Volume 5, Issue 3, March 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Advanced Data Preprocessing and Soft Computing Based Web Usage Pattern Discovery

Kadijath Thahira, Zahid Ansari
Dept. of Computer Science &Engineering
P.A. College of Engineering, Mangalore, India

Abstract— Due to the drastic raise in the field World Wide Web, the size of the file is increasing day by day. The process of discovering user's navigational behaviour from a web log file is known as Web Usage Mining (WUM). Three main stages in WUM are preprocessing, pattern discovery, and pattern analysis. In this paper, Preprocessing of the log file and clustering in order to identify the similar interests among the user's is discussed. In the pre-processing stage the click stream data is cleaned and in cleaning stage Web Robot generated requests are removed from the log file using various methods which are discussed in this paper. These cleaned log file is partitioned into a set of user profiles representing the activities of each user during different visits to the site. Each of these independent profiles consist a set of URLs representing a user session. In addition with TOH1and TOH2, this paper describes another Time-Oriented Heuristics (TOH) algorithm Hybrid TOH to identify the user sessions. Fuzzy S- shaped Weight Assignment technique is applied to filter out low support sessions. Then grouping of users based on their similar interests is done using the Fuzzy C-means clustering algorithm. Here, for the Preprocessing task the log file is taken from vtulfe.com.

Keywords— Web Usage Mining, Preprocessing, web robot removal, Time Oriented Heuristics, Fuzzy S-shaped weight assignment, Fuzzy C-means clustering.

I. INTRODUCTION

Data mining also known as knowledge discovery is the process of analyzing and exploring the large amounts of data and summarizing it into useful information. In order to discover the usage patterns that can be used to analyze the user's navigational behaviour, data mining techniques are applied to web usage log archives (server logs). This process is known as Web Usage Mining (WUM). The identity or origin of Web users along with their browsing behaviour at a Web site is stored in a web log. This web log is further processed to analyze the user's behaviour in order to understand and better serve the needs of Web-based applications.

The overall web usage mining process is divided into 3 interdependent stages: i) Data collection and pre-processing, ii) Pattern discovery and iii) Pattern analysis. In the pre-processing stage the click stream data is cleaned and partitioned into a set of user profiles representing the activities of each user during different visits to the site. In the pattern discovery stage, statistical, database, and machine learning operations are performed to obtain hidden patterns reflecting the typical behaviour of users as well as summary statistics on web resources sessions and users. In the final stage of the process, which is in pattern analysis, the discovered patterns and statistics are further processed, filtered, possibly resulting in aggregate user models.

In the preprocessing stage of Web Usage Mining the raw click stream data is transformed into a set of user profiles. Each such user profile captures a sequence or a set of URLs representing a user session. Web Usage Data preprocessing exploit a variety of algorithms and heuristic techniques for various preprocessing tasks such as data fusion and cleaning, user and session identification etc. In Data Cleaning phase of Data Preprocessing, the extraneous references to embedded objects, such as style files, graphics, or sound files are removed. In addition to these, Web Robot generated requests are removed.

Web Robots are software programs or agents that automatically traverse the hyperlink structure of the WWW in order to locate and retrieve information [1]. Web Robots are also known as Web Wanderers, Crawlers, or Spiders. Although the typical job of Web Robots are generally simple and structurally repetitive [2], some of the Web Robots may be harmful to the web. It wastes resources, misleads people and can trick search engines algorithms to gain unfair search result rankings [3]. So, it is required to identify the Web Robot requests and separate them from other users.

User Identification is the process of identifying unique users from the log file. User Session Identification is the process of dividing the user activity log of each user into sessions, where each session represents a single visit to the site. Identification of sessions is not an easy task, because of proxy servers, dynamic addresses, and cases where multiple users access the same computer or one user might use the multiple browsers or computers [4].

Sessionised data may contain thousands of sessions and each session may consist of thousands of URLs. Dimensionality reduction is accomplished by eliminating the low support URLs. But there might be small sessions with only little number of URLs. In order to filter out the noise from the data we should remove very small sessions. But direct elimination of low support URLs and small sized sessions may result in loss of significant amount of information especially when the count of low support URLs and small sessions is large [5].

Fuzzy S-shaped function is used to deal with this problem. Weights are assigned to the User sessions based on fuzzy S-shaped membership function.

Clustering is applied to this sessionized data in order to capture similar interests and trends among user's navigational patterns. After assigning the weights we apply a "Fuzzy C-means" clustering algorithm to discover the clusters of user profiles. In Fuzzy c-Means clustering algorithm, each data point may belong to more than one group (cluster), where the degree of membership for each data point is given by a probability distribution over the clusters.

II. LITERATURE REVIEW

Tanasa and Trousse, [4] have described the data preprocessing steps for Inter site web usage mining. For inter site web usage mining first they joined the different log files from various web sites generally belonging to the same organization. During cleaning step they removed all the requests concerning non-analyzed resources such as image and multimedia files and also all the request made by web robots (WR). In order to identify WR hosts they used following heuristics: i) look for all hosts that have requested the page "robots.txt". ii) Use a list of user agents known as robots. iii) Guess whether a host is web robot by Computing the browsing speed, if the browsing speed and the number of pages visited during the current visit exceed some specified threshold, then the host is considered as web robot. Once all the web robots are identified, requests generated by them are removed.

Spiliopouloe et al. [6] describes a proactive strategy for "session identification" for the websites without authentication information. Web server software is modified to associate a unique identifier to each client process accessing the server. Web log corresponding to each request made by the user's client to the server contains this identifier, which enables the unique assignment of requests to the users during one visit. Web sites without user authentication information or embedded session ids mostly rely on heuristics methods for sessionization. The sessionization heuristic helps in extracting the actual sequence of actions performed by one user during one visit to the site.

