J Intell Inf Syst (2007) 28:79–104
DOI 10.1007/s10844-006-0004-1

A process of knowledge discovery from web log data: Systematization and critical review

Zidrina Pabarskaite · Aistis Raudys

Received: 30 November 2003 / Revised: 3 April 2005 /
Accepted: 20 July 2005 / Published online: 28 December 2006

Abstract This paper presents a comprehensive survey of web log/usage mining based on over 100 research papers. This is the first survey dedicated exclusively to web log/usage mining. The paper identifies several web log mining sub-topics including specific ones such as data cleaning, user and session identification. Each sub-topic is explained, weaknesses and strong points are discussed and possible solutions are presented. The paper describes examples of web log mining and lists some major web log mining software packages.

Keywords Web log mining · Web usage · Personalisation · Survey · Data pre-processing

1 Introduction

Apart from hosting web pages, web servers are valuable source of information. Web site visitor's actions can be logged and later this information can be used for user behaviour analysis. Large quantities of such data are typically generated by e-commerce web servers (Chen, Park, & Yu, 1996; Lin, Huang, & Chen, 1999; Han, He, & Wang, 2000; Pitkow, 1997).

This information is logged into special purpose files called web logs. There are many commercial web log analysis tools.[1] Most of them focus on statistical information such as the largest number of users per time period (activity), types of users (business/education) (.edu, .com) or geographical location (.uk, .ie, .be, etc.), page popularity (number of times page have been visited) etc. However statistics which don't describe the relationships between

[1]NetGenesis, *NetGenesis5*, http://www.netgen.com/, MINEit, *EasyMiner*, http://www.mineit.com/products/easyminer/, SPSS, *Clementine*, http://www.spss.com/clementine/, Angoss, *KnowledgeWebMiner*, http://www.angoss.com/angoss.html/

Z. Pabarskaite (✉) · A. Raudys
Institute of Mathematics and Informatics, Akademijos 4, Vilnius 2600, Lithuania
e-mail: zidrina@pabarska.com

A. Raudys
e-mail: aistis@raudys.com

visited pages consequently leave much valuable information undiscovered (Cooley, Mobasher, & Srivastava, 1997a), (Pitkow & Bharat, 1994). Eric Schmitt et al. in one of Forrester Research reports (1999) said: "Using hits and page views to judge site success is like evaluating a musical performance by its volume." This lack of depth of analytic scope has stimulated web log research. Nowadays it is an individual research field beneficial and vital to e-business components and called web usage mining.

Overview of numerous research works shows main goals which might be achieved mining web log data. These are listed below:

a. *Restructuring websites.* Web log examination enables reorganization of a website to facilitate clients access to the desired pages more easily and with the minimum delay (Pirolli, Pitkow, & Rao, 1996).
b. *Improving navigation.* This is evident when directing important information to the right places, managing links to other pages in the correct sequence, pre-loading frequently used pages.
c. *Specialised/Intelligent adverts.* Attracting more advertisement capital by finding what adverts and pages match the most. Alternatively, specific groups of users may see specific adverts.
d. *Turning non-customers into customers increasing the profit* (Faulstich, Pohle, & Spiliopoulou, 1999). Analyses should be provided for both groups: customers and non-customers in order to identify characteristic patterns. Such findings would help to review both the behaviour of the clients and website maintainers incorporating these observations into the site architecture and thereby assisting in the conversion of non-customers into customers.
e. *Monitoring efficiency of the web site.* Empirical findings (Tauscher & Greenberg, 1997) have observed that people tend to revisit pages recently visited and access only a few pages frequently. Humans browse small clusters of related pages and generate only short sequences of repeated URLs. This shows that there is no need to increase the quantity of information pages on the web site. It is more important to focus on the efficiency of the contents and accessibility of these clusters of pages.

In short, web log analysis allows allocating resources more efficiently in order to find new growth opportunities, improve marketing campaigns, plan new products, increase customer retention and to discover cross selling opportunities better forecasting.

1.1 The aims of this paper

The aims of this paper are to provide the latest available and most useful information about web log mining. As far as the authors are aware there has been no such all-embracing paper describing web usage mining process in such detail. The nearest such works which come closest to discussing this issue are dealt with by Kosala and Blockeel (2000), Chakrabarti (2000), and Buchner, Mulvenna, Anand, and Hughes (1999) and provide an extensive survey of web mining categories, analysis of web mining from the point of view of different research directions: database, information retrieval, natural language processing. Chakrabarti's tutorial survey is about the existing methods of retrieving desired knowledge from search engines. Buchner and others go through knowledge discovery steps, including data collection sources and data pre-processing, discovery of patterns, understanding of knowledge and steps of refinement. However this study is very limited since it neither identifies the majority of the problems nor provides solutions in pre-processing web data.

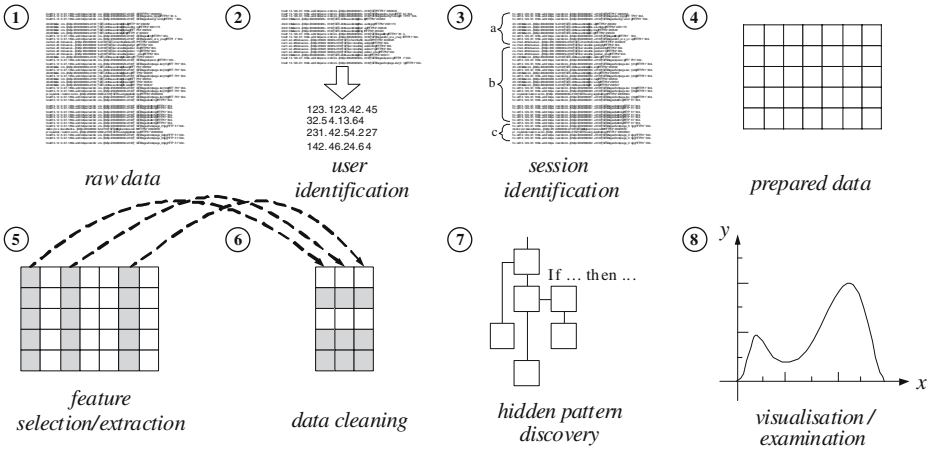

Fig. 1 Knowledge discovery process from web log data

Therefore, this paper surveys a number of research papers scattered across general purpose journals and conference proceedings. All material is gathered and systemized in one peace of work. This paper is addressed to the broad auditorium of web usage/log analysts.

1.2 What this papers is, and is not about

Web usage mining can be based on client or server data. The difference between those two is in the data used for mining. Data collected on the client side is data from the client's web browser or proxy/cache server (sits in between the client and server). As this data usually covers multiple web sites it is more suitable for problems appertaining as to how to improve response time by preloading predicted pages or preventing a user from browsing non ethical content's of web pages. This kind of analysis is more like a web proxy/cache optimisation area and is a separate research topic. Only a few papers on web caching and intelligent caching can be mentioned (Duska, Marwood, & Freeley, 1997; Wang, 1999; Davison, 1999; Barish & Obraczka, 2000; Bonchi et al., 2001b).

This paper addresses server side processes only. Server side data is generated from requests made to the web server through the hypertext transfer protocol. Results are presented in a user's browser.

1.3 Web log mining process

The process of *knowledge discovery from databases* (KDD) is applicable to many types of data. Due to the peculiarity of web log data some steps are unique (see Fig. 1). The following steps in web log mining can be identified:

1. Data collection. Web servers collect data by logging all requests to a log file.
2. Data cleaning. Certain record types are removed as they are part of other web pages and are never requested in isolation.
3. User identification. Unique users are identified. This can be done in various ways: using cookies, cleaning non unique users, using IP addresses etc.
4. Session identification. Transaction flow is broken into separate user browsing sessions.

5. Feature selection. Only relevant fields are left for analysis.
6. Data transformation. Certain features are transformed to suit methods of analysis in better way.
7. Data combination. Data is combined with other databases, e.g., user personal info database or text contained in these pages of these databases.
8. Mining the data. An appropriate method of analysis is applied depending on the task: association rules, clustering, classification or prediction.
9. Result visualisation. Preparation of various charts, graphs and reports.

