Generate Personalized usage knowledge from the web access behavior using Clustering Techniques

V. JAYAKUMAR[1] Dr.K.ALAGARSAMY[2]
[1]Assistant Professor in Computer Science, Ayya Nadar Janaki Ammal College, Sivakasi, India.
[2]Associate Professor in Computer Science, Madurai Kamarajar University, Madurai, India.

Abstract: The immense volume of web usage data that exists on web servers contains potentially valuable information about the behavior of website visitors. This information can be exploited in various ways, such as enhancing the effectiveness of websites or developing directed web marketing campaigns. In this paper we will focus on applying clustering algorithm as a data mining technique to extract potentially useful knowledge from web usage data. We conducted a comprehensive analysis of web usage mining techniques found on a website of an educational institution. Our experiments confirm that, prior to pruning, the set of generated clustering algorithm contained too many non-interesting rules, which made it very difficult for a user to find and exploit useful information. Many of these rules are a simple consequence of the high correlation between web pages due to their interconnectedness through the website link structure. We proposed and applied a set of basic clustering to reduce the rule set size and to remove a significant number of non-interesting data. The analysis of clustering algorithm in our case study confirmed the hypothesis that discovering interesting and potentially useful in web usage data that does not have to be a time consuming task and can lead to actions that increase the website's effectiveness.

Keywords: Clustering algorithm, web usage data, pruning, interestingness measures, website link structure

1. Introduction

Cluster analysis is the task of grouping a set of object in such a way that objects in the same group are more similar, based on some measure, to each other than to those in other groups. No preclassified training data is assumed to be available. Classification and cluster analysis are important techniques that partition objects that have many attributes(multidimensional data) into meaningful disjoint subgroups so that objects in each group are more similar to each other in the values of their attribute than they are to object in other groups.

1.1 Features of cluster analysis

- Scalability : Data mining problem can be large and therefore it is desirable that a cluster analysis method be able to deal with smalls well as large problems gracefully. Ideally, the performance should be linear with the size of the data. The method should also scale well to datasets in which the number of attributes is large.
- Only one scan of the dataset: For large problems, the data must be stored on the disk and the cost of I/O from the disk can then become significant in solving the problem. It is therefore desirable that a cluster analysis method not require more than one scan of disk resident data.
- Ability to stop and resume: When the dataset is very large, cluster analysis may require considerable processor time to complete the task. In such cases, it is desirable that the task be able to be stopped and then resumed when convenient.

- Minimal input parameters: The cluster analysis method should not expect too much guidance from the user. A data mining analyst may be working with a dataset about which his/her knowledge is limited. It is therefore desirable that the user not be expected to have domain knowledge of the data and not be expected to posses insight into clusters that might exist in the data.

- Robustness: Most data obtained from a variety of sources has errors. It is therefore desirable that a cluster analysis method be able to deal with noise, outliers and missing values gracefully.
- Ability to discover different cluster shapes: Clusters come in different shapes and not all clusters are spherical. It is therefore desirable that a cluster analysis method be able to discover cluster shapes other than spherical. Some applications require that various shapes be considered.
- Different data types: Many problems have a mixture of data types, ex: numerical, categorical and even textual. It is therefore desirable that a cluster analysis method be able to deal with not only numerical data but also Boolean and categorical data.
- Result independent of data input order: Although this is a simple requirement, not all methods satisfy it. It is therefore desirable that a cluster analysis method not be sensitive to data input order. Whatever the order, the result of cluster analysis of the same data should be the same.

1.2 Types of cluster analysis methods

- ✓ Partitional Methods: Obtain a single level partition of objects. These methods usually are based on greedy heuristics that are used iteratively to obtain a local optimum solution. Given n objects, these methods make k<=n clusters of data and use an iterative relocation method. It is assumed that each cluster has at least one object and each object belongs to only one cluster. Objects may be relocated between clusters as the clusters are refined. Often these methods require that the number of clusters be specified apriori and this number usually does not change during the processing.
- ✓ Hierarchical Methods: Obtain a partition of the objects resulting in a tree of clusters. These methods either start with one cluster and then split into smaller and smaller clusters(called divisive or top down) or start with each object in an individual cluster add then try to merge similar clusters into larger and larger clusters(called agglomerative or bottom up). In this approach, in contrast to partitioning, tentative clusters may be merged or split based on some criteria.
- ✓ Density-based Methods: In this class of methods, typically for each data pint in a cluster, at least a minimum number of points must exist within a given radius. Density-based methods can deal with arbitrary shape clusters since the major requirement of such methods is that each cluster be a dense region of points surrounded by regions of low density.
- ✓ Grid-based Methods: In this class of methods, the object space rather than the data is divided into a grid. Grid partitioning is based on characteristics of the data and such methods can deal with non-numeric data more easily. Grid-based methods are not affected by data ordering.
- ✓ Model-based Methods: A model is assumed, perhaps based on probability distribution. Essentially the algorithm tried to build clusters with a high level of similarity within them and a low level of similarity between them. Similarity measurement is based on the mean values and the algorithms tries to minimize the squared-error function.