Generally, sessionization heuristics are either time-oriented or structure-oriented [7]. A formal framework of measuring the effectiveness of such heuristics has been proposed in [6]. The impact of different heuristics on various web usage mining tasks has been analyzed in [8]. Time-oriented heuristics (TOH) use an upper bound on the time spent in the entire site during a visit or an upper bound on page stay time [6]. Many existing tools use 30 minutes as a default timeout [7] [9]. During a visit, the time spent on any single page by a user varies within certain limits. If a long time elapses between two consecutive requests, it is likely that the second request is the first of new visit. This type of time-oriented heuristics is called as *TOH1*. A second type of time-oriented heuristics *TOH2* uses a threshold on the total page stay time. Priyanka Patel, Mitixa Parmar [10] proposed a method where session timeout period is dynamically adjusted. Here the session timeout period is the average time spent by a user on pages.

Robert Cooley, Bamshad Mobasher, and Jaidep Srivastava [7] described methods for user identification, "sessionizing", page view identification, path completion, and episode identification. Bettina Berendt and her colleagues [9] compared time-based and referrer based heuristics for visit construction. They found that heuristics appropriateness depends on the web site's design and on the visit's length.

Robert et al. [11] used a new concept for better session identification called as integer programming. This method generates session simultaneously and produced session better match to an empirical distribution.

Pang-Ning Tan and V. Kumar proposed an approach that uses the navigational patterns in the click-stream data to determine if it is due t a robot. Athena Stassopoulou and Marios D. Dikaiakos [2] introduced a probabilistic-reasoning approach to detect Web robots (crawlers) from human visitors of Web sites. They used a Naive Bayes network to classify the HTTP sessions of a Web-server access log as crawler or human induced.

After sessionization, there might be low support URLs or very small sessions, which need to be removed. Direct removal may result in loss of important information. So, Zahid Ansari, M.F.Azeem, A. V. Babu and W. Ahmed [5] proposed a fuzzy set teoritic approach for feature evaluation and dimensionality reduction. Here they assign weights to the sessions using linear fuzzy membership function based on the number of URLs accessed by the sessions.

Mobasheret. al. [12][13] have used user-based clustering as well as item-based clustering in a personalization framework based on Web usage mining.

III. METHODOLOGY

Web Usage Mining is the process of identifying the user access patterns from Web server logs. Web Usage Mining performs three main steps: Preprocessing, Pattern Recognition and Pattern Analysis. Fig. 1 describes Web Usage Mining process.

A. Web Log Data Preprocessing

Due to varied nature of log data, preprocesssing step is considered as most complicated and time consuming process. In order to boost up the performance and scalability of basic data mining techniques applied on log data it is very much essential to perform preprocessing of log data. The Primary tasks involved in data Preprocessing Data Cleaning, User Identification, Session Identification, which are discussed below.

1) Data Cleaning: When a user requests for a particular page it results in several log entries since graphics and scripts are down loaded in addition to HTML file. Since the main goal of Web Usage Mining is to get the picture of users behaviour, It is not necessary to include file requests that the user did not explicitly. The first step in Data Cleaning process is the elimination of the irrelevant items. It is accomplished by checking the suffix of the URL. All log entries with the suffix of the URL such as, gif, jpeg, jpg, GIF, JPEG, and JPG are removed. Default list of suffixes were used to remove undesired files. The second step is to remove web robot (web spiders, web crawlers) generated log entries. These web robot generated request is removed by checking , i) URL of the request, ii) IP address of the request, iii) User Agents of the request and iv) Head request method and referrer field of the request.

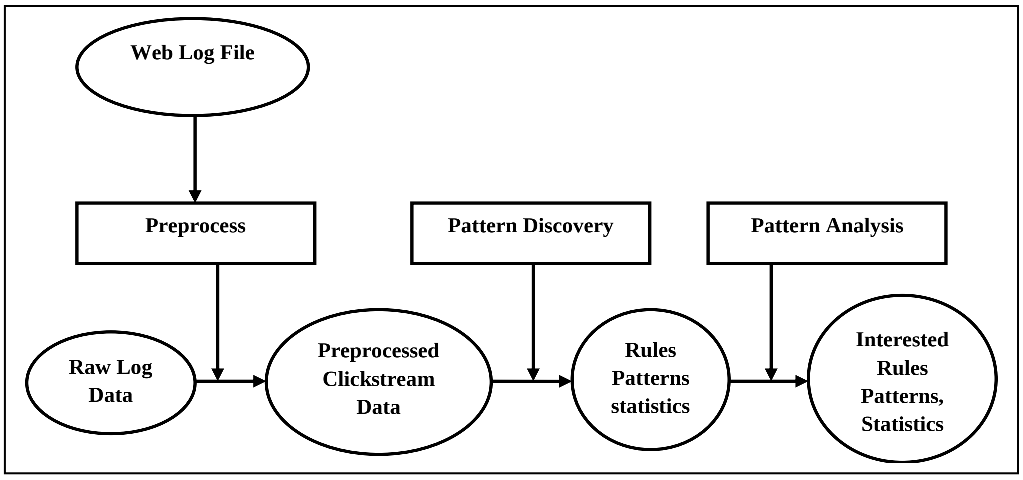
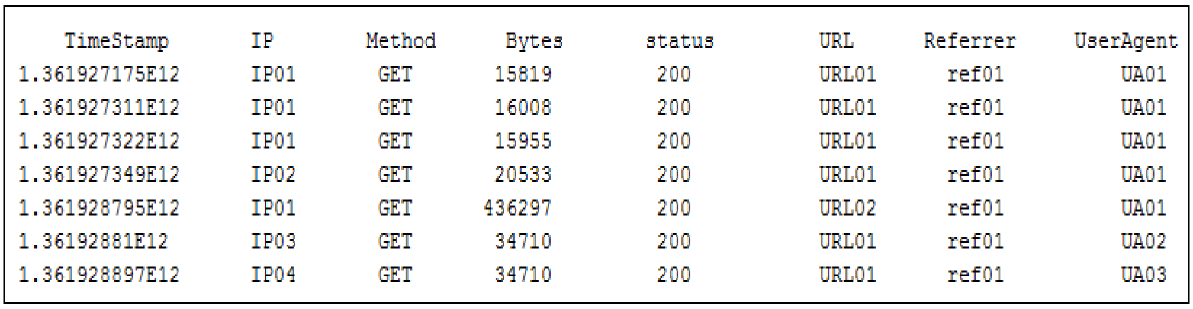