These are main steps. Depending on the task and available data, the sequence of tasks may vary.

Due to the nature of web log data many unclear issues exist in steps 2, 3 and 4. Questions like what records from the raw file are relevant, how unique user must be identified and how browsing sessions must be detected are not always answered correctly. Thus theoretical concepts and a variety of different techniques are discussed here to address these kinds of problems.

1.4 Paper structure

The paper is organized as follows. Section 2 depicts web protocol, web servers, web log data, the most popular log formats and some related topics. Section 3 explains data pre-processing and various issues related to that. Section 4 covers web log mining methods. Section 5 surveys final KDD step: visualisation and result presentation. Section 6 discusses software tools available on the market. Conclusions are presented in Section 7 and references in Section 8.

2 Web log data

In order to understand web log mining better, web browsing process must be clear. Web browsing is performed as follows:

1. In the beginning, user types URL in his web browser.
2. Web browser sends request for specified page to the internet.
3. If a proxy/cache server is located browser and web server it intercepts the request and acts as a middleware. It resends the request to the internet and sends the received response back to the client.
4. Web server receives request from client, proxy finds requested page and sends this page back. At the same time server logs request to the log file.

2.1 Data formats

Web log data is usually generated by web servers and written to special logs files. These files contain all information about visitors' activities (Kosala & Blockeel, 2000). Typically the amount of information generated is huge. For example, the web log file generated while running the online information site[2] produces log with the size of 30–40 MB over a period

[2] http://www.munichfound.de/

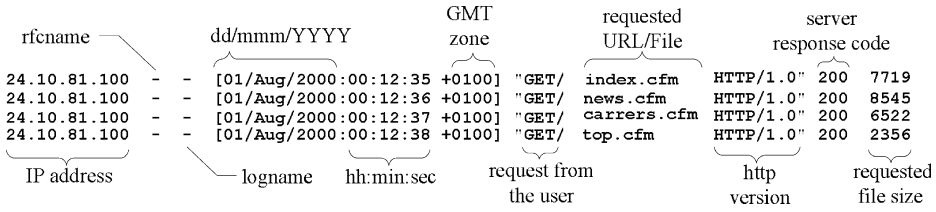

Fig. 2 Example of the Common Log Format: IP address, authentication (rfcname and logname), date, time, GTM zone, request method, page name, HTTP version, status of the page retrieving process and number of bytes transferred

of 1 month. Another advertising company[3] collects the log of size 6 MB each day. Popular web sites as news and search engines probably generate gigabytes of web logs each day.

Various web servers generate different formatted logs: Apache Common Log File Format, CERF Net, Cisco PIX, Gauntlet, IBM Firewall, IIS standard/Extended, Microsoft Proxy, NCSA Common/Combined, Netscape Flexible, Open Market Extended, Lotus Domino Log File Format and Raptor Eagle. Examples of different log file formats may be found on the web.[4] Nevertheless, the most common is so called common log format (CLF), a.k.a. Combined Log Format.

2.1.1 Common log format

Common Log Format a.k.a. Combined Log Format appears as follows (see Fig. 2):

`[host/ip rfcname logname [DD/MMM/YYYY:HH:MM:SS -0000] "METHOD /PATH HTTP/1.0" code bytes]`

`host/ip`	is the visitor's hostname[5] or IP[6] address, i.e., from where the visitor is making a connection.
`rfcname`	returns user's authentication. It operates by verifying specific TCP/IP connections and returns the user identifier of the process who owns the connection. If the value is not present (as is usually the case) it is indicated by a "-" character.
`logname`	user's login name in local directory gives access to authentication, if the value is not present (as is usually the case), a "-" is indicated.
`[DD/MMM/YYYY:HH: MM:SS -0000]`	access date defined by day (DD), month (MMM), year (YYYY), hours (HH), minutes (MM) and seconds (SS). The last symbol stands for the difference from Greenwich Mean Time (for example, Pacific Standard Time is − 0800).
`METHOD`	page/file access method: PUT, GET, POST and HEAD. PUT allows the user to transfer/send a file to the web server. By default, PUT is used by web site maintainers having administrator's privileges. For example, this method allows uploading files through the given form on the web. Access to this method by ordinary users is forbidden. This method rarely appears in web logs. GET transfers the content of the web document to the user. This is the most popular method.

[3] http://www.ipa.co.uk

[4] http://www.utpb.edu/siteserver/docs/. 2003

[5] Hostname—is the name of the machine where www or mail server is running.

[6] IP stands for the computer identification number. It can be static or dynamic.

	POST is similar to GET as it retrieves the file, but it posts information to the web server as part of the request. Information is enclosed in the body of the request and is transferred to the server. POST information usually goes as an input to the dynamic pages such as Common Gateway Interface (CGI) programs.
	HEAD demonstrates the header of the "page body". Usually it is used to check the availability of the page, date and size (Savola et al., 1996).
`PATH`	stands for the path and file name retrieved from the web server.
`HTTP/1.0`	defines the version of the protocol used by the user to retrieve information (1.1 and 1.0 are the most common).
`Code`	identifies server's response (success status). For example, 200 means that the file has been retrieved successfully, 404—the file was not found, 304—the file was reloaded from cache, 204 indicates that upload was completed normally and etc. The complete list of response statuses can be found on the web.[7]
`bytes`	number of bytes transferred from the web server to the user.

2.1.2 Extended common log format

The special type of common log format is extended common log format which is very valuable in web log mining, since it has some additional information like REFERER_URL, HTTP_USER_AGENT and HTTP_COOKIE (Fleishman, 1996). REFERER_URL defines the URL where the visitor came from. HTTP_USER_AGENT identifies the visitor's browser version. HTTP_COOKIE is a persistent token, which defines the cookie sent to a visitor. The extended CLF form containing those three additional fields is displayed in Fig. 3.

3 Data preparation

Web log data pre-processing is a complex process. It can take up to 80% of the total KDD time (Ansari, Kohavi, Mason, & Zheng, 2001) and consists of stages presented in Fig. 4. The aim of data pre-processing is to select essential features, clean data from irrelevant records and finally transform raw data into sessions. The latter step is unique, since session creation is appropriate just for web log datasets and involves additional work caused by user identification problem.

3.1 Data cleaning

Data cleaning is an important step in data mining, data warehousing and other database areas, but it has received relatively little attention from the research community (Lup Low, Li Lee, & Wang Ling, 2001). *Data cleaning* involves various tasks to produce data which can be exploited successfully (Fayyad, 1996). It is also a time consuming process. In many cases it is still semi-automatic and therefore not efficient (Famili, Shen, Weber, & Simoudis, 1997). Data cleaning had to be automated to make data sharing affordable to monitor and evaluate. Infoshare[8] a leading data cleaning enterprise estimated the ratio of costs for cleaning 1,000 records manually and automatically. In comparison, automated data cleaning takes about 20 min and cost 0.048 EUR per record, while manual human work takes 10 days and 1.491 EUR per record.

[7] http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html. 2003

[8] Infoshare, http://www.infoshare-is.com/. 2002

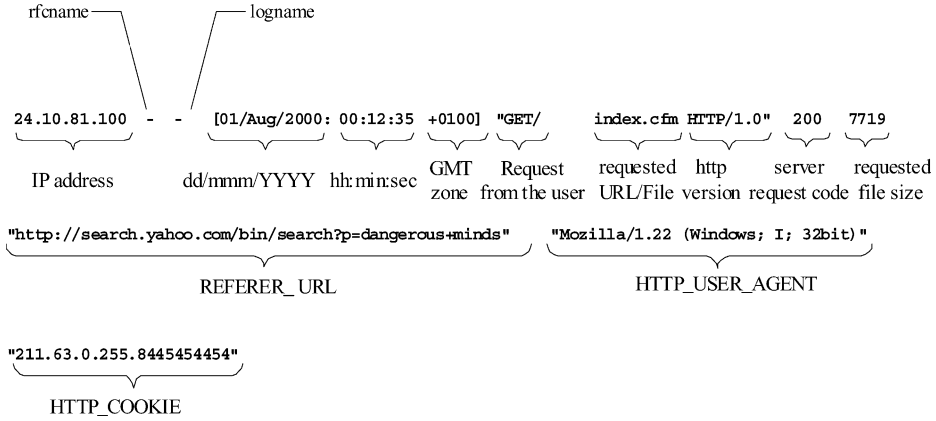

Fig. 3 Extended common log format followed by additional data fields: web page name visitor gets from – referrer page, browser type and cookie information

3.1.1 Feature selection (column removal)

Log files usually contain nonessential information from the analytical point of view. Thus the first data pre-processing step is selection of features. Moreover, reducing the number of features at this stage decreases the memory usage and improves performance. It is also beneficial from the computational point of view, since log files contain thousands of megabytes of data. The final output of the pre-processing must be divided into sessions. The key attributes to build sessions are page ID, computer's IP address (or host) and page request time. These are the main features to work with in web usage mining. Other features are less relevant unless participating in some specific tasks. For example, the status of the retrieved page or number of bytes downloaded whilst accessing the page may be utilised for statistical purposes only: measuring the network load or number of bytes sent to the client. For example, HTTP_USER_AGENT and HTTP_COOKIE are used by some methods identifying unique users (see Sections 3.2.3 User identification by the client type and 3.2.5 User identification by the cookies) and used for session creation.