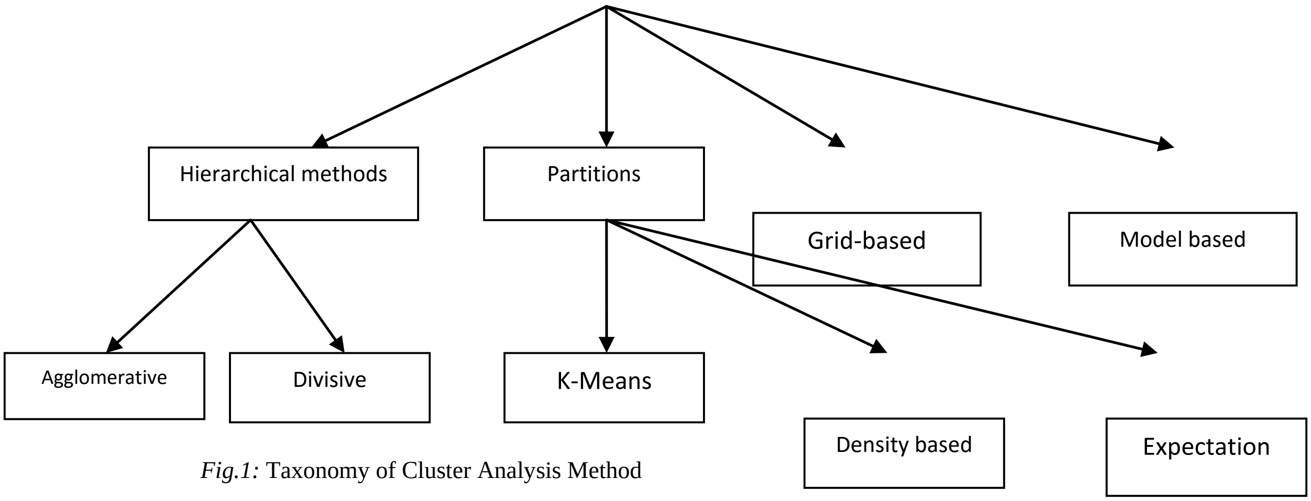


Fig.1: Taxonomy of Cluster Analysis Method

1.2 Partitional methods

Partitional methods are popular since they tend to be computationally efficient and are more easily adapted for very large datasets. The hierarchical methods tend to be computationally more expensive. As noted earlier, the algorithms in this section are iterative refinement methods(sometimes called hill-climbing or greedy methods) and they coverage to a local minimum rather than the global minimum. These methods not only require specifying the number of clusters a priori, they also require the user to normally specify the starting states(or seeds) of the clusters. This may be difficult for a user since the user may not have such knowledge and there is no simple reliable method for finding initial conditions for the clusters.

The aim of partitional methods is to reduce the variance within each cluster as much as possible and have large variance between the clusters. Since the partitional methods do not normally explicitly control the inter-cluster variance, heuristics may be used for ensu8ring large inter-cluster variance. One may therefore consider the aim to be minimizing a ration like a/b where a is some measure of within cluster variance and b is some measure of between cluster variation. We will discuss two methods in this paper, the K-mean methods and the Expectation Maximisation method. Both of these methods coverage to a local minimum. Which minimum they coverage to depends primarily on the starting points.

1.3 K-Means method

K-means is the simplest and most popular classical clustering method that is easy to implement. The classical method can only be used if the data about all the object is located in the main memory. The method is called K-means since each of the K Clusters is represented by the mean of the objects (called the centroid) within it. It is also called the centroid method since at each step the centroid point of each cluster is assumed to be known and each of the remaining points are allocated to the cluster whose centroid is closeset to it. Once this allocation is completed, centroids of the cluster are recomputed using simple means and the process of allocating pointers to each cluster is repeated until there is no change in the clusters. The method also be looked at as a search problem where the aim is essentially to find the optimum clusters given the number of clusters and seeds specified by the user. Obviously we cannot use brute-force or exhaustive search method to find the optimum so we consider solutions that may not be optimal but may be computed efficiently.

