Preference-Function Algorithm: a novel approach for selection of the users’ preferred websites

S. Kami Makki* and Seema Jani

Department of Electrical Engineering and Computer Science,
College of Engineering, University of Toledo,
2801 W. Bancroft St., Toledo, Ohio 43606, USA
E-mail: kmakki@eng.utoledo.edu
*Corresponding author

Xiaohua Jia

Department of Computer Science,
City University of Hong Kong,
83 Tat Chee Ave, Kowloon, Hong Kong
E-mail: jia@cs.cityu.edu.hk

Wuxu Peng

Department of Computer Science,
Texas State University,
San Marcos, TX 78666, USA
E-mail: wuxu@txstate.edu

Abstract: Designing a website which is helpful to its users requires knowledge of the users’ preferences and their motivations. Therefore, a designer requires to anticipate the users’ needs and structures the website accordingly. This paper implements a novel approach for selecting the users’ preferred web pages. In this approach, the navigated web pages are modelled as a finite state graph, where each visited web page is defined as a state. This graph then is used to provide the framework for determining the users’ interest. The viability of this approach is demonstrated with a user-created website scenario.

Keywords: web mining; clustering; classification; personalisation; customisation; usage mining; pattern discovery.


Biographical notes: S. Kami Makki has received his PhD from the University of Queensland, Australia. He is now with the Department of Electrical Engineering and Computer Science at the University of Toledo. He is an active researcher in the fields of distributed systems, databases, mobile, wireless networks and information systems. He has numerous publications in peer-reviewed journals and international proceedings. He is an Editor of International Journal of Infnomics and International Journal of Communications. He has also served as a Technical Program Chair for a number of IEEE and ACM sponsored technical conferences.

Copyright © 2007 Inderscience Enterprises Ltd.
1 Introduction

The continuous growth of the world wide web poses many challenging issues for a person using it. Most web structures are large and complicated and users often miss the goal of their inquiry, or receive ambiguous results when they try to navigate through the web pages (Magdalini and Vazirgiannis, 2003). The business environment is such that in order to do well it is important to meet the needs of the customers and also create revenue by doing so. Therefore, the need to retain the attention of the users of a website and understand the needs of those users leads to the importance of analysing the users’ behaviour. Once the behaviour of the users is known, a company can use this information for a variety of objectives and actions such as, to personalise the website, make recommendations, enhance the website and target advertise.

The objective of a web personalisation system is to “provide users with information they want or need, without expecting them to ask for it explicitly” (Mulvenna et al., 2000). This objective, in fact, is three fold: how can a company dynamically customise their website on the fly, how can a company use different filtering techniques to select related data to identify the users’ patterns, and lastly, how can a company store, effectively, the pertinent data for personalising their website? The implications of these questions are extremely significant to e-commerce as tailoring a website to a user’s needs can greatly increase the revenue of a business (Chen and Liu, 2005).

Personalising a website can generally be done in various ways. One way is to have the information provided by the users via questionnaires, surveys or registration forms. Another approach is to use demographic, geographic, or psychographic profiles or other information to divide or segment large populations into smaller groups (http://www.cs.uu.nl/docs/vakken/e-com/personalisation.pdf). In this approach the following have been used in one or any combination (Pan et al., 2006):
• content-based filtering uses the users’ behaviours and preferences to suggest items or links which are related to the users’ interests
• collaborative filtering makes comparison on one user interest against other users who might share the same interest in order to generate recommendation on the basis of the common interest
• rule-based filtering allows website administrators to specify rules based on static or dynamic profiles that are used to affect the content served to a particular user.

The last approach is to seek to understand the behavioural preferences of a specific, individual user and then deliver website content specifically targeted at that person. This is known as web usage mining (Magdalini and Vazirgianannis, 2003; Pan et al., 2006; Srivastava et al., 2000; Nath, 2004; Shahabi and Banaei-Kashani, 2001) and is the one that this paper targets. This method is more dynamic in nature and specifically takes into account the navigational tendencies of the user (usage data) within a specific website.