3.1.2 Data filtering (record removal)

There are several reasons for record cleaning/removal. These are listed below.

3.1.2.1 Specific files

Most web log records are irrelevant and require cleaning because they do not refer to pages clicked by visitors. Research in this area is confined by removing graphics (images, sound, video) files (Cooley, Mobasher, & Srivastava, 1999a, b; Faulstich, Spiliopoulou, & Wilkler, 1999; Han, Xin, & Zaïane, 1998;). Graphic files in web pages are downloaded together with the requested html document. They are not objects requested by the visitor therefore they are likely to be redundant. In the real world data, irrelevant files are found in a ratio 10:1. It can vary depending how many graphic and other files web pages contains (Cooley et al., 1999b). However, more intelligent record removal methodologies exist. Some of methodologies of record removal are (a) automated processes (search robots) (Tan & Kumar, 2002), (b) pages whose time interval between requests is too short to catch the contents of the page, (c)

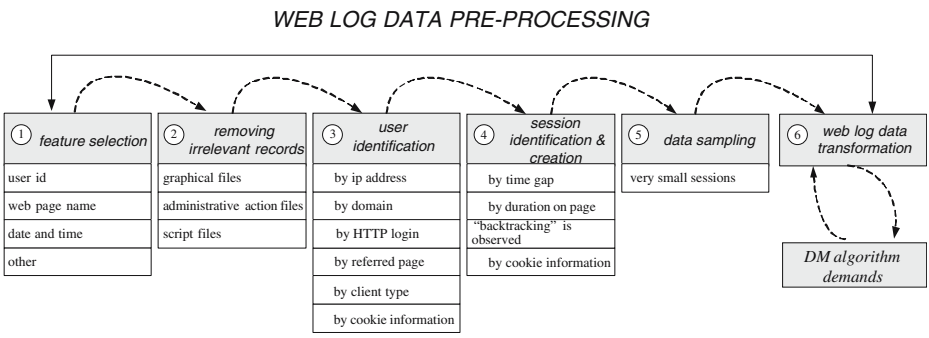

Fig. 4 Web log data pre-processing is the most complex part in KDD process since it requires additional knowledge and understanding of the field. Horizontal axis contains preparation steps and vertical axis contains methods to make this step

repeated requests for the same URL made from the same host during the same session, (d) requests where referrer page is empty, meaning this URL was typed by the user or this page was bookmarked or requested by some program-script (Berendt & Spiliopoulou, 2000), (e) common .cgi scripts (Cooley et al., 1999b).

However, the process is not so insignificant, as sometimes the files to be removed depend on the data source and on the domain to be analysed. For example, if graphical repository is analysed and the task requires identifying associations between accessed/viewed images, then files referring to images should remain. To some extent CGI commands can also be a part of analysis, especially in information retrieval, because they include information about query terms, search method (i.e., simple search or advanced search), page number (i.e., subsequent pages for the same search) (He & Goker, 2003).

We recommended the most advantageous up to date web log data cleaning methodology by imitating real clicks (Pabarskaite, 2002). In the first cleaning stage information from the web server is retrieved. Web log file serves as a source of web pages to be downloaded. Special developed script takes every distinct page and requests it from the web server. This engine simulates mouse click operation. Web server returns the content (body) to the page analyser engine, which then parses the HTML code, detects links by an anchor <a>.... The procedure is repeated for all pages from the log file. The next step is filtering. Those requests on which are impossible to click are removed. This approach performs very high quality cleaning, saves time for on updating files for removal but, at the same time requires access to the internet and gives rise to a high volume of traffic.

3.1.2.2 Too rear or too frequent records

The last group of irrelevant records are very frequent items occurring in every transaction or session. Such frequently occurring pages are usually entry pages to the web site (\index. html) or login pages accessed by every user.

Another group is rear pages, for example a session with only one entry. Sometimes the user just gets into the web site and leaves after one or two steps. However such visits cannot construct meaningful clusters of visited pages and do not bring significant knowledge about the web site usage.

The same can be said when the pages accessed only once. They can not be a part of a meaningful pattern as there is no repetition of that page in other sessions.

An operator who performs data pre-processing should, in fact, refer to requirements of the task and filter items, which are too small.

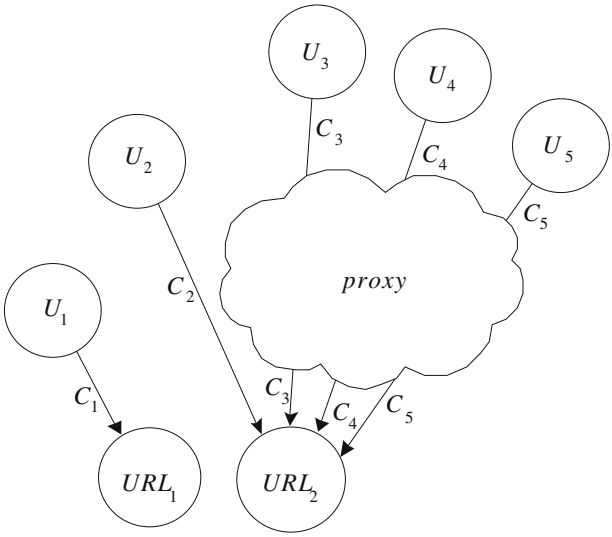

Fig. 5 Communication between users and requested pages with and without proxy/cache server. U_3, U_4 or U_5 will be seen as one user (one IP)

3.2 User identification

The next important step is unique user identification. This problem arises because external/local proxy, cache systems and Internet sharing do not allow easy user identification.

One drawback in exploiting proxy servers is that they do not permit every user to be identified. When a request to the web server is submitted and several users access the Internet through the same cache/proxy server, the web server logs only one user (web server will see only one proxy IP address—the address of the proxy server). In practise, more than one user could be making these requests. The proxy effect on the web server is illustrated in Fig. 5. Different users are defined as U_1, U_2, U_3, U_4, U_5. Connections/requests are indicated by C_1, C_2, C_3, C_4, C_5 and URL_1, URL_2 represent web pages on different web servers. If U_1 or U_2 browse URL_1, URL_2 everything is fine as they make connections C_1, C_2 directly. The situation is different when U_3, U_4 or U_5 (connections C_3, C_4, C_5) request the page URL_2, then the server identifies three connections with the same IP/host (user identification) and assumes that the same user makes all three requests. Similarly with proxy servers, cache server existence does not allow every user to be identified when the request of the page is repeated. Cache keeps all previous requests in a special repository—cache. So, whenever a user accesses previously requested pages, the browser takes them from the local cache repository and the web server of the original page location does not log any activity. So in a case of C_3, C_4, C_5 only one request for URL_2 may be logged.

Surely there are methods by which users could be correctly identified. These are by obtaining cookies from the users or by sending remote agents, described in the research works of Elo-Dean and Viveros (1997) and Shahabi, Zarkesh, Adibi, and Shah (1997). But these methods require user's collaboration which is not always possible.

Solutions on how to address these problems are described in the following sections.