2. Web Personalization

The overall process of usage-based Web personalization can be divided into two components. The offline component is comprised of the data preparation tasks resulting in a user transaction file, and the specific usage mining tasks, which in our case involve the discovery of association rules and the derivation of URL clusters based on two types of clustering techniques. Once the mining tasks are accomplished, the frequent item sets and the URL clusters are used by the online component of the architecture to provide dynamic recommendations to users based on their current navigational activity. The online component is comprised of a recommendation engine and the HTTP server. The Web server keeps track of the active user session as the user browser makes HTTP requests. This can be accomplished by a variety of methods such as URL rewriting, or by temporarily caching the Web server access logs. The recommendation engine considers the active user session in conjunction with the URL clusters and the discovered association rules to compute a set of recommended URLs. The recommendation set is then added to the last requested page as a set of links before the page is sent to the client browser. A generalized architecture for the system is depicted in Figure 1. We now discuss the details of each of the architectural components.

2.1 Extracting Usage Data for Web Personalization

The offline component of usage-based Web personalization can be divided into two separate stages. Thefirst stage is that of preprocessing and data preparation, including, data cleaning, filtering, and transaction identification. The second is the mining stage in which usage patterns are discovered via methods such as association-rule mining and clustering. Each of these components is discussed below.

The prerequisite step to all of the techniques for providing users with recommendations is the identification of a set of user sessions from the raw usage data provided by the Web server. Ideally, each user session gives an exact accounting of who accessed the Web site, what pages were requested and in what order, and how long each page was viewed. Two of the biggest impediments to forming accurate user sessions are local caching and proxy servers. In order to improve performance and minimize network traffic, most Web browsers cache the pages that have been requested. As a result, when a user hits the "back" button, the cached page is displayed and the Web server is not aware of the repeat page access. Proxy servers provide an intermediate level of caching and create even more problems with identifying site usage. In a Web server log, all requests from a proxy server have the same identifier, even though the

requests potentially represent more than one user. Also, due to proxy server level caching, multiple users throughout an extended period of time could actually view a single request from the server. The most reliable methods for resolving a server log into user session are the use of cookies or dynamic URLs with an embedded session ID. However, these techniques are not always available due to privacy concerns of the users, or limitations of the capabilities of the Web server. As described in detail, several simple heuristics using the referrer and agent fields of a Server log can be used to identify user sessions and infer missing references with relative accuracy in the absence of additional information such as cookies.

2.2 Preprocessing Tasks

In addition to identifying user sessions, the raw log must also be cleaned, or transformed into a list of page views. Due to the stateless connection properties of the HTTP protocol, several file requests (HTML, images, sounds, etc.) are often made as the result of a single user action. The group of files that are sent due to a single click are referred to as a *page view*. Cleaning the server log involves removing all of the file accesses that are redundant, leaving only one entry per page view. This includes handling page views that have multiple frames, and dynamic pages that have the same template name for multiple page views. It may also be necessary to filter the log files by mapping the references to the site topology induced by physical links between pages. This is particularly important for usage-based personalization, since the recommendation engine should not provide dynamic links to "out-of-date" or non-existent pages. Each user session in a user session file can be thought of in two ways; either as a single transaction of many page references, or a set of many transactions each consisting of a single page reference. The goal of transaction identification is to dynamically create meaningful clusters of references for each user. Based on an underlying model of the user' s browsing behavior, each page reference can be categorized as a *content* reference, *auxiliary* (or *navigational*) reference, or *hybrid*. In this way different types of transactions can be obtained from the user session file, including content-only transactions involving references to content pages, and navigation-content transactions involving a mix of pages types. The details of methods for transaction identification are discussed in. For the purpose of this paper we assume that each user session is viewed as a single transaction. Finally, the session file may be filtered to remove very small transactions and very low support references (i.e., URL references that are not supported by a specified number of user transactions). This type of support filtering can be important in removing noise from the data, and can provide a form of dimensionality reduction in clustering tasks where URLs appearing in the session file are used as features. Given the preprocessing steps outline above, for the rest of this paper we assume that there is a set of *n* unique URLs appearing in the preprocessed log:

$$U\ url\ url\ url_n = \{ \mathrm{url}_1, \mathrm{url}_2, \ldots., \mathrm{url}_n \}$$

and a set of *m* user transactions:

$$T\ t\ t\ t_m = \{ \mathrm{t}_1, \mathrm{t}_2, \ldots, \mathrm{t}_m \}$$

where each $t_i \in T$ is a non-empty subset of *U*.