Web usage mining can be regarded as a three-phase process, consisting of data preparation, pattern discovery, and pattern analysis phases (Srivastava et al., 2000). In data preparation phase, web server logs are processed in order to identify users, sessions, viewed pages and so on. This phase is the most time-consuming but technically uncomplicated phase. In pattern discovery phase, methods such as those in the area of Statistics or Artificial Intelligence are applied to detect interesting patterns. This phase is the most diverse and expanding of all three phases and has been the subject of continuing research and advancement. In the final phase of the web usage mining process, the detected patterns from the second phase are analysed and explored further. Recent areas of these researches will be expanded in Section 2. One main area of web usage mining is identifying users’ preferred or target websites, where the information obtained from the second phase is used in conjunction with the specific web page relationships and actions. These relationships and actions are determined by the owners of the website.

2 Related research

Web customisation has generated a lot of interest from the early days of usage for commercial purposes. It is generally defined as predictive analysis of consumer data used to adapt targeted media, advertising, or merchandising to consumer needs (Chen and Liu, 2005). Therefore it tries to adapt the content, structure and presentation of the web to each individual user’s characteristics, behaviour, and environment. Of late, many commercial websites have been taking advantage of this technology and gaining popularity for their provided commercial and social services, such as Amazon.com, Overstock.com, Borders.com Facebook.com, MySpace.com, and others. They use a variety of techniques (generally identified as explicit or implicit) to collect information regarding the users and their preferences (usage data). The collected information will be processed to recognise the general users’ access pattern for creation of not only a realistic user profile but also an adaptive content, and a user centric structure for a website.

Therefore, pattern discovery and analysis in the context of web usage mining has been the subject of numerous research works, and a variety of data mining techniques such as Statistics, Artificial Intelligence, Neural networks, and Genetic algorithms have been utilised. These methods and their applications have been discussed extensively
in the literature (Chen and Liu, 2005; Lagus et al., 2004; Nath, 2004; Agrawal and Srikant, 1994). The Statistics and Artificial Intelligence methods have predominantly being used for web usage mining. The first method uses Statistics that consists of a range of applications from overall analysis to an adapted version of statistical data mining techniques (Srivastava et al., 2000). These techniques are those that mine for rules using the pre-defined support and confidence values. The second approach uses Artificial Intelligence techniques which draw upon methods and algorithms developed from machine learning and pattern recognition (Srivastava et al., 2000). The next two sections provide a brief description of these two methods.

2.1 Statistical mining techniques

The research and application of statistical techniques in the analysis of web usage data is a wide area. Three specific research areas are covered in this section. These are the overall analysis of the web usage data, probability analysis and lastly, the standard data mining techniques.

The overall analysis uses statistical techniques, which are the most common techniques to extract knowledge about the users of a website. There are different levels of analysis which have been used from the area of Statistics in web mining. One general area is the use of overall statistical information, which is given by some software programs. Several commercial software packages such as Analog (http://www.statslab.cam.ac.uk/~sret1/analog) and OLAP (http://www.olapreport.com/) are available for web log analysis. Common reports include a list of the most requested web pages, a summary report, and a list of the browsers used. An example of a partial report generated by OLAP is given in Appendix A (Berendt, 2002). Currently, these packages provide limited mechanisms for reporting user activity (Yao et al., 2002). The given information provides a quick overview of how a website is being used. It requires minimal disk space or processing power; however, it does not give the ability to dig deeper into the data. This type of analysis does not provide information about particular users and is mainly used for improving system performance, enhancing the security of the system, facilitating website modification tasks and providing support for marketing decisions.

Probability Analysis is another statistical approach which has been used often in past research, and it is basically the use of probability in the form of Markov chain modelling (Borges, 2004). The Markov chain is defined as a set of states \( X \), a transition matrix \( T \), and a vector of initial probabilities \( \pi \). The set of states \( X \) is composed of the start state \( S \), the final state \( F \), and the states that correspond to the web pages that have been visited. The transition matrix records the transition probabilities, which are estimated by the proportion of times the corresponding link was traversed from the anchor. The initial probability of a state is estimated as the proportion of times the corresponding web page was requested by the user.