3.2.1 Rejecting records from proxy servers

An alternative on how to partially solve the problem of proxy/cache users was already proposed by us in our recent article (Pabarskaite, 2003) and this is by rejecting log entries

from proxy and cache server. However, that leads us to another problem: how to identify that a host is a proxy server. A domain[9] name can help to detect if a visitor is connecting through a proxy/cache server. For example, if the Internet connection is done through a modem, then domain definition can contain the name such as "dialup" or "ADSL" (if user uses broadband connection). If a user accesses the Internet through the proxy/cache server then the domain definition may contain words such as "proxy" or "cache" (e.g., spider-th031.proxy.aol.com or cache-dr05.proxy.aol.com). Removing records with domain names containing "proxy" or "cache" words allows filtering unfamiliar visitors. One limitation of this approach is that some important patterns made by proxy/cache users will remain undiscovered. Another drawback of this method is that data size is significantly reduced, sometimes even several times. This is extremely painful if the quantity of available data is small.

3.2.2 User identification by the http login

The ideal way to identify users would be using registration information entered by users themselves. Fields in the log file *rfcname* and *logname* (see Section 2.1.2) can uniquely identify a visitor if they are used, although this login process is not popular. Moreover, humans usually try to avoid web sites where registration information is required, unless, it is a special purpose site such as on-line banking. Then login is inevitable. However no evidence of such cases was found in the literature. When login is used for user identification this is not a problem at all.

3.2.3 User identification by the client type

Most literature sources do not investigate proxy names but use some other heuristics to identify unique visitors with the same IP address. Extended CLF file may contain additional client information—agent field HTTP_USER_AGENT containing browser's name and operating system. Thus, one approach to identify a unique visitor is to trace the agent and look at the changes in a browser and operating system (Pirolli et al., 1996). If two log entries with the same IP have different agent information, an assumption is made that requests are made from two different sources. An example of such situation is presented in Table 1. Though this approach justifies itself, sometimes it can produce misleading results. There maybe a small theoretical chance that the same visitor uses different browsers or even different machines and explores the same web site at the same time (for example user has 2 windows opened to compare some info on two different pages of the same web site). In this case a visitor's browsing history will be assumed as representing the activities of two. The opposite can also occur as well. If, for example, two visitors with the same IP address (they are connected to the Internet through the proxy server or Internet connection sharing NAT[10]) use the same kind of browser and operating system and navigate the same web site, they can be confused as being a single user.

3.2.4 User identification using site topology

This approach uses site topology to construct browsing paths for each user. Cooley et al. (1999b) developed a new hypothesis: if from a set of pages a new page appears which is not accessible from the previously viewed pages, a new user is assumed.

[9] Group of computers are united physically (through the network) into some unit. The name of that unit is called domain name or just domain.

[10] NAT—Network Address Translation

Table I Two entries in extended CLF web log file. As can be seen, here is a typical situation when the same IP shows the possibility of two different visitors. However, web agents are different on these two records and therefore, different visitors are assumed:

IP	Login	Password	Date	Method/page/ version	Response code	File size	Referral URL	Browser version
24.10.81.100	–	–	[01/Aug/2000: 00:12:35 +0100]	"GET/ index.cfm HTTP/1.0"	200	7719	"http://search.yahoo.com/bin/ search?p=dangerous+minds"	"Mozilla/1.22 (Windows; I; 32bit)"
24.10.81.100	–	–	[01/Aug/2000: 00:12:35 +0100]	"GET/news.cfm HTTP/1.0"	200	7719	"/index.cfm"	"Mozilla/4.0+(compatible;+MSIE+6.0; +Windows+NT+5.1;+Q312461)"

Say the web site topology is this A B C D E, A K L M N and a user browsing path is [A B C L]. Then it is assumed that page L is accessed by another user, because it is not directly accessible from the sequence of pages A B C.

Another example is when web site topology is like this: A B C D E K L M N. Say, the browsing path is [A B C D B C D], then it is assumed that two users contain this path as B was accessed twice. In the result the first users browsing path is assumed as [A B C D] and the second as [B C D].

Another condition by which a new user is assumed is when in a path of previously viewed pages it appears that the page has already been accessed. This occurrence is very limited and not accurate. It does not accept repeated pages in the same user's session which do not access through the cache.

3.2.5 User identification by the cookies

The most straightforward way to identify visitors is by using cookie information. When a user visits the web site for the first time, a cookie does not yet exist and no information is sent from the browser to the web server. The web server "understands" that it is a new user and passes unique cookie information to the browser together with the requested file. Then the web browser saves the cookie in a special directory.[11] Next time the visitor accesses the same web site, the browser sends the already existing cookie to the web server (if it is long lasting cookie). The web server recognises the cookie and thus does not send it again but passes just the requested page. This cookie can identify a user uniquely. *This is the best known user identification method*. The problem is that cookies are often not logged and other session identification techniques must be used.

The importance of cookies has received attention from the European Parliament (Roberts, 2002) as it allows monitoring users actions. Parliamentarians voted for allowing companies to use cookies to study online visitors' behaviour. However, because of the availability of methods to control cookies users sometimes disable them.

3.3 Session identification

When the user is identified the data needs to be broken into browsing sessions. The major problem here is to identify requests belonging to the same session. A session is assumed as a user's connection to the Internet but not necessary to a single web server. However this terminology is not very common and in most literature sources like Spiliopoulou (1999) a session is assumed as a sequence of requests made by a single user during a certain time period to the same web site. Typically analysis is performed from the server side as a user's requests to other servers are not available. Activity, when a user connects to one server is also known as "real sessions" (Berendt, Mobasher, Nakagawa, & Spiliopoulou, 2002). In some literature sources episodes are analysed. These are pages until *backward reference* is detected in a set of accessed pages (Wu, Yu, & Ballman, 1998).

[11] For Window XP and IE6 it is "C:\Documents and Settings\USERNAME\Cookies\"

Formally, sessions can be defined in the following way. One session S is a set of entries s made by the user while browsing some particular web site. Then session model can be interpreted as:

$$S = \langle s.ip, \{(s.wp_1, s.t_1), \ldots, (s.wp_n, s.t_n)\}\rangle$$

where $s \in S$ is one visitor's entry which contains $s.ip$ (IP address), $s.wp$ (web page) and $s.t$ (time of the entry), n (number of transactions in a session).

The analysis does not require knowledge of who is the user but it is important to distinguish every user from the rest. The task is not so simple due to the web site architecture and the way information about the requests is collected. Besides, there is no single solution on how users should be identified. Methodology applied to one type is not suitable for another web log data type. This is because different formats for handling web logs exist. Some web servers collect cookie information, some not. Some web sites require user authentication, others don't. There are two major ways to identify a session. One is to use login or cookies. In this case it is not a problem at all. The other way is the guessing way. Pages belonging to the same session are determined by time intervals between their being accessed, site topology and so on.

Spiliopoulou (1999) named two major strategies on how "real sessions" are identified. These are so called "proactive" and "reactive" methods. "Proactive" constructs sessions using sessions id gathered from cookies. The process is performed before or during the individual's interaction with the web site. "Proactive" strategies include user authentication, the activation of cookies that are installed on a user's machine. However, cookies raise a number of other issues. Firstly, there are those related to the privacy among users. Secondly, cookies are not stored into web logs, as not all web site developers and web server administrators understand the importance of collecting and storing complete web log information.

The second strategy, suggested by authors, is called "reactive". Here sessions are created from web logs. Because of the problems on how to identify a unique user, sessions can be inaccurate. Therefore, a huge amount of research is dedicated to "reactive" strategy. The following sections discuss various session identification methods in more detail.

3.3.1 Session identification by the time gap

The most popular session identification techniques use a time gap between the entries. If the time gap between two page requests made by the same user exceeds some threshold, a new session is created:

$$\text{if } s.t_{n+1} - s.t_n \geq time_{threshold} \text{ then new session}$$

Different threshold values can be found in research literature and usually vary from 10–15 min (He & Goker, 2003), 25–30 min (Pitkow & Margaret, 1994) to 1 (Paliouras, Papatheodorou, Karkaletsis, Spyropoulous, & Tzitziras, 1999) or even 2 h (Montgomery & Faloutsos, 2001). The most widely used time gap is 25.5 min calculated by Catledge and Pitkow empirically (Catledge & Pitkow, 1995). The authors calculated the mean inactivity time within a site. The typical value was found to be 9.3 min, 1.5 standard deviations were added to this typical inactivity time. Hence, the 25.5 min cut off for the duration of a visit was defined as standard inactivity time. Most commercial and free available applications use the rounded 30 min time gap.