3. *Data Set*

For experimental purposes, we used the log file containing information about all web requests to the institution’s official website on June, 2016. This was an arbitrarily selected day, and we were not aware of any special activities at the institution on that day which could have led to any unusual behaviour of the website visitors. Each line in the web usage log file contains information about one web resource request, the time of request, the URL requested, as well as other information (IP, web browser info, etc.) that can be ignored when mining association rules of the web pages requested in various sessions. The raw web log file we used for the experiment contained 5999 web requests. This file can be found at http://www.kamarajarmatriculationschool.com/vtsnsNov16 . After the data preparation process, the file contained 426 user sessions (sets of pages visited during the same visit to the website).

3.1. Analysis of the Log File

First Part of Analysis was preprocessing. Preprocessing segregated all the details provided in the log file into a structured form.

Language used: JAVA

Data Structures used: Linear Arrays: ip[], time[], content[], httpmethod[], httpstatus[], bandwidth[], browser[] etc.,

The preprocessing program collected the details in the appropriate data structures and also identified whether an entry is a bot entry or a valid user entry. After much research and analysis of bot identification and removal, we came up with a method specific to Dspace log file to do the same. The pseudocode for the method is as follows.

3.1.1 Pseudocode for BotID

```
begin
  while(!EOF)
  begin
    readLine();
    Check for keywords (bot, slurp, spider) in
    browser[] array
    if the array contains keyword
    begin
      botflag=true;
      botcounter++;
    end
    else
      botflag=false;
  end
end
```

The bot entries are not considered as valid entries while defining user sessions.

Identification of User Sessions

User or Sessions in Web Usage Mining generally refers to the usage or access of any content of the website from a fixed IP over a fixed period of time. The period of time is subjective to the analyzer. Considering the above requirements, we came up with a method specific to Dspace log file to identify user sessions in the log file. The pseudocode for the method is as follows:

3.1.2Pseudocode for SessionID

```
begin
while(!EOF)
begin
  i=1;
  add first not bot entry to session i;
  for each (next entry)
  begin
    if(entry != bot)
      if(IP == Previous IP)
        if(time[this entry] - time[this entry -1] < x)
          add entry to session i;
        else
        begin
          i++;
          add entry to session i;
        end
      else
      begin
        i++;
        add entry to session i;
      end
  end
end
```

3.2 Transforming the file format

The association rule finding algorithm in (Weka3) accepts input files in a format called Arff (Weka, n.d.). An Arff file has a header containing the list of all attributes and a list of transactions, each of which contains the list of all attributes and their values in each transaction. This more naturally corresponds to mining data in relational tables then the market basket type data, which is analogous to web session data in our case. There are two possible representatives of data in the Arff file – dense and sparse. In both formats a web page must be considered as a binary attribute that takes the values true or false in each transaction, depending on whether the page occurs in the transaction or not. Since the occurrence of web pages in transactions (user sessions) is scarce, we found the sparse format to be more appropriate for presenting sessions in web log data. Since WumPrep (or any other tool that we could find) does not contain any script that converts web log data into either sparse or dense Arff file format, we developed a Perl script for this purpose. The Arff file that resulted from our data preparation that we used in our experiments can be found at http://www.kamarajarmatriculationschool.com/vtsnsJune16 .