Another well known area of Statistics is data mining techniques, which has been used in the area of web usage mining. These techniques involve finding association rules within a set of data and mining by determining the rules based on the rules’ confidence and support (Agrawal and Srikant, 1994). One example of work done in this area is with the application of the ‘a priori’ algorithm. The association rule mining searches for relationships between items in the data. Basic concepts are that, there is
a set \( I, I = \{i_1, i_2, \ldots, i_m\} \), of items and a set \( T, T = \{t_1, t_2, \ldots, t_m\} \), of transactions. Therefore, each transaction \( t_i \) is a set of items such that \( t_i \subseteq I \) (Borges, 2004).

An Association Rule is an implication on itemsets \( X \) and \( Y \), denoted by \( X \Rightarrow Y \), where \( X \subseteq I, Y \subseteq I, X \cap Y = \emptyset \). The rule meets a minimum confidence of \( c \), meaning that \( c\% \) of transactions in \( T \) which contain \( X \) also contain \( Y \) (Mobasher et al., 2002). In addition, for each itemset a minimum support of \( s \) must be satisfied (Mobasher et al., 2002; Berendt, 2001):

\[
\begin{align*}
    s & \leq |X \cup Y|/|T| \\
    c & \leq |X \cup Y|/|X|.
\end{align*}
\]

One approach taken in using the association rules is to store discovered frequent itemsets into an ‘itemset graph’ (Mobasher et al., 2001). Each node at depth \( d \) in the graph corresponds to an itemset \( l \) of size \( d \) and is linked to itemsets of size \( d + 1 \) that contain \( l \) at level \( d + 1 \). The single root node at level 0 corresponds to the empty itemset. The frequent itemsets are matched against a user’s active session \( S \) by performing a search of the graph to depth \( |S| \). A recommendation \( r \) is an item at level \( |S| + 1 \) whose recommendation score is the confidence of rule \( S \Rightarrow r \).

### 2.2 Artificial intelligence techniques

The second type of technique involves the use of Artificial Intelligence techniques such as clustering and classification. Clustering and classification are machine learning techniques that are used to group together a set of items having similar characteristics (Mobasher et al., 1999; Baglioni et al., 2003; Shahabi et al., 2001; Giannakakis et al., 2003; Perrin and Petry, 2003). In the Web Usage domain, there are two kinds of interesting clusters to be discovered: usage clusters and page clusters. Clustering of users tends to establish groups of users exhibiting similar browsing patterns. Clustering of pages, however, discovers groups of pages having related content. Classification is the task of mapping a data item into one of several predefined classes. Classification is done by using supervised inductive learning algorithms such as decision tree classifiers, naive Bayesian classifiers, k-nearest neighbour classifiers, Support Vector Machines, and other similar methods.

Several researches use clustering approach to identify the general usage patterns. For example, Mobasher et al. (1999) modelled the profile based on the clustering approach and named it PACT, which stands for Profile Aggregations based on Clustering Transactions. The goal is to effectively capture common usage patterns from potentially anonymous click-stream data. The data are pre-processed and then used to create the profile. Pre-processing of the data is done in two steps: identifying users and determining pageviews \( p_i \)'s, where \( 1 \leq i \leq n \). In the first step, unique users are identified from the anonymous usage data, and the erroneous or redundant references are removed from the data. In the second step, the pageviews are identified. The pageview identification is the task of determining which page file accesses has contributed to a single browse display. Relevant pageviews are included in transaction files and weights are assigned to reflect the significance of the pageviews. The pre-processing tasks will result in two sets:
Preference-Function Algorithm: a novel approach

- a set of \( n \) pageviews, \( P = \{p_1, p_2, \ldots, p_n\} \), appearing in the transaction file with each pageview uniquely represented by its associated web page
- a set of \( m \) user transactions, \( T = \{t_1, t_2, \ldots, t_m\} \), where each \( t_i \in T \) is a subset of \( P \).