Of course such time gap can vary depending on a number of reasons and could be determined dynamically. There is a lack of research works in this area. None of the research applied dynamic time gap.

3.3.2 Session identification by the duration spend observing page

Another session identification methodology based on time is proposed by Cooley et al. (1999b). The assumption is that depending on the duration (time spent on viewing this page), pages can be divided into two groups—navigational (or auxiliary) and information (or content) pages. Information pages are desired visitor destinations and the duration time on these pages is much longer than on navigational pages, which are passed by quickly. The duration time of navigational pages is respectively smaller. The next important step in this approach is to define the threshold between the duration time of navigational and information pages. If we could assume that the percentage of the navigational pages is known in the log file, the maximum length (if the length is longer, then it is an information page) of navigational pages is calculated as:

$$q = \frac{-\ln(1-\gamma)}{\lambda}$$

where, q is a threshold of navigational pages. This formula is derived from the exponential distribution of navigational and information pages (see (Cooley et al., 1999b) for distribution graphic), γ is the percentage of navigational pages (it is assumed that is known) and λ is the observed duration time mean of all pages in the log. Based on this study scientists agreed to restrict maximum time per page to 10 min (Faulstich & Spiliopoulou, 1998).

3.3.3 Session identification by the maximum forward references

A different approach is to identify sessions using maximal forward references presented in Chen et al. (1996). This approach converts the original sequence of log data into a path from the first to the last page before the result of button "back" action is processed. In other words, if in the sequence of unrepeated pages (called forward reference) there appears the page already found in a set (is called backward reference), a new session is defined starting from the next page. The last page before the backward reference page is called maximum forward reference. However this method doesn't seem to justify itself since button "back" in practice is clicked very often and actually is a part of the common browsing activity. Additionally, if a button back is clicked, it is often the case that a browser does not request previous page and uses one from the cache and this event is not reflected in the web log.

3.3.4 Session identification by a referrer

Session identification by the referrer described by Berendt et al. (2002) is based on the referred page. The referred page is available from the web data in extended log format (see 0 Extended common log format). Say there are two consecutive requests m and n, where m ə S (m is a page and S is a session). If referrer (r) for a page n was invoked within session S: r ə S, then n is added to S, otherwise to a new session. Experiments archived with sessions created using referrer page showed that recommendation systems work slightly better but just on certain structure web sites—without containing frames.

3.4 Transformation

Sometimes original raw format is not the most convenient form for data analysis and it would be advantageous to modify it prior to analysis (Hand, Mannila, & Smyth, 2001). The last data pre-processing step is data transformation, a.k.a. data formatting (Cooley et al., 1999b). This process involves final data preparation for the analysis (mining). This may include:

a. Final selection of necessary fields. For example, a time stamp is not required when applying association rules. In that case the time stamp is removed from data.
b. Creating additional and/or combined features. For example, it may be worth introducing duration (the time the user spends on viewing the page) to find some correlation between the page context and viewing duration.
c. Preparation related to the data mining model. For example, some models (e.g., neural networks) require (or recommend) transforming discrete data into vectors with binary attributes. The discrete attributes are replaced by the set of "0" and "1". Other techniques, however, accept discrete attributes.

4 Web log mining/analysis

Traditional data mining and statistical methods are not suitable for web log mining as it exists, and neither is web log data suitable for data mining algorithms. Therefore, data, algorithms and applications require specific preparations. This section discusses different applications of web log mining as well as methods used in here.

4.1 Mining aims and applications

There are several aims of web log mining. The most popular one is probably recommendations. Web log mining can be used for intelligent caching, customer retention, site structure improvement as well. These and other topics are discussed below.

4.1.1 Recommendations

Recommendation is the process when from the historical user's behaviour and current user's activity the new user is advised to enter certain pages or purchase certain products. In Cooley et al. (1999a) the authors presented a feedback system which constantly adapts to user behaviour and produces better recommendations for the following day. A number of data mining techniques were used in the proposed system amongst which are: clustering, frequent itemsets and association rules.

The system, which is based on gathering usage patterns on everyday data collection, is described in Balabanovic, Shoham, and Yun (1995). Firstly, the data is collected and analysed. Secondly, the system called LIRA is able to make recommendations for the following day. Recommendations consist of document selections which a system thinks users will find interesting. Accordingly, selected pages produced much better results than randomly selected pages.

One of the best illustrations of recommendations is Amazon's recommendation engine, where a user is informed that "Customers who bought this item also bought this" or "Customers who bought music by this artist also bought music by these artists."

4.1.2 Next step prediction

To improve a web site's performance and retention, an important issue is what pages are likely to be entered by a certain path and with a certain probability. The method to predict future requests using a path that contains the ordered list of URLs accessed by the users within a specified time constraint is described in Schechter, Krishnan, and Smith (1998). The given methodology predicts this with the high level of accuracy and therefore reduces the time the server spends generating or loading the page.

The problem with the huge web sites is that they produce too many models of existing combinations. In order to reduce the number of such cases, Pitkow and Pirolli utilized the longest repeating subsequence models without losing the ability to make accurate predictions of the next user step with Markov models (Pitkow & Pirolli, 1999).

Since the Markov chain model consists of a matrix of state probabilities, the Markov chains allow the system to dynamically model access patterns. Therefore the research in Sarukkai (2000) used Markov chains for link prediction and path analysis.

Web log analysis described in Pitkow and Recker (1994) is dedicated to improving web page caching on cache/proxy servers. The implemented algorithm is based on psychological human memory retrieval research. It collects past access patterns and predicts further user actions. The authors also noticed that recent document access rate is a stronger indicator for future document requests than frequency indicator. In other words, recently accessed documents are more important than old pages.

Other authors made an assumption that sequential patterns are more effective for prediction tasks and are better suitable for pre-fetching web pages (Mobasher, Dai, Luo, & Nakagawa, 2002). A framework using both association rules and sequential patterns for data analysis is presented in Jain, Han, Mobasher, & Srivastava (1996). The authors combine association rules and sequential patterns to improve the performance.

4.1.3 Behavioural analysis

It is important to understand a user's needs and to improve their browsing. The authors in Perkowitz and Etzioni (1997) challenge scientists to focus on creating adaptive web sites using modern artificial intelligence techniques. It means that web sites must automatically expand their administration and management by learning from user access patterns. To achieve this, site developers should concentrate on customization (modify web pages to suit users needs) and optimisation (to make navigation of the site easier). Perkowitz and Etzioni described algorithm PageGather which uses new cluster mining technique for indicating URLs sets for adaptive Web sites (Perkowitz & Etzioni, 1998, 1999). This semi-automatic algorithm improves the organization and presentation of the web site by learning from visitors behaviour.

A model that takes both customers travelling and purchasing patterns into consideration is described in Yun and Chen (2000a, b). Developed algorithms have extracted meaningful web transaction records and determined large transaction patterns.

4.1.4 Intelligent caching

Web logs to be deployed for intelligent web caching. If the next user step can be predicted with reasonable accuracy then the cache can pre-fetch it by improving response times (Bonchi et al., 2001a). This process is also illustrated in the following research works (Glassman, 1994; Luotonen & Altis, 1994).

4.2 Data mining methods

Most known data mining techniques described in this section can be applied to any data mining task. In this paper, these algorithms are presented from a web log mining perspective. Use of these algorithms in web log mining is illustrated by relevant literature sources.

4.2.1 Clustering

Clustering is suitable for huge datasets and is the first data mining technique to examine the structure and patterns in data (Berry & Linoff, 1997). For example, clustering might be the first step in consolidating perspective customers or visitors which contains similar browsing behaviour and characteristics into groups (Hand et al., 2001; Jain, Han, Mobasher, & Srivastava, 1997; Kanth & Siva, 2002; Xiao & Zhang, 2001). These groups of related clients are called clusters. For example, a group could be described by customers having income more than £25000 per year and age between 30 and 45. Clusters can be related page occurrences as well. For example a cluster can be composed of users browsing pages A, B, C from Mondays to Fridays. If there is a group of users behaving similarly it will be highlighted as a cluster.