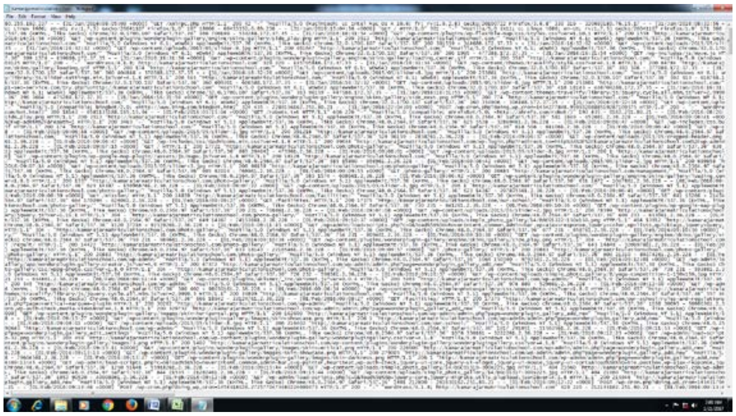


Fig.2: Sample log file before preprocessing

3.3 WEB LOG FILE FORMAT

A. Apache Log Format

Access log records hits and related information. Moreover in Apache, access log are formatted in three ways: Common Log Format, Combined Log Format, Multiple Access Log. By default Apache uses the Common log format, however, the majority of hosting providers set the Combined log format for Apache on their servers. Log format can be configured by editing the "httpd.conf" file in the Apache conf directory (if you have access to this file)[4]. Combined log format with the addition of two more attributes: User agent and Referer. Therefore, Combined log format contains more information than Common log format. The configuration of the combined log format is given below: LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\ " \"%{User-agent}i\"" combined CustomLog log/acces_log combined [5]. Each element of format string that specifies the log format is described below:

%h: IP address of the client (Remote Host) accessing the server.

%I: RFC 1413 identity of the user determined by identd (Normally Unavailable).

%u: User id of the user determined by HTTP authentication.

%t: The date, time and time zone when process of HTTP request is completed by Web server.

”%r\”: The request line from the client to website.

%>s: Status code of the request that the server delivers to the client.

%b: Size of the server’s response in bytes returned to the client.

”%{Referer}i\”: “Referer” is an HTTP request header field. This gives the address of Web page from where the request originated.

”%{User-agent}i\”: The type of Web browser (software) that acts on user’s behalf.

B. IIS Log Format

Different log file formats supported by Internet information server where user gathers information about client request can be IIS, W3C, NCSA, CUSTOM. This format logs client’s activity and server’s activity into a log file in selected log file format. Therefore, W3C extended format is customizable ASCII format. This is probably the default log file format and commonly used log format, moreover it offers a selection of fields that are included in the log file with which user can limit the size of log file and obtain the detailed information. W3C extended log format is given below: #Software: Microsoft Internet Information Services 7.5 #Version: 1.0 #Date: 2015-06-01 06:44:16 #Fields: date time s-ip cs-method cs-uri-stem cs-uri-query s-port cs-username c-ip cs(User-Agent) cs(Referer) sc-status sc-substatus sc-win32-status time-taken The available fields in W3C extended log format is described as follows:

date: The date on which the HTTP request received by server.

time: The time at which the request occurred.

s-ip: Server IP

cs-method: The HTTP requested action.

cs-uri-stem: Stem (Path) portion of the requested Uniform Resource Identifier.

cs-uri-query: Query portion of the requested Uniform Resource Identifier.

s-port: The server port number of the listener that is configured for the service.

cs-username: Name of authenticated user who accessed the server. If user was anonymous, a hyphen (-) is logged instead of user name.

c-ip: Client IP address.

cs(User-Agent): The browser type used on the client.

cs(Referer): The Referer field identifies the URI that linked to the server being requested..

sc-status: The server sends HTTP response code.

sc-substatus: The sub status error code of the HTTP.

sc-win32-status: The server sends the status of the action in window system error code.

time-taken: The length of time taken to complete the request in milliseconds.