Web Log File
Preprocess
Pattern Discovery
Pattern Analysis
Raw Log Data
Preprocessed Clickstream Data
Rules Patterns statistics
Interested Rules Patterns, Statistics

Fig. 1 Web Usage Mining Process

If the URL of the request matches "robots.txt" then that request is confirmed as web robot generated request. If the IP address of the users request matches a set of known IP address of the robots, then they are identified as Probable Robot generated request. Under the proposed ethical guidelines for robot designers, a supportive robots must claim its identity to a web server via its User Agent field. So User Agent of the users request is checked against a set of known User Agents of the robots. If match is found, then that request is identified as probable robot generated request. Then the HTTP request method and request with unassigned referrer filed is checked. If the request method is "HEAD" and referrer field matches "-" then that request is identified as probable robot generated request. According to ethical guidelines provided to robot designers, they should use HEAD request method. The referrer field contains the address of the web page that contains the link that the user followed to reach the current requested page. Referrer field value appears as "-" if the request is from Web Robot, because most robots do not assign any value to their referrer field. But sometimes, any value may not be assigned to referrer field if the user follows direct link. So we should use this method in addition to some other method.

Since the IP addresses dynamically changes and the IP addresses used by robot may also be used by a standard user, it is difficult to confirm a request as web robot generated request. And also because all robot designers do not follow the guidelines and they attempt to hide their identities by using the same User Agent as the standard web browsers, Robot detection becomes more complicated. In this case, it is impossible to detect these robots using this User Agent fields alone. So instead of identifying the robot generated request using a single method, method ii, iii, iv is combined together. That is, if a request is identified as robot generated request by IP address, then that request should be identified as Web robot by User Agent check, HTTP Request method and referrer field check. Then that request will be confirmed as web robot generated request. Then the result of robots.txt method and method ii, iii, iv is combined together. Then these requests are removed from the log file. Finally the status field of the request is checked. If status of the request is between 300 to 499 then that particular request is removed from log file. Because status between 300 - 499 indicates the unsuccessful requests, we should remove them. The algorithm for Data Cleaning is shown in Fig. 3.

Let L= $\{e_1, e_2, \dots\dots e_n\}$ be the web server log containing several log entries. Let U = $\{url_1, url_2, \dots\dots url_n\}$ be the set of URLs corresponding to all the requested Web Resources of a Web site. Let IP = $\{ip_1, ip_2, \dots\dots ip_{nip}\}$ be the set of IP addresses of all the client machines accessing that Web site. Let C be the output file containing cleaned log entries.



TimeStamp	IP	Method	Bytes	status	URL	Referrer	UserAgent
1.361927175E12	IP01	GET	15819	200	URL01	ref01	UA01
1.361927311E12	IP01	GET	16008	200	URL01	ref01	UA01
1.361927322E12	IP01	GET	15955	200	URL01	ref01	UA01
1.361927349E12	IP02	GET	20533	200	URL01	ref01	UA01
1.361928795E12	IP01	GET	436297	200	URL02	ref01	UA01
1.36192881E12	IP03	GET	34710	200	URL01	ref01	UA02
1.361928897E12	IP04	GET	34710	200	URL01	ref01	UA03

Fig. 2 Output After removing Graphical contents and robot generated requests

Fig. 2 shows the output file C which is generated as a result of cleaning and transformation. In order to hide the user's identity client IP addresses are replaced with aliases as shown in above figure. The URL, referrer and user agent field of the output shows that URL, referrer and User agent strings are replaced by some name in order to enhance further processing. A map is maintained with URL, referrer, and User Agent string and their corresponding names.