The similarity between cluster objects (data samples) can be measured utilising any metric function but Euclidean distance is one of the most popular (Duda, Hart, & Stork, 2000).

Clustering techniques are widely used in web personalization. Personalisation means adoption to an individual user's needs. Usually personalisation withdraws historical data about similar visitors' behaviour and suggests individual navigation paths to those groups. Dai et al. utilized clustering to discover overlapping profiles which are later used by recommendation systems for real-time web site personalization (Dai, Luo, Mobasher, Nakagawa, & Witshire, 2000). Cooley et al. (1999a) used a full spectrum of data mining algorithms for web personalization based on transaction clustering and usage clustering. They proposed an approach for web personalization based on past users activities. This information is later used for online recommendations.

4.2.2 Classifications

Classification task allows developing a profile of pages accessed by certain user groups and thereby allowing this profile to be used in classifying new users according to their attributes (Han, Cai, & Cercone, 1993; Weiss & Kulikowski, 1991). In other words based on user browsing pattern and some additional information a user can automatically be assigned to one of the predefined classes (i.e., customer, guest). For example, potential buyers are users between age 25 and 40 and accessing the web site from the United Kingdom. Classification is usually performed using one of the well known techniques like neural networks or decision trees (Pabarskaite, 2003).

4.2.3 Associations

Due to an increasing number of transactions in real world datasets, fast and efficient association rules are becoming more and more popular. They are able to discover related items occurring together in the same transaction (or session in web log mining) (Kanth & Siva, 2002). Rules are very useful in web log mining and have already been applied in some research works, e.g., Kato, Hiraishi, and Mizoguchi (2001).

In practice it is impossible to test all possible combinations of items in transactions, as this number is huge. Therefore, association rule algorithms significantly reduce the number of combinations for examination. One of the most popular association rules algorithm's is Apriori, introduced by Agrawal, Imielinski, and Swami (1993) and Agrawal and Srikant (1994). The output of this algorithm produces human understandable rules of the form $X \Rightarrow Y$, where X and Y are sets of items—itemsets.

Apriori uses a lot of a priori information to reduce the number of combinations to be examined. The algorithm consists of two phases. Firstly, frequent (sometimes called large) itemsets (a.k.a. baskets) are generated. Next, these itemsets are used to generate rules. Frequent itemsets as well as rules are output of majority association rule algorithms. They are both useful and important.

However, the number of rules produced can be very large. For this reason, minimum support and minimum confidence are used. These measures determine the importance and quality of the rules. At the same time, the amount of results is reduced dramatically (to human acceptable amount) and speed is increased. As a matter of fact, some authors contradict such an approach of rule generation (Padmanabhan & Tuzhilin, 2000; Piatetsky-Shapiro & Matheus, 1994; Silberschatz & Tuzhilin, 1995). They argue that subjective measures such as unexpectedness and actionability and finding interesting patterns depend on the decision maker and not solely on the statistical measure (Adomavicius, 1997). The algorithm of discovering unexpected rules is presented in Padmanabhan and Tuzhilin (1999).

The most applicable association rules area in the retail sector where sets of items purchased together are computed. However association rules are successfully used in web usage mining as well. Apriori algorithm has been used in a number of research works (Cooley et al., 1997a, b; Cooley et al., 1999a; Fong, Hughes, & Zhu, 2000; Han et al., 2000; Jain et al., 1997).

Example 1 Denote S_i as a session, where i is session's identification number. Say there are number of sessions made by the same user:

```
S1 ("Products.html","Software.html");
S2 ("Products.html","Software.html","Hardware.html","Services.html");
S3 ("Software.html","Hardware.html","Services.html");
S4 ("Products.html","Software.html","Hardware.html","Contacts.html").
```

The first session S_1 includes two visited pages "Products.html" and "Software.html", the second contains four pages etc. From the data above the following rule "IF Software.html and Hardware.html THEN Services.html" can be derived with the support s=2/4=50% (the number of such transactions occurring across all transactions) and confidence c=2/3=66% (transactions on the first part of the rule supporting the rule divided by transactions on the first part which do not necessary support the rule).

Yang, Wang, & Zhang (2002) propose to extend the traditional association rules paradigm by imposing new temporal information and the confidence of each rule in relation to the prediction if certain page in a certain time moment will occur.

4.2.4 Sequential rules

Another group of rules is based on sequential items. These items follow consecutively in the time ordered set (Agrawal & Srikant, 1995; Mannila, Toivonen, & Verkamo, 1995; Pirjo, 2000).

Definition 1 *Let say I is a set of items. A transaction is a pair (i, t), where i ∈ I is a type of the item and t ∈ T, t is the time stamp of the event. Then itemset S is an ordered sequence of transactions, e.g.,:*

$$S = \langle (i_1, t_1), (i_2, t_2), \ldots, (i_k, t_k), \ldots, (i_m, t_m) \rangle$$

where $i_k \in I$ for all k= 1, ..., m and $t_k \leq t_{k+1}$ for all k= 1, ..., m−1. The length of the sequence S is denoted as $|S| = m$.

An empty sequence of items is denoted as $S = \langle\ \rangle$.

The problem of mining sequential patterns is to find the maximal frequent sequences among all sequences that have a certain user specified minimum support (see for details Srikant & Agrawal, 1996).

Definition 2 *The support is defined as a percentage of patterns among all data containing the sequence S. Each maximal sequence represents a sequential pattern if supp(S) ≥ minsupp, when minsupp is defined by the analyst and S is considered as a frequent sequential pattern.*

Example 2 Requested web pages may represent a sequential pattern. For example, 45% of users accessed /Jobs_Online which was followed by the page /About_company and /Careers.

Standard algorithms usually discover only frequent sequential rules. This limits the ability to detect rare but still valuable web usage patterns. Spiliopoulou proposed to extend sequential mining not merely identifying frequent itemsets but also by discovering "interesting" rules (Spiliopoulou, 1999). The author developed PostMine algorithm. It transforms frequent sequences of accessed pages into a set of sequential rules and then filters this set using various statistical measures and heuristics.

4.2.5 OLAP, queries, data warehousing

One of the ways to analyse web log data is to load it into the database or data warehouse and run analytical queries against it. It has proved to be a good method of data analysis. However it is quite costly as this is a manual process and an analyst is required to design and run queries.

Online analytical Processing (OLAP) is the way to present relational data to facilitate understanding of the data and important patterns inside it (Berry & Linoff, 1997). In other words it is the way to transform data into valuable information. OLAP is known to be a valuable tool in business intelligence and decision support systems but can successfully be used for web log mining as well.

A systematic study about web log data warehousing development is presented in Han et al. (1998). The authors developed a tool called WebLogMiner (Han et al., 1997). The tool contains OLAP engine to characterise and examine data, visualise associations, predict attribute values, construct models for each given class based upon web log data features, produce time-series analysis, etc.

4.3 Enhancing web log mining with additional information

It is obvious, that more information can lead to better results of analytical process. If there is some additional information which can be linked to web log data and this information is relevant to a problem, then this can significantly improve results. Two types of additional information are discussed below.

4.3.1 Web log and customer data mining

If the web site requires login and a web site owner has user personal information, then it opens entirely new horizons for analysis. Web usage and additional information can be combined together and interesting patterns can be discovered. For example—browsing patterns may depend on age. So that automatically allows sending users into different web site parts depending on their age. Additional data about customers can be obtained in a different way. One such methods is to collect information about geographical user location at the time of registration (Kanth & Siva, 2002).

In addition, that allows improving quality of any techniques applied as more information leads to more accurate results.