Each transaction \( t_i \), where \( 1 \leq k \leq n \), is set up as a pageview vector. The elements of this vector are \( w(p_i, t) \), where \( 1 \leq i \leq n \), and are the weights in the transaction vector \( t \), associated with the pageview. The transaction \( t \) vector is shown as

\[
t = \langle w(p_1, t), w(p_2, t), \ldots, w(p_n, t) \rangle.
\]

Using a standard clustering algorithm such as k-means, the transaction vectors are partitioned into groups of transactions which are close to each other and also based on a measure of distance or similarity. Such clustering will result in a set \( TC = \{c_1, c_2, \ldots, c_k\} \) of transaction clusters, where each \( c_i \) is a subset of the set of transactions \( T \). However, transaction clusters by themselves are not an effective means of capturing a view of common user profiles. The ultimate goal in clustering is to reduce these clusters into weighted collections of pageviews which represent profiles. Depending on the clustering method used, a weight is associated with each web page \( u \) in a cluster \( c \), which is denoted by \( \text{weight}(u, c) \). Each cluster \( c \) is represented as \( \hat{c} \).

\[
\hat{c} = \langle u_1^c, u_2^c, \ldots, u_n^c \rangle
\]

where \( u_i^c = \text{weight}(url_i, c) \) if \( url_i \in c \) or 0 otherwise.

The Aggregate Usage Profile is determined by taking from the set of pageview-weight pairs those that fit the following criteria: for transaction cluster \( C \), select each pageview \( p_i \) such that \( u_i^c \) is greater than a pre-specified threshold. The pre-specified thresholds are the support and interest of an item set. These concepts are those used in traditional data mining techniques, but are incorporated into the clustering method. A support threshold is specified before mining and is used by ‘a priori’ algorithm for pruning the search space. The itemsets which are returned by the algorithm satisfy this minimum support threshold. The clustering approach, however, requires the structure of the website to be known.

Classification is also used to determine user preferences (Baglioni et al., 2003). Classification algorithms require training data as inputs. As an example, the input is a set of cases whereby each case specifies values for a collection of attributes and for a class. The output of the classification algorithm is a model that describes or predicts the class value of a case on the basis of the values of the attributes of the case. The predictive accuracy of the extracted model is evaluated on a test set for which the actual class is known.

This method requires registered users to provide information about themselves. The registered users’ information is broken up, whereby 67% become the training set and 33% become the test set. The attributes of a class consists of the site pages or sections visited by the users and the class consists of the users’ sex. In this case the goal is to accurately determine the user’s sex based on the web pages that a user visited. The classification accuracy of this approach is 54.8%. The major drawback is the information about the users which has to be obtained ahead of time.
3 The proposed solution

To extract the related information from the users’ log, we use a new approach called the Preference-Function Algorithm (PFA). The Preference-Function (PF) consists of two components defined as the Likeness-Factor (LF) and the Time-Factor (TF). The LF provides an overall idea as to whether or not a particular web page has been visited frequently by a user. The TF provides the overall interest of that web page. These two components are further elaborated upon in the next few sections. Also in this approach, knowledge of the structure of the website is not required because this knowledge will be gained from the user’s specific actions. The overall process is divided into two steps as shown in Figures 1 and 2.

In step 1, the user profile will be determined by pre-processing the data, which is also known as data preparation, and then the specific usage mining is performed using the PF.

Figure 1 Step 1 in Preference-Function web mining

The data preparation part consists of creating useful information from the log files. Examples of log files and the meanings of the specific entries in the log files are given in Appendix B. The log files contain many entries that are irrelevant or redundant for the mining task. Initially, the raw data are cleaned, which includes removing all redundant or unnecessary information. All information related to image files and map files, which exist in the log files, is irrelevant to identifying the user’s behaviour. Then, the sessions and transactions are identified based on the available data. A session length can be defined to be a period of time (for example a 6 h or 12 h length) and will start when the first page is requested from a designated website. The end of the session will be determined when the user leaves the website, or when the time on one web page has exceeded a predefined threshold (for example a 30 min length). This 30 min timeout assumption is based on the results from Catedge and Pitkow (1995). Various transactions can occur during each session. Individual entries for page accesses are grouped into meaningful transactions. The transactions are first defined as being unique and are grouped together based on the IP addresses, since any access from different IP addresses is identified as a different transaction.