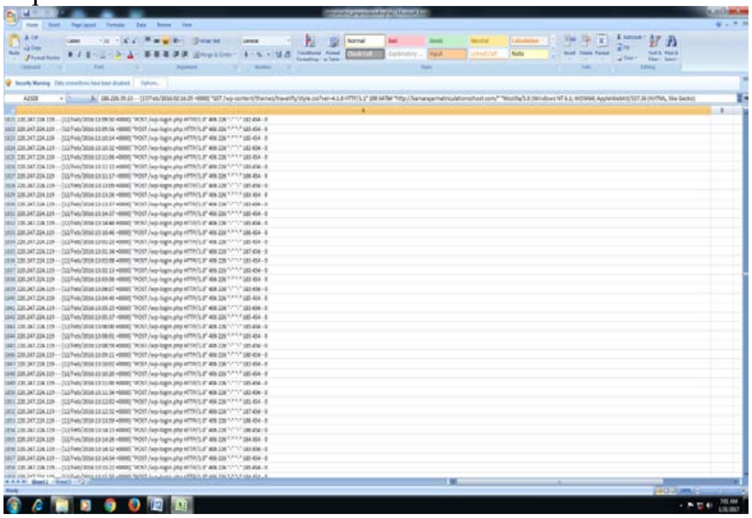


Fig.3: Log file after preprocessing

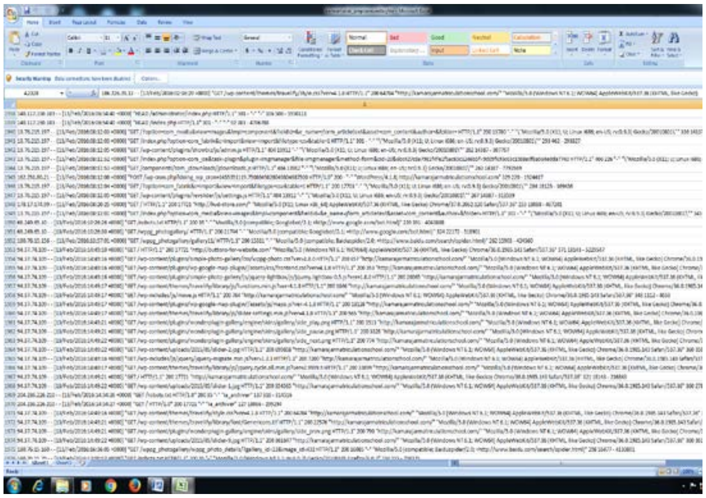


Fig.4: Log file after preprocessing

3.4 WEB LOG ANALYSIS SOFTWARE

Web log analysis software (Web analytics software) are essential tools, which are categorized based on their popularity, functionality and simplicity of usage. A variety of Web log analyzers are available that take Web server logs as an input and graphical reports are generated from the log files immediately. In addition to this, a powerful Web log analyzer performs analysis and brings visibility into the website access, which makes it an essential analyzer for business decision making and market research. Some of the tools that are available are: WebLog Expert Lite, AlterWind Log Analyzer, IIS and Apache Log Analyzer, Deep Log Analyzer, AWStats.

RESULTS AND INTERPRETATION: WebLog Expert Lite (Web server log analyzer) is a free Web mining tool, light weight version of WebLog Expert for windows based computer. It can analyze the log file of Apache, IIS and get information about the sites visitors: general statistics, activity statistics, access statistics, referrers, search engine, browser, operating systems, errors and more. The software supports the Apache web server in Common (default log format) and Combined log formats and IIS Web server in W3C Extended log format (Default log format) of 4/5/6/7/8. It can read logs in formats such as LOG, ZIP, GZ and BZ2.WebLog Expert Lite generates easy- to- read HTML-file reports with graphical and tabular formats. Thus, researcher analyzed Apache and IIS sample log file with the help of WebLog Expert tool. Time range of sample log files: Apache: 10/June/2015 07:04:34 – 31/December/2015 22:48:23 and IIS: 01/January/2016 00:00:03 – 31/Mar/2016 23:59:51.

3.5 Experimental results

A. Analysis on Apache Log File: The following results were obtained to identify the behavior of the website users on Apache Web server.

General Statistics: This shows the Web usage details (General Information) that also includes total hits and average hits per day, total page views and average page views per day, total visitors, total bandwidth etc.

Activity Statistics: This statistics gives the information about the user's activity by date and hour of day. Date and time are essential attributes in web log file because user's activity by date and hour of day provides the details of hits, page views, visitors, bandwidth etc. According to graph given above the Website is hit maximum at 05:00 hrs and is least visited at 03:00 hrs.

Access Statistics: Here the statistics for most popular pages, most downloaded files, most requested images, most requested directories, top entry pages and daily page access, image access, directory access, entry pages are shown. Entry page is a first Web page visited by visitor on the site. This statistics also provide an idea of the navigational behaviour of visitors.

Visitors: It shows the list of IP address/domain names of hosts that accessed the website along with the number of times the website was hit by a particular host. Visitors section analyse the IP address field of Web log file.

Referrers: Here the report displays the top referring sites, top referring URLs and top search engine. This section collected referring sites and referring URLs from referer field of Apaches Web log file.

Browser: This report helps the website owner to analyze the Web browser mostly used by the visitors so that the website can be made compatible with that particular browser. It also provides the list of most preferred Operating systems used by the users and different versions of Internet Explorer. Therefore, report is generated with the help of user agent field of web log file.