4.3.2 Web log and web content mining

To improve personalisation tasks scientists incorporated text from web pages and used it together with web log data (Dai & Mobasher, 2003; Mobasher, Dai, Luo, Sung, & Zhu, 2000). In Pabarskaite and Raudys (2002) we proposed to use retrieved texts from HTML link tags. The most frequently used words from the text tags were selected and mapped into fixed length vectors as additional features. Then experiments were performed using data containing only log data. The second round of experiments included only text information from HTML link tags. The third round of experiments contained both web browsing (log) and text information. The last round provided the best results and confirmed the correctness of the hypothesis. The quality of results increases when text tags are involved in the mining process. This discovery might be important when a small amount of data is available or when the web site changes constantly. Moreover, it appears that when using context data it is possible to distinguish user groups with different interests even thought their navigational patterns match. Besides, this combined approach plays an important role in recommendation systems since it relies on the existing—historical (user navigational data) and hence items added recently into a web site can not be recommended.

Another approach is to utilize semantic knowledge (context features associated with the items such as keywords, phrases, category names or other textual information) and ontology (takes domain information as an input and usually contains data of the objects: links, categories and their relationships) for web usage mining and web personalization (Dai & Mobasher, 2003).

4.4 Large scale web log mining

Sometimes datasets sizes can be huge and there is a high risk that they won't fit into an existing memory. Therefore researchers are working on better/compact ways to represent such data in memory. The authors in Xiao and Dunham (2001) investigated very large web logs. There was no way of loading all this data in memory because loads quantities of records were duplicated. An effective suffix tree algorithm was presented which compressed and pruned the database and reduced the main memory usage.

5 Visualisation and results

Once data analysis techniques have been applied, it is necessary to make the most out of results. It is very common that various data analysis techniques produce extensive reports

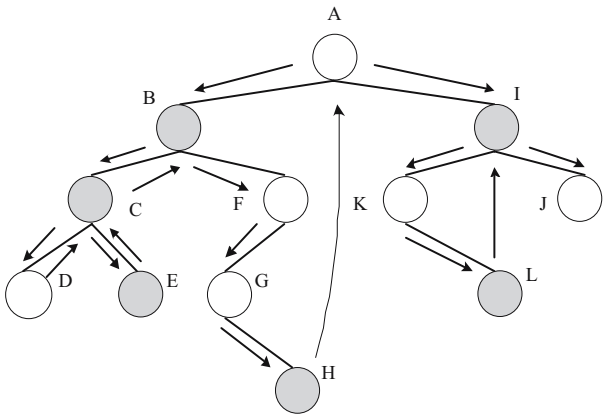

Fig. 6 Representing web usage using graphs.

which are not always interpreted in the right way and can not be used efficiently (Chi, 2002). One of the reasons could be that humans are very good at identifying patterns from visualized data (Ansari et al., 2001) but not from text. Therefore different result visualization techniques are extensively used to present web stats and user behaviour patters.

5.1 Simple statistics

Most of statistical web log analysis software uses text based reports ant bar/line based chats for result presentation. Some tools use 3D chats. MS Excel and Matlab are good examples of tools for spreadsheet data visualisation. In most cases 2D data is enough to represent two dimensions data because two dimensional plots are well interpreted by humans and therefore suitable to present various statistics. Typical information of these kinds of techniques is web site usage (number of hits) by time, number of hits from various domains like edu, com etc.

There are lacks of research works involving innovative visualisation of web log statistical information. For example, Wein diagrams[12] could be employed here successfully.

5.2 Web site topology

Another way representing web usage results is using graphs. Pages correspond to nodes and links to edges (Pitkow & Bharat, 1994; Dyreson, 1997; Consens et al., 1994), see Fig. 6. The technique is very effective, as it is easy to understand an overall structure of the web site. Representation through nodes and edges allows the exploration of a web site's hierarchical structure and identifies typical usage paths. Such graph implementation allows the observance of graphical information about accessed documents and the URL paths through the site. It enables the selection of users to be filtered in the access logs (restrict view by features such as domain names or IP numbers, directory names, time), control

[12] http://www.venndiagram.com/

graphs attributes (node size, colour and etc.) and events in the access log by "playing back" possibility.

5.3 On-line analytic processing

(OLAP) is a way to present data in multidimensional view for the end user. OLAP provides the business analyst with a million spreadsheets at a time available in a logical and hierarchical structure. The analysis can go to higher or deeper levels to look at the data from different perspectives (Peterson & Pinkelman, 2000). Scientists encourage using OLAP as applicable for analysis and as visualisation tool mining web logs (Dyreson, 1997). Now almost any data mining and data warehousing tool provides OLAP capabilities, e.g., MS SQL OLAP.

6 Web log mining software

As web log mining is evolving over time, more and more web log mining software is appearing on the market. Some of those software packages are quite simple and do just stats, others are more sophisticated and can do more advanced analysis. For discussion on commercial and free web log mining/analysis software refer to the following sections.

6.1 Commercial

A huge number of commercial web mining techniques are presented on kdnuggets web site (http://www.kdnuggets.com/software/web.html). However this list is more oriented to web mining, text mining and only a small part is about web log mining. Hundreds of other tools can be found on the web. Among available software packages it is worth mentioning: Accrue Software, Autonomy systems, Amadea, Angoss, BlueMartini, Blossom software, Clementine, Quadstone, Data Mining Suite, OK-log, Lumio, Megaputer WebAnalyst, MicroStrategy, net.Analysis, NetTracker, Prudsys, SAS Webhound, Torrent WebHouse, Webtrends, XAffinity(TM), XML Miner, 123LogAnalyser, Caesius Software, Funel.

6.2 Free/academic/GNU

There are two known software packages. WUM has been developed by a scientist at Berlin University, see also papers of Faulstiche, Pohle et al. (1999) and Faulstich, Spiliopoulou et al. (1999). The tool applies sequence mining to analyse users navigational behaviour. They also implemented a special querying language to retrieve sequential patterns. The disadvantage is that it is limited in features it supports and is comparatively slow, since it is implemented using Java.

Other web log analysis software "Analog"[13] has not been tested by the authors, but it claims to be very fast in analysing huge web log files.

[13] http://www.analog.cx/, Analog. 2003

7 Conclusions

Over the last decade the web has been moving into every aspect of human professional and social life. Thus analysing user behaviour in web site navigation has become more and more popular. This analysis is very useful as it allows adopting of web pages for user needs, detecting valuable customers, increasing customer retention and so on. This paper has surveyed a number of research papers in this area, thereby showing capabilities of web usage analysis and noting existing techniques and methods.

The paper surveys the whole web log mining KDD process starting from data collection, pre-processing, mining and finishing with result analysis and visualization. Strengths and weaknesses are highlighted in these areas and possible solutions are discussed. The following standpoints can be concluded.

(1) Web log data pre-processing is probably the most complex and important stage step in web log mining and consumes about half of the time spent on the whole process. It requires additional research to create confident and efficient data preparation methods. In many cases, there is no straightforward answer about which technique to use since exceptions always exist. Pre-processing and cleaning methods depend on the site structure, log format and fields available.
(2) Data cleaning (removing irrelevant records) is strongly related to the domain and web page design. Analysis shows that many researches do not clean web log data properly and therefore results are inaccurate. The list of files to clean can vary from site to site and should change dynamically. Data cleaning methods must be researched further.
(3) The survey shows that web log mining is no different to any other transactional data mining if data is cleaned and pre-processed properly. The main difference is the application as web log mining results is quite often in the form of recommendations which is not available in retail or other transactional data analysis.
(4) A brief review of web usage visualisation techniques is provided. The main problem is that research scientists use web links for the presentation of results and statistics. More user friendly techniques should be considered. For example, links can be replaced by text tags on links.

References

Adomavicius, G. (1997). Discovery of actionable patterns in databases: The action hierarchy approach. *Knowledge discovery and data mining*. Newport Beach, CA, Menlo Park, CA.

Agrawal, R., Imielinski, T., & Swami, S. (1993). Mining association rules between sets in large databases. *Conference on Management of Data (ACM SIGMOD)*, Washington, DC.

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules. *Proceedings of the 20th VLDB*, Santiago, Chile.

Agrawal, R., & Srikant R. (1995). Mining sequential patterns. *Data engineering*. Taipei: IEEE.

Ansari, S., Kohavi, R., Mason, L., & Zheng, Z. (2001). Integrating e-commerce and data mining: Architecture and challenges. *Data mining*. San Jose, CA: IEEE Computer Society.