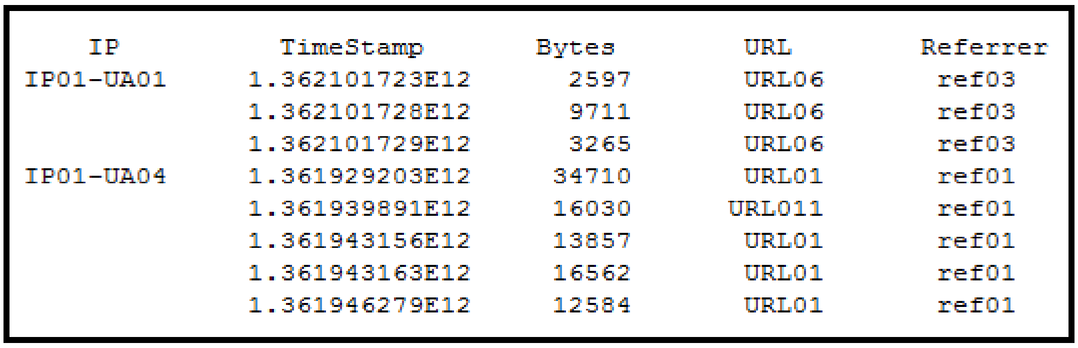
```
Algorithm: Data Cleaning
Input: Web Log File L
Output:  Cleaned Log File C
For each entry E € L
Step 1: Split E and extract various fields
Step 2: If the URL contains query string
                    Then Remove the query string
                     Step 3: If URL.suffix matches .gif, .jpeg, .jpg, .css, .GIF, .JPEG, .JPG
                    Then Remove that entry from the log file
Step 4: If the log file contains Web Robot generated request        //discussed in figure 3.6
                    Then remove that entry from the log file
Step 5: if Status filed of the request matches(300-499)
          Then remove that entry from the log file
Step 6: Repeat the steps 2-4 until eof
Step 7: Hide the user's identity by encrypting the IP address
Step 8: Store the URL in a URL map with corresponding URL number
Step 9: Store the User Agent in User Agent map with corresponding User Agent number
Step 10: Store the Referrer in a Referrer map with corresponding Referrer number
Step 11: Print all the required fields into the output file C
```

Fig. 3 for Data Cleaning and Transformation

2) User Identification: After Cleaning the Web log file, User Identification is the next step in Data Preprocessing. Due to existence of local/external proxy servers, cache systems, cooperate firewalls and shared internet, User Identification becomes complicated [7, 14]. Since the log files of web server we are working on do not contain the user login information, pair of unique IP address and User Agent is taken as the user. For identification of users we assign different user id to the pair of unique IP address and user agent. If the IP address of a user is same as previous entry and user agent is different, then the user is assumed as a new user. Fig. 4 shows the algorithm for separating out the requests for each individual user. Output of the User Identification is shown in Fig. 5.

```
Algorithm: User Identification
Input: Cleaned Log File C
Output: File U that contains the set of user with the URLs accessed by them.
For each entry E € C
Step 1: Split E to get required fields
Step 2: Compare the IP addresses of two consecutive entries
Step 3: Take the combination of unique IP address and User Agent as the user
Step 4: If IP address is same for two consecutive entries
Step 5: Then check the User Agent field of those entries
Step 6: If both IP address and User Agent is same for two consecutive entries
Step 7: Then consider the two consecutive requests as belonging to same user
Step 8: Else consider the two consecutive as belonging to the two different users
Step 9: Repeat the steps 2-8 until eof
Step 10: Print the output of Step 7 and Step 8 to the output file U
```

Fig. 4 Algorithm to separate requests for each individual user



```
    IP            TimeStamp        Bytes       URL       Referrer
IP01-UA01      1.362101723E12      2597       URL06        ref03
               1.362101728E12      9711       URL06        ref03
               1.362101729E12      3265       URL06        ref03
IP01-UA04      1.361929203E12     34710       URL01        ref01
               1.361939891E12     16030      URL011        ref01
               1.361943156E12     13857       URL01        ref01
               1.361943163E12     16562       URL01        ref01
               1.361946279E12     12584       URL01        ref01
```

Fig. 5 Output of User Identification

3) User Session Identification: After the Identification of users, there is a need to identify sessions. Session is set of requests done to a particular web site by single user for prescribed duration. Usually sessionization is done using Heuristics methods for the web sites without user authentification information or embedded session ids. The sessionization heuristic helps in extracting the actual sequence of actions performed by one user during one visit to the site. Sessionization heuristics are usually divided into either time-oriented or structure-oriented. Time Oriented Heuristics(TOH) are used for identification of sessions. Different TOH methods such as TOH1, TOH2 and Hybrid TOH are used.

- ***TOH_1*:** The time duration of a session must not exceed a threshold α. For instance, let timestamp of the first request in a session is T_1. A request with timestamp T_i is assigned to this session if and only if $T_i- T_1\le\alpha$. If the timestamp of the request is larger than $T_1+\alpha$, then that request is considered as the first request of the next session.
- ***TOH_2*:** The time spent on a page visit must not exceed a threshold β. For instance, let T_i be the timestamp of the request most recently assigned to a session. The next request with timestamp T_{i+1} belongs to same session if and only if $T_{i+1}-T_i\le\beta$. Otherwise, current request is considered to be the first request of the next session.
- ***Hybrid TOH*:** This method is the combination of *TOH_1* and *TOH_2*. Here, the time duration of a session must not exceed a threshold α and also the time spent on a page visit must not exceed a threshold β. For instance, let T_1 be the timestamp of the first request in a session. Let T_i be the timestamp of the request most recently assigned to a session. Then next request with timestamp T_{i+1} belongs to the same session if and only if $T_{i+1} -T_1\le \alpha$ &&$T_{i+1}-T_i\le\beta$. Otherwise that request is considered as the first request of the next session.

```
Algorithm: User Session Identification using TOH1
Input: File U containing access logs of various users
Output: File S that contains the different user sessions
For each entry E € U
Step1: Split and extract various fields
Step 2: If user field is not empty
            UserId = E.user
            Output E to S
            $T_1$=E.time
Step 3: Else
            $T_2$=E.time
Step 4: If $T_2 - T_1\le\beta$
Step 5: Then                    // Compare the time stamps of current and first request
            Output E to S
Step 6: Else
            Output UserId to file S with new session Id
            Output E to file S
            $T_1$=E.time
Step 7: Repeat the steps 2-6 until eof
```