Step 2, as shown in Figure 2, involves making recommendations based on a company-provided scenario. In the case of this project, the website owner will include the suggested actions in the scenario.
Figure 2  Step 2 in Preference-Function web mining

3.1 Preference-Function (PF)

As the topology of the website is not known, information on the server log will provide a way to reconstruct a partial topology. The following assumptions will provide a framework for analysing the server logs:

- a session is to start when the first page of a designated website is requested from any IP address
- the beginning of a user session is determined to be the first time a unique IP address is observed
- the end of the session will be determined when the user has left the website or when a timeout (for example 30 min) has been exceeded on one web page (Catledge and Pitkow, 1995)
- each unique IP address will be compared with those that share the same operating system
- during a session, visitation of previous web pages is allowed.

Relevant information will be extracted from the server logs and combined to determine the PF. The PF will be determined from the multiplication of the two variables: LF and TF.

3.2 Likeness-Factor (LF)

In this paper, the sessions are modelled as a finite state machine, in which each visited web page within a website is defined as one state. Two additional states also will be added, S and F, to denote the start and final states. The transition from each state to the next will be denoted as an edge on a weighted directed graph. The LF is determined by summing the path weights \( pw \) of the individual paths that were used to reach a particular state. The path weights are defined as follows: each unique path toward a particular state, by definition, will be given a partial path weight of 1; any repetition that takes place, such as leaving the current page and then returning, will be given a partial path index of 1/2; and the total path weight for the path is determined by summing the values for the partial path weight of the unique path plus each of the various detours. The LF is determined by adding various path weights to a particular state. The path weight of a state \( s \) is determined from equation (1) where ‘\( n \)’ is the number of detours once a specified state is reached:

\[
pw_s = 1 + \sum_{i=1}^{n} pw_i.
\]
The formula for the LF is given by summing up all the path weights of all the paths \((m)\) to a particular state \((s)\):

\[
LF_s = \sum_{j=1}^{m} pw_{sj}.
\]  

### 3.3 Time-Factor (TF)

As mentioned in Cooley et al. (1997) the reference-length approach is based on the assumption that the amount of time a user spends examining an object is related to the interest of the user for the object’s contents. However, it needs to be mentioned that there are cases where either some users leave a page unattended for a long period of time, or they are extremely slow on a web page that is not of their interest; or the length of their visit to each page may be very close to the predefined threshold (predefined 30 min); or they may have reached to a page by mistake which is not of their interest, but they have already spent a good amount of time to discover that they are looking at a wrong page; or other similar cases. These cases can be detected by examining the log files and eliminating them from the result of the experiments or through other measure such as making the user respond or visit a certain spot in each page after an interval, depending on the page intricacies. It should also be mentioned that the removal of unnecessary information from the log files allows longer period storage time for the log files (as shown in the example of ‘sever log analysis’ on Section 5.1). On these bases, a model for user sessions is obtained by distinguishing the navigational objects (i.e., containing only links interesting to the user) from the content objects (i.e., containing the information the user was looking for). The distinction between navigational and content accesses is related to the distance (in time) between one request and the next. This TF is added to the evaluation function which was developed above in equations (1) and (2).

The TF considers that the more time a user spends on an object, the higher interest the user has for the information on that object and consequently the web page that contains the object. (With the assumption that, false long stoppage are identified and eliminated as mentioned above, also bearing in mind that the process of identifying the exact preferred web pages is tend to be more a trial and error process than one time process.) Therefore, the TF is determined by dividing the time (sec) at a single web page (determined from the web server logs) by the total time for that session, where ‘\(l\)’ is the number of sessions. The formula for TF is shown in equation (3) where ‘\(l\)’ is the number of total states and ‘\(p\)’ is the number of times that the state ‘\(s\)’ appears:

\[
TF_s = \frac{\sum_{k=1}^{p} t_{sk}}{\sum_{k=1}^{l} t_k}.
\]  

Therefore, from equations (2) and (3) we can compute the PF as follow:

\[
\text{preference-function}(PF_w) = \text{Likeness-factor}(LF_w) \times \text{Time-factor}(TF_w).
\]
3.4 Example