S.NO	File Type	Hits	Percentage
1	GIF	2,806,156	35.6
2	(no type)	1,726,743	21.9
3	HTML	1,289,428	16.3
4	JPG	812,420	10.3
5	TXT	217,048	2.8
6	JPEG	203,393	2.6
7	C	154,992	2.0
8	H	148,686	1.9
9	PL	108,468	1.4
10	ICO	95,986	1.2
11	194 other items	326,363	4.1
	Total	**7,889,683**	**100.0**

Fig. 5: File types from 01/June/2015 to 31/March/2016

S.No	Operating system	Visitors	Size	Page views
1	Windows XP	3,325	1.51 G	3,986
2	unknown	408	1.03 G	4,610
3	Windows 98	829	0.76 G	2,580
4	Windows 2000	476	0.77 G	2,184
5	Windows 95	2,865	1.35 G	24,537
6	Windows ME	1,957	0.2 G	9,650
7	Windows NT	279	0.30 G	7,245
8	Power Macintosh	366	0.03 G	7,063
9	unspecified	522	0.91 M	799
10	Linux	1,854	0.27 G	4,431
	173 other items	3,854	2.79 G	13,902
	Total	**16,735**	**9.92G**	**80,987**

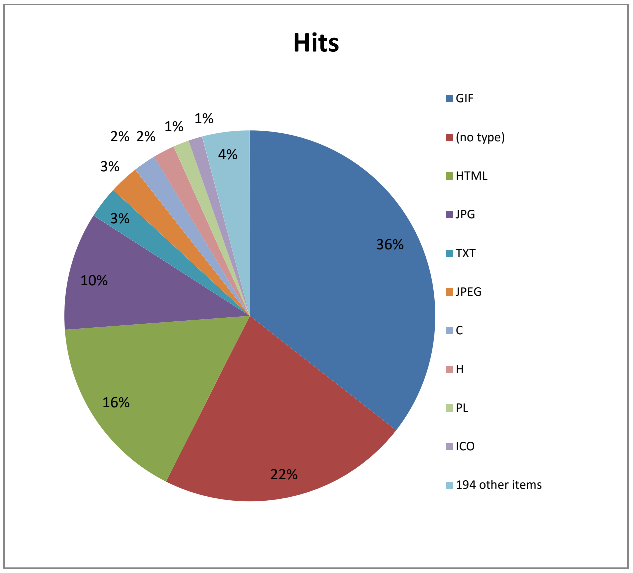


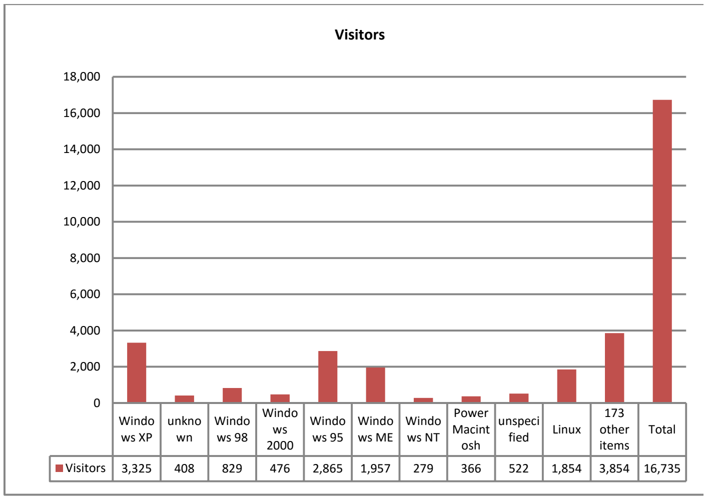
Fig. 6 : Graphical representation for the File types from 01/June/2015 to 31/March/2016

B. Analysis on IIS Log File: WebLog Expert tool analyzed IIS Web server with the same way as the Apache Web server. So that researcher described only Access statistics features **.**

Access Statistics: Here it shows the same information that is described in access statistics of Apache log file. URI Stem field contains information about pages, images, files, directory etc. Thus, WebLog Expert tool analyze the data from URI Stem attribute of log files.

Important Attributes of Web log files According to the analysis of WebLog Expert Lite tool that used sample log files of Apache and IIS, researcher analyzed fields and obtained important attributes of Web log files that are IP Address, Date, Time, Request Line or URI Stem, Status Code, Referer and User-Agent etc.,

Fig.7 : Operating system used from 01/Jun/2015 to 31/March/2016



	Windows XP	unknown	Windows 98	Windows 2000	Windows 95	Windows ME	Windows NT	Power Macintosh	unspecified	Linux	173 other items	Total
🟥 Visitors	3,325	408	829	476	2,865	1,957	279	366	522	1,854	3,854	16,735

Fig.8 : Graphical representation for used Operating system from 01/Jun/2015 to 31/March/2016