Fig. 6 Algorithm to generate User Sessions Based on *TOH_1*

```
Algorithm: User Session Identification using TOH2
Input: File U containing access logs of various users
Output: File S that contains the different user sessions
For each entry E € U
Step1: Split and extract various fields
Step 2: If user field is not empty
            UserId = E.user
            Output E to S
            $T_1$=E.time
Step 3: Else
            $T_2$=E.time
Step 4: If $T_2 - T_1\le\beta$
Step 5: Then
            Output E to S
            $T_1$=E.time
Step 6: Else
            Output UserId to file S with new session Id
            Output E to file S
            $T_1$=E.time
Step 7: Repeat the steps 2-6 until eof
```

Fig. 7 Algorithm to generate User Sessions Based on *TOH_2*

Algorithm for session Identification using TOH_1 is described in Fig. 6. Algorithm for session Identification using TOH_2 is described in Figure 7.

In Time-oriented heuristic *TOH1*, an upper bound (threshold) α is applied on the time spent in the entire site during a visit. Here, the timestamp of every URL access request is compared with that of first access request of the current session. If the time difference is larger than the specified threshold, this request becomes the first request of the new session, otherwise it belong to the current session.

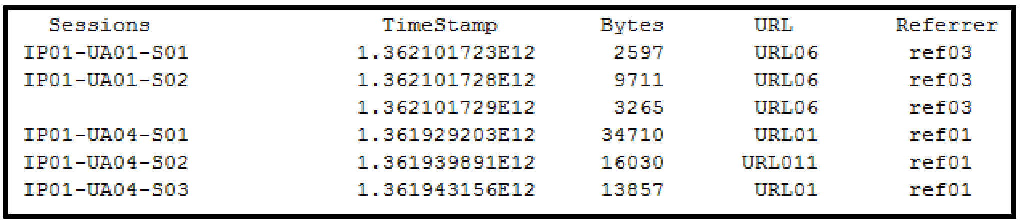
In second type of time-oriented heuristics *TOH2*, a threshold β is applied on the total page stay time. The timestamp of every URL access request is compared with that of the previous access request. If the time difference is larger than threshold, this request becomes the first request of the new session; otherwise it belongs to the current session. In *Hybrid TOH*, a threshold α is applied on the time spent in the entire site during a visit and also a threshold β is applied on the total page stay time. The timestamp of every URL access request is compared with that of first access request of the current session and also with that of the previous access request. If the time difference is larger than threshold, this request becomes the first request of the new session; otherwise it belongs to the current session. Here *TOH1* uses the threshold as one hour and *TOH2* uses threshold as 30 minutes. In *Hybrid TOH*, both threshold values are used.

Algorithm for session Identification using *Hybrid TOH* is described in Fig. 8.

```
Algorithm: User Session Identification using Hybrid TOH
Input: File U containing access logs of various users.
Output: File S, containing different sessions
For each line E € U
Step 1: Split and extract various fields
Step 2 : If user field is not empty
            UserId =E.user
            Output E to S
            $T_1$=E.time
            $T_i$ =E.time
Step 3: Else
            $T_2$=E.time
Step 4: If $T_2 - T_1 \leq \alpha$ and $T_2 - T_i \leq \beta$
Step 5: Then
            Output E to S
            $T_1 = T_i$=E.time
Step 6: Else
            Output UserId to file S with new session Id
            Output E to file S
            $T_1 = T_i$= E.time
Step 7: Repeat the steps 2-6 until eof
```

Fig. 8 Algorithm to generate User Sessions Based on *Hybrid TOH*

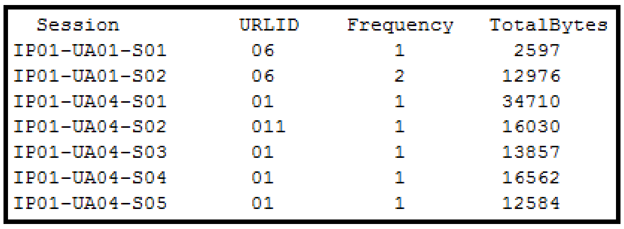
Fig. 9 shows the output file S containing user sessions.



Sessions	TimeStamp	Bytes	URL	Referrer
IP01-UA01-S01	1.362101723E12	2597	URL06	ref03
IP01-UA01-S02	1.362101728E12	9711	URL06	ref03
	1.362101729E12	3265	URL06	ref03
IP01-UA04-S01	1.361929203E12	34710	URL01	ref01
IP01-UA04-S02	1.361939891E12	16030	URL011	ref01
IP01-UA04-S03	1.361943156E12	13857	URL01	ref01

Fig. 9 Sample of output file after sessionization using TOH

After session Identification, we need to remove duplicate URLs from each session. Once user sessions are generated we scan each session and remove the duplicate URLs from each session. Within a session, for each unique URL a single copy of the URL is kept along with its frequency of occurrence. Also count of total number unique URLs in each session is kept. Output File After removing the duplicate URLs is shown in Fig. 10.



Session	URLID	Frequency	TotalBytes
IP01-UA01-S01	06	1	2597
IP01-UA01-S02	06	2	12976
IP01-UA04-S01	01	1	34710
IP01-UA04-S02	011	1	16030
IP01-UA04-S03	01	1	13857
IP01-UA04-S04	01	1	16562
IP01-UA04-S05	01	1	12584

Fig. 10 Output after removing duplicate URLs

B. Dimensionality Reduction by Fuzzy Weight Assignment

Before Pattern Discovery phase, very small sessions from session files can be removed in order to eliminate Noise [7]. After session identification, since the sessionized data may contain thousands of user sessions and each user session may consist of hundreds of URL accesses, dimensionality reduction is achieved by eliminating the low support URLs. But direct removal of low support URLs may result in loss of significant information. In this project fuzzy S-shaped membership function is used to deal with this problem. Here, Instead of removing small sessions directly weight is assigned to all sessions using "Fuzzy S-shaped membership function" based on number of URLs accessed by the sessions.