Suppose we have a website which consists of web pages A, B and C, and based on the available server log it is known that a unique user used the following paths:

\[
\{A \rightarrow B \rightarrow C \rightarrow B\} \\
\{A \rightarrow C \rightarrow B \rightarrow C \rightarrow C\} \\
\{B \rightarrow C \rightarrow B\} \\
\{C \rightarrow A \rightarrow B\}.
\]

Based on the first path, the path weights \((pw)\) would be calculated as follows:

\[
pw_{A \rightarrow B \rightarrow C} = 1 \\
pw_{C \rightarrow B} = 1/2
\]

Therefore the \((pw_{B1})\) for this path would be:

\[
pw_{B1} = 1 + 1/2 = 1\frac{1}{2}.
\]

For the second path the \(pw\) session and the \(pw_B\) would be as follows:

Therefore, the \(pw_{B2}\) for the second path would be:

\[
pw_{B2} = 2.
\]

For the third path the \(pw\) session and the \(pw_B\) would be as follows:
Therefore the $pw_{B3}$ for the third path would be:

$$pw_{B3} = 1 + \frac{1}{2} = 1\frac{1}{2}.$$ 

For the fourth path the $pw$ session and the $pw_B$ would be as follows:

Therefore, the $pw_{B4}$ for the fourth path would be:

$$pw_{B4} = 1.$$ 

Then the $LF$ for site $B$ would be equal to

$$LF_B = pw_{B1} + pw_{B2} + pw_{B3} + pw_{B4} = 1\frac{1}{2} + 1 + 1\frac{1}{2} + 1 = 5.$$ 

The overall graph is as below

Also, based on the server log it is known that the user spent 10 s in A, 240 s in B and 600 s in C. Then the $TF$ for each of the web pages becomes

$$TF(A) = \frac{10}{850} = 0.012$$

$$TF(B) = \frac{240}{850} = 0.282$$

$$TF(C) = \frac{600}{850} = 0.706.$$ 

Therefore the preference factor for webpage B would be:

$$PF_B = LF_B \times TF_B$$

$$= 5 \times 0.282$$

$$= 1.41.$$ 

4 System design and implementation

4.1 System architecture

A web system has been developed using a three-tier architecture model. At the bottom layer an Apache Tomcat server is installed on a Windows environment. The bottom layer
also contains the storage system to store the users’ navigational log files. A simple interface allows administrators/owners to monitor the website’s performance and also provide various facilities for the cleaning process of the users’ log files. The middle layer contains the system logic and all other necessary software for providing the required communication to the other layers. This layer is also responsible for maintaining the security and other related rules for the communication between the client and server layers. The client layer, which is usually the web browser, interacts with the lower layers using the web communication software.

The users’ navigational information is saved in log files, and the process of analysing and filtering of the unwanted data in the log files are done at this stage using the proposed preference algorithm.

4.2 Programming design and implementation

The following pseudocode provides the general steps required to implement the preference-factor algorithm. The pseudocode is then implemented in Java programming languages, since Java provides features such as distributiveness, portability, simplicity, dynamic linking, GUI features that the other modern languages such as C++, Visual Basic, Perl or Python do not easily provide.

```
Begin
Create GUI layout
Show GUI
Instantiate class variables
Wait for action from user

If "Mine Logs" button pressed {
    Display dialog box to choose file
    Open Server Log File
    Read Log File
    Break each line into individual components
    ip address, Date & Time, Command, Web page, Status, Bytes,
      Name, Zone, Type, Reference, Agent
    Create new record with components
    Eliminate all the unnecessary and null items
    gif, jsp, css, .png, .ico, search type, .db
    Create a session based on unique ip addresses and times
    Store within each session the web pages (states) and clock time
    Create a linked list of each session
    Within each session create various paths taken by the user
    Determine the time in seconds for each state
    Determine the Likeness-factor by following the paths and
      determining if any states are visited again
    Determine the total time for the session
    Determine the Time-factor for each web page by dividing the time by the
      total time
    Determine the preference-function for each web page by multiplying the
      Likeness-factor and the Time-factor
    Determine the highest and second-highest preference factor and its web page
    Append to text area in GUI
}
end if

If "Web Site File" button pressed {
    Display dialog box to choose file
    Open web-site file
}
end if

If "Suggested Action" button pressed {
    Read web-site file
    Check to see if each line is equal to the web page with the
      highest or second-highest preference-function
    If equal then append text area with suggestions
    Close web-site file
}
end if

If "Exit" button pressed
    Exit program

End
```
5 Experimental procedure