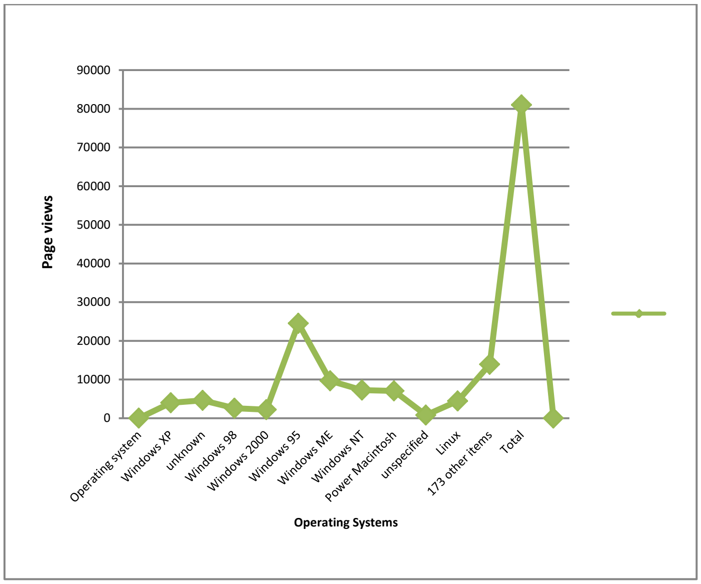


Fig.9 : Graphical representation for number of pages viewed from 01/Jun/2015 to 31/March/2016

4. Conclusion

The Web is providing a direct communication medium between the vendors of products and services, and their clients. Coupled with the ability to collect detailed data at the granularity of individual mouse clicks, this provides a tremendous opportunity for personalizing the Web experience for clients. Recently there has been an increasing amount of research activity on various aspects of the personalization problem. Most current approaches to personalization by various Web-based companies rely heavily on human participation to collect profile information about users. This suffers from the problems of the profile data being subjective, as well getting out of date as the user preferences change over time. We have provided several techniques in which the user preference is automatically learned from Web usage data, by using data mining techniques. We describe a general architecture for automatic Web personalization based on the proposed techniques, and discuss solutions to the problems of usage data preprocessing, usage knowledge extraction, and making recommendations based on the extracted knowledge. Our experimental results indicate that the techniques discussed here are promising, each with its own unique characteristics, and bear further investigation and development.

5. References

[1] Han, E-H, Karypis, G., Kumar, V., and Mobasher, B., Clustering based on association rule hypergraphs. In *Proccedings of SIGMOD'97 Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD'97),* May 2007.

[2] Han, E-H, Karypis, G., Kumar, V., and Mobasher, B., Hypergraph based clustering in highdimensional data sets: a summary of results. *IEEE Bulletin of the Technical Committee on Data Engineering*, (21) 1, March 2008.

[3] Eric Schmitt, Harley Manning, Yolanda Paul, and Sadaf Roshan, Commerce Software Takes Off, *Forrester Report*, March 2000.

[4] Eric Schmitt, Harley Manning, Yolanda Paul, and Joyce Tong, Measuring Web Success, *Forrester Report*, November 1999.

[5] Gregory Piatetsky-Shapiro, Ron Brachman, Tom Khabaza, Willi Kloesgen, and Evangelos Simoudis, An Overview of Issues in Developing Industrial Data Mining and Knowledge Discovery Applications, *Proceeding of the second international conference on Knowledge Discovery and Data Mining*, 1996.

[4] Ralph Kimball, *The Data Warehouse Toolkit: Practical Techniques for Building Dimensional Data Warehouses*, John Wiley & Sons, 1996.

[5] Ralph Kimball, Laura Reeves, Margy Ross, Warren Thornthwaite, *The Data Warehouse Lifecycle Toolkit : Expert Methods for Designing, Developing, and Deploying Data Warehouses*, John Wiley & Sons, 1998.

[6] Mao Chen, Andrea S. LaPaugh, and Jaswinder Pal Singh. Predicting category accesses for a user in a structured information space. In Proceedings of the 25th annual international ACM SIGIR conference on Research and development in in-formation retrieval, pages 65–72, 2002.

[7] R. Cooley. Web Usage Mining: Discovery and Application of Interesting Patterns from Web Data. PhD thesis, University of Minnesota, 2000.

[8] Robert Cooley, Bamshad Mobasher, and Jaideep Srivastava. Data preparation for mining world wide web browsing patterns. Knowledge and Information Systems, 1(1):5–32, 1999.

[9] Magdalini Eirinaki and Michalis Vazirgiannis. Web mining for web personalization. ACM Transactions on Internet Technology (TOIT), 3(1):1–27, 2003.