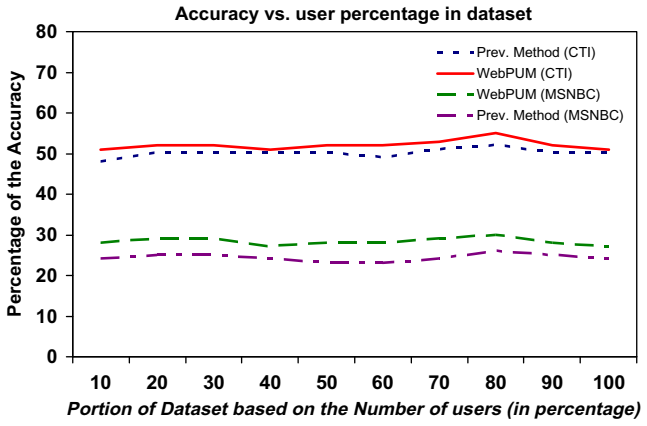

Fig. 16. Accuracy vs. number of user in dataset.

The experimental results indicates that our approach for classifying current user's activities to predict user next request is able to improve the quality of predictions in the Web usage mining recommendation systems.

6. Conclusions and future works

In this paper, we advanced a Web usage mining architecture called WebPUM and proposed a novel approach to classify user navigation pattern for online prediction of user future intentions through mining Web server logs. We also utilized graph a partitioning algorithm to model user navigation patterns. In order to mine user navigation patterns, we established an undirected graph based on connectivity between each pair of the Web pages. Next, we proposed a novel formula for assigning weights to edges of the undirected graph. To classify current user activities we applied the longest common subsequence algorithm to predict user near future movement. We used some evaluation methodologies to evaluate the quality of clusters found and quality of recommendations. The experimental results show that our approach improved the quality of clustering for user navigation pattern and the quality of recommendations for both CTI and MSNBC datasets.

There are a number of aspects that merit further improvement by the system. First is to take into account the semantic knowledge about underlying domain to improve the quality of the recommendations. Second is to integrate semantic Web and Web usage mining in achieving best recommendations from the dynamic and huge Web sites.

References

Aho, A. V., Hirschberg, D. S., & Ullman, J. D. (1974). *Bounds on the complexity of the longest common subsequence problem*. Paper presented at the IEEE conference record of 15th annual symposium on switching the University of California at Los Angeles.

Apostolico, A. (1997). String editing and longest common subsequences. *Handbook of Formal Languages, 2*, 361–398.

Baraglia, R., & Silvestri, F. (2004). *An online recommender system for large Web sites*. Paper presented at the IEEE/WIC/ACM international conference on Web Beijing, China.

Baraglia, R., & Silvestri, F. (2007). Dynamic personalization of Web sites without user intervention. *Communications of the ACM, 50*(2), 63–67.

Berendt, B., Mobasher, B., Spiliopoulou, M., & Wiltshire, J. (2001). *Measuring the accuracy of sessionizers for Web usage analysis*. Paper presented at the proceedings of the Web mining workshop at the first SIAM.

Cadez, I., Heckerman, D., Meek, C., Smyth, P., & White, S. (2000). *Visualization of navigation patterns on a Web site using model-based clustering*. Paper presented at the proceedings of the sixth ACM SIGKDD international conference on data mining and knowledge discovery, Boston, Massachusetts, United States.

Cooley, R., Mobasher, B., & Srivastava, J. (1997). *Web mining: Information and pattern discovery on the World Wide Web*. Paper presented at the ninth IEEE international conference on tools with artificial intelligence, Newport Beach, CA, USA.

Cooley, R., Mobasher, B., & Srivastava, J. (1999). Data preparation for mining World Wide Web browsing patterns. *Knowledge and Information Systems, 1*(1), 5–32.

Dancik, V. (1994). *Expected length of longest common subsequences*. University of Warwick: Department of Computer Science, University of Warwick.

Gormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (1990). Introduction to algorithms. *MIT Press, 44*, 97–138.

Hirschberg, D. S. (1977). Algorithms for the longest common subsequence problem. *Journal of the ACM (JACM), 24*(4), 664–675.

Huan, J., & Kamber, M. (2000). *Data mining: Concept and techniques*. San Mateo, CA: Morgan-Kaufmann.

Liu, H., & Kešelj, V. (2007). Combined mining of Web server logs and Web contents for classifying user navigation patterns and predicting users' future requests. *Data & Knowledge Engineering, 61*(2), 304–330.

Mobasher, B., Cooley, R., & Srivastava, J. (1999). *Creating adaptive Web sites through usage-based clustering of URLs.* Paper presented at the knowledge and data engineering exchange, Chicago, IL, USA (pp. 19–25).

Mobasher, B., Cooley, R., & Srivastava, J. (2000). Automatic personalization based on Web usage mining. *Communications of the ACM, 43*(8), 142–151.

Nakagawa, M., & Mobasher, B. (2003). *A hybrid Web personalization model based on site connectivity*. Paper presented at the fifth WEBKDD workshop, Washington, DC, USA (pp. 59–70).

Perkowitz, M., & Etzioni, O. (1999). *Adaptive Web sites: Conceptual cluster mining*. Paper presented at the international joint conference on artificial intelligence, Stockholm, Sweden.

Perkowitz, M., & Etzioni, O. (2000a). Adaptive Web sites. *Communications of the ACM, 43*(8), 152–158.

Perkowitz, M., & Etzioni, O. (2000b). Towards adaptive Web sites: Conceptual framework and case study. *Artificial Intelligence, 118*(1–2), 245–275.

Shahabi, C., Banaei-Kashani, F., Chen, Y. S., & McLeod, D. (2001). Yoda: An accurate and scalable Web-based recommendation system. *Lecture Notes in Computer Science*, 418–432.

Spiliopoulou, M., Mobasher, B., Berendt, B., & Nakagawa, M. (2003). A Framework for the evaluation of session reconstruction heuristics in Web-usage analysis. *INFORMS Journal on Computing, 15*(2), 171–190.

Srivastava, J., Cooley, R., Deshpande, M., & Tan, P. N. (2000). Web usage mining: discovery and applications of usage patterns from Web data. *ACM SIGKDD Explorations Newsletter, 1*(2), 12–23.

Wang, J., & NetLibrary, I. (2006). *Encyclopedia of data warehousing and mining*: Idea group Reference.

Yan, T. W., Jacobsen, M., Garcia-Molina, H., & Dayal, U. (1996). From user access patterns to dynamic hypertext linking. *Computer Networks and ISDN Systems, 28*(7–11), 1007–1014.

Zhou, B., Hui, S.C., & Chang, K. (2004). *An intelligent recommender system using sequential Web access patterns*. Paper presented at the IEEE conference on cybernetics and intelligent systems, Singapore (pp. 393–398).