5.1 Step 1: Server log analysis

As mentioned in Section 1, data preparation is one of three tasks and also the central task in web mining (Baglioni et al., 2003). In fact, this task can usually take up a large amount of time from the time required for the analysis task. Data preparation consists of two steps. The first step in this task is to filter out the unwanted requests. (These unwanted requests are usually image files, javascript files and style sheets, to name a few which are in the form of rows in the log file as shown below). The second step is to filter any unwanted information about the request, such as user identification, request method and query strings. The following are several lines from a server log file used for experimentation:

looking at the file, all but the following lines will be filtered out:

5.2 Step 2: A recommendation scenario

Ultimately, the owner of the website has to give guidelines regarding the correlations and importance of web pages. This information is provided by the owner in the action file. Figure 3 is a sample of an action file for the sample website created for this project. For example, in the first line the web page name is ‘phpdev\’ and the action to be taken is ‘php books’.
Preference-Function Algorithm: a novel approach

5.3 An example of PFA

For the purpose of easy interaction between the PFA program and the website administrator, a Graphical User Interface (GUI) has been developed. As shown in Figure 4, this screen contains four action buttons that a website administrator can select.

Figure 4 The required actions

When the ‘Mine Logs’ button is pressed, a dialogue box appears for the user to choose among server’s log files as shown in Figure 5.

Figure 5 Selection of server’s log file
Once the file has been selected, the results of the mining are shown on the screen as depicted in Figure 6. (The text area in Figure 7 shows the results of the mining steps and the selected server log file.)

**Figure 6**  GUI after mine logs button is pressed

![GUI after mine logs button is pressed](image)

Next, a website action file has to be selected. This is done by pressing the ‘Website File’ button. When this button is pressed, a dialog box appears for selecting the file to use for actions as shown in Figure 7.

**Figure 7**  Selection of appropriate action files

![Selection of appropriate action files](image)

Figure 8 shows the GUI after the ‘Website File’ button is pressed. The circled area in this figure shows the file that has been selected.

Next, when the ‘Suggested Action’ button is pressed, the content of the selected file will be shown, which presents the suggested actions for the user preferred webpages. (The circled text area in Figure 9 shows the recommended actions to the user. These recommended actions are chosen from the actions to be taken in lines 14, 19 and 25 in Figure 3.)
Figure 8  GUI after website file button is pressed

Figure 9  GUI after suggested action button is pressed

6 Conclusions

In planning for construction of web applications, information about the users’ preferences makes the design of web pages more relevant and useful. This paper implements a novel approach, defined as the PFA, for identifying the preferred websites for the users. The algorithm extracts future preferences from the users’ past web navigational activities. The server web logs are used as the basis for the navigational information to formulate the PF, and the finite state graph is used to model the users’ navigational patterns. The states in this graph are used to model each web page or session, and the navigations among the various states are determined to be the path for a particular user. The PFA is implemented based on two different factors that make up the PF: LF and TF. The LF of a web page is computed by determining the importance of the web page in
earlier navigational activities. The TF adds the time into the equation by using a formula for the overall time spent in the particular web page.

The viability of the PFA is shown with a user-created website and the automatic creation of the server logs. The overall process is divided into two steps. In step 1, the logs are mined and a PF is determined for this website. In step 2, the collected information from step 1, along with a constructed website scenario, is used to make the best possible recommendation for design to the owners of the websites.

References


Websites
Analog http://www.statslab.cam.ac.uk/~sret1/analog.
http://www.cs.uu.nl/docs/vakken/e-com/personalisation.pdf

Appendix A: OLAP

Web Server Statistics for [Deutscher Bildungsserver März 2002]
Analyzed requests from Fri-01-Mar-2002 00:00 to Sun-31-Mar-2002 23:59 (31.00 days).