Let x be the number of URLs accessed in a session then Fuzzy S-shaped membership function is given as follows:

$$S(x,a,b,c) = \begin{cases} 2\left(\frac{x-a}{c-a}\right)^2 & \text{if } a < x \le b \\ 1 - 2\left(\frac{x-a}{c-a}\right)^2 & \text{if } b < x \le c \\ 1 & \text{if } x > c \\ 0 & \text{if } x \le a \end{cases}$$

With $b = \frac{a+c}{2}$

a -> lower threshold on session support count

c -> upper threshold on session support count

C. Web Usage Pattern Recognition

Recognition of patterns and uniformity in data is known as Pattern recognition. Pattern Recognition can be supervised learning or unsupervised learning. Pattern Recognition with supervised learning is called classification and pattern recognition with unsupervised earning is known as clustering. Here clustering algorithm is used. Grouping set of physical or abstract objects into classes of similar objects are known as clustering. Clustering divides the given data set into groups or clusters such that the inter cluster similarities are minimized while intra cluster similarities are maximized. Here Fuzzy C-Means clustering algorithm is used to find the similar interests among the users.

In Fuzzy c-Means clustering algorithm, each data point may belong to more than one group (cluster), where the degree of membership for each data point is given by a probability distribution over the clusters.

The Fuzzy C-means (FCM) algorithm is similar to K-means algorithm but the difference is that, in K-means a firm decision about which data point belongs to which cluster is made. In fuzzy C-Means, a value between 0 and 1is assigned indicating the probability that a data point belong to particular cluster. Fuzzy rule states that the sum of the membership value of a data point to all clusters must be 1. If the membership value of a data point to a particular cluster is high then more likely that data point belongs to that cluster. This algorithm uses a fuzziness index parameter m€[1,∞] which determines the degree of fuzziness in the clusters. As the value reaches to 1 the algorithm works like a crisp partitioning algorithm. As the value of m increases, probability of data point belonging to multiple clusters increases.

Let X = $\{x_1, x_2, . . ., x_n\}$be the set of n data points. Here each session is considered as one data point. Let V_j = $\{v_1, v_2, . . ., v_n\}$ be the set c cluster centres. Let U_{ij} represent the grade of membership of data point xi in cluster j. The performance index J(U, V, X) of FCM clustering can be specified as the weighted sum of distances between the data point and the corresponding cluster centres. The equation for it is given as:

$$J(U, V, X) = \sum_{j=1}^{c} \sum_{i=1}^{n} u_{ij}^{q}\, d_{ij}^{2}(x_i, v_i) \qquad (1)$$

Where $d_{ij}^2(x_i, v_j) = \sum_{k=1}^{n} w(x_i) \left\| x_k^i - v_k^i \right\|$ is the distance between x_i and v_j. Performance Index is minimized by updating grade of memberships of data points and cluster centers until certain condition is met. In each of the iterations, cluster centres are updated using the following formula:

$$\overline{v_j} = \frac{\sum_{i=1}^{m} u_{ij}^{q} x_i}{\sum_{i=1}^{m} u_{ij}^{q}} \qquad (2)$$

Membership matrix is calculated using the following formula:

$$U_{ij} = \frac{1}{\sum_{l=1}^{c} \left(\frac{|x_i - v_k|}{|x_i - v_l|}\right)^{\frac{2}{m-1}}} \qquad (3)$$

Output of this clustering algorithm is the group of user sessions that are similar co-occurrence patterns of URL references. After performing clustering on user sessions a cluster of user sessions are formed. Here each cluster represents group users with similar navigational behaviour.

Algorithm for FCM is shown in Fig. 11.

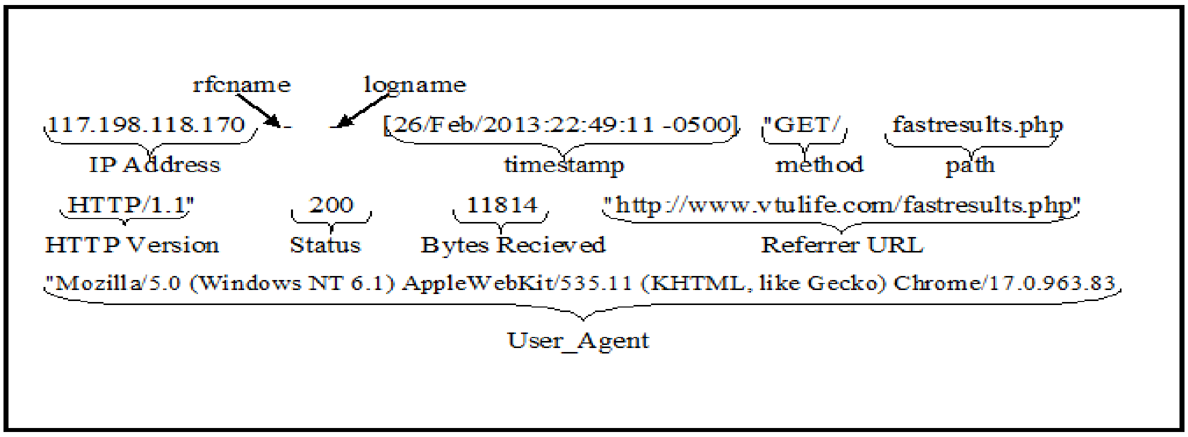
```
Algorithm: Fuzzy C-means Clustering Algorithm
Input: c-no of clusters, error threshold ϵ, fuzziness index m, maximum iterations and set of user sessions S as data points
Output: Set of c cluster centers and membership matrix M
Step 1: Initialize c cluster centers by randomly selecting c user sessions from S
Step 2: iteration=0
Step 3: repeat
        For i =1 to no of data points
                For j =1 to no of cluster
                        Compute membership matrix $U_{ij}$ using (3)
                End For
        End For
Step 4: For j=1 to no of clusters
                Compute cluster center $\overline{v_j}$ by using (2)
        End For
Step 5: Compute objective function J(iteration) using (1)
Step 6: iteration=iteration+1;
Step 7: Repeate until (J(iteration)-J(iteration-1) < ϵ) || iteration =n
```

Fig. 11 Fuzzy C-means clustering algorithm

IV. RESULTS AND DISCUSSION

A. *Web Server Log File*

Server log file contains the entries which are recorded when a user requests for a particular page on web. This file can be accessed only by administrative person or server owners [20]. There are various formats of log files such as Common log format (CLF), IIS standard/extended log format, Combined/Extended common log format (ECLF), Log markup language (LogML), because of different setting parameters [20, 21]. Among them commonly used log format is CLF. CLF is a format with a fixed number of attributes which are suitable for http web sites. CLF includes user's IP address/hostname, rfcname, log name, date with time zone, page access method, PATH, http version, server response code and byte received [20, 22]. ECLF is a log format with additional attributes like referrer, user agent and cookies [20, 23]. Description of various attributes of ECLF is shown in Table 1and a single entry of ECLF is given in Figure 12. Here the log file is taken from the vtulife.com server.



```
117.198.118.170 - - [26/Feb/2013:22:49:11 -0500] "GET/ fastresults.php HTTP/1.1" 200 11814 "http://www.vtulife.com/fastresults.php" "Mozilla/5.0 (Windows NT 6.1) AppleWebKit/535.11 (KHTML, like Gecko) Chrome/17.0.963.83
```

Fig. 12 Sample of ECLF

Table I Attributes of Extended Log Format

Atrributes	Desription
IP Address/hostname	Address of the client
rfcname	Remote login name of the user. "-" indicates that field is empty
logname	Login name of the user. "-" indicates that the field is empty
Date with time zone	Returns date, time with time zone of user's request
Page access method	Mode of request (GET, POST, HEAD, PUT)
PATH	Path of the requested page
http version	Version of http protocol

Status code	http status code returned to client by the server(404-page not found)
Bytes received	Size of the file sent from the server
referrer	It is the URL of the web page from where the requested page is coming. "-" indicates the case of direct request
User_Agent	It contains user's browser name, its version and Operating system information. It is sent by web client to server.
cookies	It is a piece of information sent to visitors by web server to identify the details of particular user.

B. Results

Results after performing the pre-processing steps defined in section III.A are as follows.

Table III Result of preprocessing

Item	count
Initial number of log entries	138534
Number of log entries after removing irrelevant requests	53543
Number of robots generated requests	1430
Number of log entries after removing robot requests	52107
Number of log entries after removing unsuccessful requests	50480
No of IP addresses	1268
Number of URLs	264
Number of user agents	490
Threshold value for TOH1	30 mins
Threshold value for TOH2	60 mins

Result of The Fuzzy C-means clustering is as follows.
Here result of TOH1, TOH2, HybridTOH is used as the input for Fuzzy C-means clustering and then based on Sum of Squared error (SSE) for different number of clusters, different datasets are compared. SSE is calculated using the following formula:

SSE $(C_1, C_2, \ldots, C_k) = \sum_{j=1}^{k} \sum_{i=1}^{n} u_{ij}^{q}\, dist(x_i, c_j)^2$ (4)

From the Table III, we can see that the SSE values for Hybrid TOH are lesser than TOH1 and TOH2. SSE values for various number of clusters for each dataset is compared. So the value of SSE minimizes as number of clusters increases. Fig. 13 depicts the graph of SSE versus number of clusters by applying Fuzzy C-means clustering algorithm to the user sessions extracted through various time oriented heuristics namely TOH1, TOH2 and Hybrid TOH. From the figure it is clear that user sessions discovered by Hybrid TOH results in better cluster formation as compared with other two approaches by minimizing the clustering error.

Table IIIII Comparision of toh1, toh2, hybrid toh using sse values

Data Sets	No Of Clusters	SSE
TOH1	4	2448.5679
	8	2201.0417
	12	2116.5352
	16	2073.8604
	20	2048.2825
	24	2031.1082
TOH2	4	2358.5167
	8	2122.3259
	12	2057.1834
	16	2001.4529
	20	1973. 8749
	24	1953.3927