This report contains overall statistics.
Successful requests: 744,864
Average successful requests per day: 24,028
Successful requests for pages: 677,439
Average successful requests for pages per day: 21,853
Distinct files requested: 59,668
Distinct hosts served: 93,908
Corrupt logfile lines: 151
Unwanted logfile entries: 73,437
Data transferred: 12,698 Gbytes
Average data transferred per day: 419,471 Mbytes
Appendix B: Example log files

Extended common log format

sample.txt

1Cust216.int1.santa-monica.ca.da.uu.net - - [08/May/1999:12:11:54 -0700] "GET / HTTP/1.0" 200 13516
"http://dir.yahoo.com/Science/Physics/High_Energy_and_Particle_Physics/Research/Accelerators/Stanford_Linear_Accelerator_Center_SLAC/" "Mozilla/3.01-C-MACOS8 (Macintosh; I; PPC)" GET / - "HTTP/1.0"
1Cust216.int1.santa-monica.ca.da.uu.net - - [08/May/1999:12:11:55 -0700] "GET /welcome/images/blank.gif HTTP/1.0" 200 812
"http://www.slac.stanford.edu/" "Mozilla/3.01-C-MACOS8 (Macintosh; I; PPC)" GET /welcome/images/blank.gif - "HTTP/1.0"
1Cust216.int1.santa-monica.ca.da.uu.net - - [08/May/1999:12:11:55 -0700] "GET /welcome/images/new-welcome.gif HTTP/1.0" 200 8032
"http://www.slac.stanford.edu/" "Mozilla/3.01-C-MACOS8 (Macintosh; I; PPC)" GET /welcome/images/new-welcome.gif - "HTTP/1.0"
1Cust216.int1.santa-monica.ca.da.uu.net - - [08/May/1999:12:11:55 -0700] "GET /welcome/images/welcome-border.gif HTTP/1.0" 200 4844
"http://www.slac.stanford.edu/" "Mozilla/3.01-C-MACOS8 (Macintosh; I; PPC)" GET /welcome/images/welcome-border.gif - "HTTP/1.0"
1Cust216.int1.santa-monica.ca.da.uu.net - - [08/May/1999:12:11:59 -0700] "GET /welcome/images/welcome-p.gif HTTP/1.0" 200 709
"http://www.slac.stanford.edu/" "Mozilla/3.01-C-MACOS8 (Macintosh; I; PPC)" GET /welcome/images/welcome-p.gif - "HTTP/1.0"
1Cust216.int1.santa-monica.ca.da.uu.net - - [08/May/1999:12:12:01 -0700] "GET /welcome/images/welcome-s.gif HTTP/1.0" 200 782
"http://www.slac.stanford.edu/" "Mozilla/3.01-C-MACOS8 (Macintosh; I; PPC)" GET /welcome/images/welcome-s.gif - "HTTP/1.0"
1Cust216.int1.santa-monica.ca.da.uu.net - - [08/May/1999:12:12:02 -0700] "GET /welcome/images/welcome-comp.gif HTTP/1.0" 200 532
"http://www.slac.stanford.edu/" "Mozilla/3.01-C-MACOS8 (Macintosh; I; PPC)" GET /welcome/images/welcome-comp.gif - "HTTP/1.0"
1Cust216.int1.santa-monica.ca.da.uu.net - - [08/May/1999:12:12:02 -0700] "GET /welcome/images/welcome-org.gif HTTP/1.0" 200 563
"http://www.slac.stanford.edu/" "Mozilla/3.01-C-MACOS8 (Macintosh; I; PPC)" GET /welcome/images/welcome-org.gif - "HTTP/1.0"
1Cust216.int1.santa-monica.ca.da.uu.net - - [08/May/1999:12:12:04 -0700] "GET /welcome/rotate/slacwest.jpg HTTP/1.0" 200 21646
"http://www.slac.stanford.edu/" "Mozilla/3.01-C-MACOS8 (Macintosh; I; PPC)" GET /welcome/rotate/slacwest.jpg - "HTTP/1.0"