TOH3	4	2340.6514
	8	2104.4395
	12	2023.5669
	16	1983.0848
	20	1958.8405
	24	1942.7104

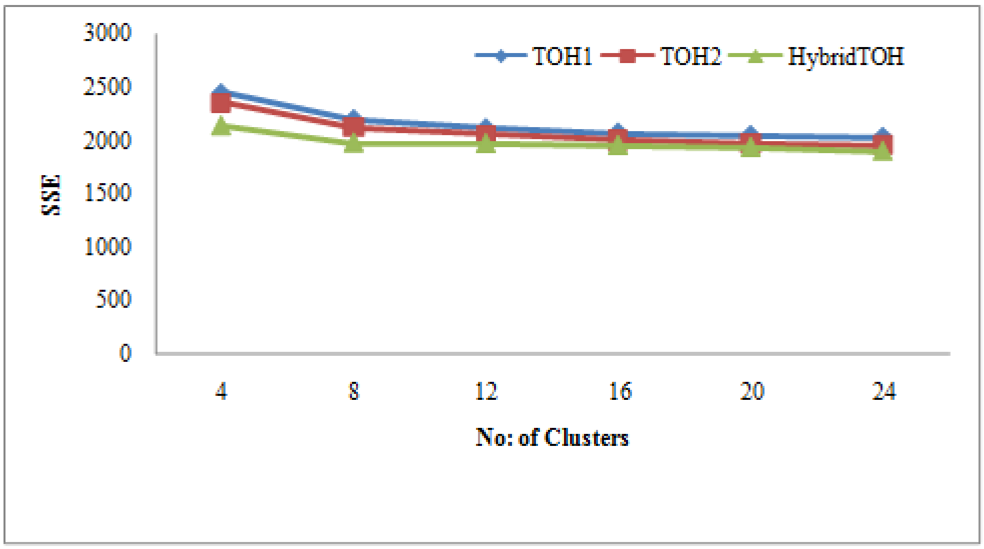


Fig. 13 Graph plot of No. Of clusters Vs SSE

V. CONCLUSIONS

In this paper we discussed methodology to pre-process the web log data including data cleaning, user identification and session identification. In data cleaning stage, the techniques to detect and remove robot generated request is discussed. In order to remove low support URLs, fuzzy S-shaped weight assignment technique is used. We analysed three time oriented heuristics *TOH1*, *TOH2* and *Hybrid TOH* using *Fuzzy C-means* clustering algorithm. Time oriented Heuristics TOH1, TOH2, Hybrid TOH are compared based on SSE values. Our experimental results show that the user sessions identified through the *Hybrid TOH* methodology results in better formation of user session clusters as compared with those identified using other two time oriented heuristics. And also we can conclude that Fuzzy C-means clustering works better for more number of clusters, because the SSE is minimized as the number of cluster increases. This improved performance is achieved by considering session duration time and page access time in the *Hybrid TOH* while identifying the user session clusters.

REFERENCES

[1] P.-N. Tan and V. Kumar, "Discovery of Web Robot Sessions Based on their Navigational Patterns," *Data Mining and Knowledge Discovery*, vol. 6, pp. 9-35, 2002.

[2] Athena Stassopoulou, Marios D. Dikaiakos" Web robot detection: A probabilistic reasoning approach", *ELSEVIER, Computer Networks*, Volume 53, Issue 3, 27 February 2009, Pages 265–278.

[3] Z. Gyongyi and H. Garcia-Molina, "Web spam taxonomy," in Proceedings of the 1st International Workshop on Adversarial Information Retrieval on the Web, Chiba, Japan, 2005.

[4] D. Tanasa and B. Trousse, "advanced data pre-processing for intersites web usage mining," *IEEE Intelligent Systems*, vol. 19, no. 2, pp. 59-65, 2004.

[5] Zahid Ansari, M.F.Azeem, A.V.Babu and W.Ahmed. "A Fuzzy Approach for Feature Evaluation and Dimensionality Reduction to improve the quality of Web usage mining results", *International Journal on Advanced Science Engineering and Information Technology*, pp.67-73 Vol.2 No. 6, 2012.

[6] M. Spiliopoulou, B. Mobashar, B. Berendt, and M. Nakagawa, "A framework for the evaluation of session reconstruction heuristics in web usage analysis," *INFORMS J. on Computing*, vol. 15, pp. 171-190, April 2003.

[7] R. Cooley, B. Mobasher, J. Srivastava et al. "Data preparation for mining world wide web browsing patterns," *Knowledge and Information Systems*, vol. 1, no. 1, pp. 5-32, 1999.

[8] B. Berendt, B. Mobasher, M.Nakagawa, and M. Spiliopoulou, "The impact of site structure and user environment on session reconstruction in web usage analysis," in WEBKDD 2002 – *Mining Web Data for Discovering Usage Patterns and Profiles*, ser. Lecture Notes in computer science. Springer Berlin/Heidelberg, 2003, vol.2703, pp. 159-179.

[9] B. Berendt and M. Spiliopoulou, "Analysis of navigation behavior in web sites integrating multiple information systems," *TheVLDB Journal*, vol. 9, pp. 56-75, 2000.

[10] Priyanka Patel, Mitixa Parmar, "Improve Heuristics for User Session Identification through Web Server Log in Web Usage Mining," *(IJCSIT) International Journal of Computer Science and Information Technologies*, Vol. 5 (3), 2014, 3562-3565.

[11] R. F. Dell (2008), "Web user session reconstruction using integer programming," *in International Conference on Web Intelligence and Intelligent Agent Technology*, IEEE/ACM/WIC, Vol. 1 Page(s): 385-388.

[12] B. Mobasher, H. Dai, T. Luo, and M. Nakagawa.Discovery and evaluation of aggregate usage profiles for web personalization. Data Mining and Knowledge Discovery, 6(1):61–82s, 2002.

[13] BamshadMobasher, Robert Cooley, and JaideepSrivastava. Automatic personalization based on web usage mining. Commun.ACM, 43:142– 15.

[14] Zidrina Pabarskaite, Aistis Raudys (2007), "A process of knowledge discovery from web usage data: Systemization and critical review," *in Journal of Intelligent Information System,* Springer Vol.28 Issue.1 Page(s): 79-104.