Balabanovic, M., Shoham, Y., & Yun, Y. (1995). An adaptive agent for automated web browsing. *Visual Communication and Image Representation*, 4.

Barish, G., & Obraczka, K. (2000). World wide web caching: Trends and techniques. *IEEE Communications Magazine, 5*, 178–184.

Berendt, B., Mobasher, B., Nakagawa, M., & Spiliopoulou, M. (2002). The impact of site structure and user environment on session reconstruction in web usage analysis. *4th WebKDD 2002 Workshop, ACM-SIGKDD Conference on Knowledge Discovery in Databases (KDD'2002)*, Edmonton, Alberta, Canada.

Berendt, B., & Spiliopoulou, M. (2000). Analysis of navigation behaviour in web sites integrating multiple information systems. *VLDB*, 56–75.

Berry, M. J. A., & Linoff, G. (1997). *Data mining techniques: For marketing, sales, and customer support*. New York: Wiley.

Bonchi, F., Giannotti, F., Gozzi, C., Manco, G., Nanni, M., Pedreschi, D. et al. (2001a). *Web log data warehousing and mining for intelligent web caching*. Elsevier Science.

Bonchi, F., Giannotti, F., Gozzi, C., Manco, G., Nanni, M., Pedreschi, D., et al. (2001b). Web log data warehousing and mining for intelligent web caching. *Data and Knowledge Engineering, 2*, 165–189.

Buchner, A. G., Mulvenna, M. D., Anand, S. S., & Hughes, J. G. (1999). An Internet-enabled knowledge discovery process. *International database conference; Heterogeneous and internet databases*. Hong Kong: City University of Hong Kong.

Catledge, L. D., & Pitkow, J. E. (1995). Characterizing browsing strategies in the world-wide web. *Computer Networks and ISDN Systems, 6*, 10–65.

Chakrabarti, S. (2000). Data mining for hypertext: A tutorial survey. *ACM SIGKDD Explorations, 2*, 1–11.

Chen, M. S., Park, J. S., & Yu, P. S. (1996). Data mining for path traversal patterns in a web environment. *Distributed computing systems*. Hong Kong: IEEE Computer Society.

Chi, E. H. (2002). Improving web usability through visualization. *IEEE Internet Computing*, 64–71.

Consens, M. P., Eigler, F. C., Hasan, M. Z., Mendelzon, A. O., Noik, E. G., Ryman, A. G., et al. (1994). Architecture and applications of the Hy^+ visualization system. *IBM Systems Journal, 3*, 458.

Cooley, R., Mobasher, B., & Srivastava, J. (1997a). *Grouping web page references into transactions for mining world wide web browsing patterns.IEEE Knowledge and Data Engineering Exchange Workshop (KDEX'97)*. Los Alamitos, CA: IEEE Computer Society.

Cooley, R., Mobasher, B., & Srivastava, J. (1997b). Web mining: Information and pattern discovery on the word wide web. *9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97)*.

Cooley, R., Mobasher, B., & Srivastava, J. (1999a). *Automatic personalization based on web usage mining*. Chicago, IL: Depaul University.

Cooley, R., Mobasher, B., & Srivastava, J. (1999b). Data preparation for mining world wide web browsing patterns. *Knowledge and Information Systems, 1*, 5–32.

Dai, H., Luo, T., Mobasher, B., Nakagawa, M., & Witshire, J. (2000). Discovery of aggregate usage profiles for web personalization. *Mining for E-Commerce Workshop (WebKDD'2000, held in conjunction with the ACM-SIGKDD on Knowledge Discovery in Databases KDD'2000)*, Boston, MA.

Dai, H., & Mobasher, B. (2003). A road map to more effective web personalization: Integrating domain knowledge with web usage mining. *Proceedings of the International Conference on Internet Computing (IC'03)*, Las Vegas, NV.

Davison, B. (1999). A survey of proxy cache evaluation techniques. *4th International Web Caching Workshop (WCW'99)*.

Duda, R. O., Hart, P. E., & Stork, D. G. (2000). *Pattern classification*. New York: Wiley.

Duska, B. M., Marwood, D., & Freeley, M. J. (1997). The measured access characteristics of world-wide-web client proxy caches. *Proceedings of the USENIX Symposium on Internet Technologies and Systems*, Monterey, CA: USENIX Association.

Dyreson, C. (1997). Using an incomplete data cube as a summary data sieve. *Bulletin of the IEEE Technical Committee on Data Engineering* (March): 19–26.

Elo-Dean, S., & Viveros M. (1997). *Data mining the IBM official 1996 Olympics Web site*, IBM T.J. Watson Research Center.

Famili, A., Shen, W. M., Weber, R., & Simoudis, E. (1997). Data preprocessing and intelligent data analysis. *Intelligent Data Analysis*, 3–23.

Faulstich, L. C., & Spiliopoulou, M. (1998). WUM: A tool for web utilization analysis. *EDBT Workshop WebDB'98*. Valencia, Spain: Springer-Verlag.

Faulstich, L., Pohle, C., & Spiliopoulou, M. (1999). Improving the effectiveness of a web site with web usage mining. *KDD Workshop WEBKDD'99*, San Diego, CA.

Faulstich, L., Spiliopoulou, M., & Wilkler, K. (1999). A data mining analyzing the navigational behaviour of web users. *Workshop on Machine Learning User Modeling of the ACAI'99 International Conference*, Creta, Greece.

Fayyad, U. M. (1996). *Advances in knowledge discovery and data mining*. Menlo Park, CA: AAAI Press.

Fleishman, G. (1996). *Web log analysis, who's doing what, when*? Web Developer.

Fong, J., Hughes, J. G., & Zhu, J. (2000). *Online web mining transactions association rules using frame metadata model*.

Glassman, S. (1994). A caching relay for the world wide web. *1st World Wide Web Conference*. Geneva, Switzerland: Elsevier.

Han, J., Cai, Y., & Cercone, N. (1993). Date-driven discovery of quantitative rules in relational databases. *IEEE Transactions on Knowledge and Data Engineering*, 29–40.

Han, J., Chiang, J., Chee, S., Chen, J., Chen, Q., Cheng, S., et al. (1997). DBMiner: A system for data mining in relational databases and data warehouses. *CASCON'97: Meeting of Minds*, Toronto, Canada.

Han, J., He, Y., & Wang, K. (2000). Mining frequent itemsets using support constraints. *International Conference on Very Large Databases (VLDB'00)*, Cairo, Egypt.

Han, J., Xin, M., & Zaïane, O. R. (1998). Discovering web access patterns and trends by applying OLAP and data mining technology on web logs. *Conference on Advances in Digital Libraries*, Santa Barbara, CA.

Hand, D., Mannila, H., & Smyth, P. (2001). *Principles of data mining*.

He, D., & Goker, A. (2003). *Detecting session boundaries from Web user logs*.

Jain, N., Han, E., Mobasher, B., & Srivastava, J. (1996). *Web mining: Pattern discovery from world wide web transactions*. Minneapolis, MN: University of Minnesota.

Jain, N., Han, E., Mobasher, B., & Srivastava, J. (1997). *Web mining: Pattern discovery from world wide web transactions*. Minneapolis, MN: University of Minnesota.

Kanth, K. V. R., & Siva, R. (2002). Personalization and location-based technologies for e-commerce applications, eJETA.

Kato, H., Hiraishi, H., & Mizoguchi, F. (2001). Log summarizing agent for web access data using data mining techniques. *IEEE Intelligent Systems and Their Applications*, 2642–2647.

Kosala, R., & Blockeel, H. (2000). Web mining research: A survey. *ACM SIGKDD Explorations, 1*, 1–15.

Lin, I. Y., Huang, X. M., & Chen, M. S. (1999). Capturing user access patterns in the web for data mining. *Tools with artificial intelligence*. Chicago, IL: IEEE Computer Society.

Luotonen, A., & Altis, K. (1994). World-wide web proxies. *Selected Papers of First World-Wide Web Conference*, Elsevier Science Division. 147.

Lup Low, W., Li Lee, M., & Wang Ling, T. (2001). A knowledge-based approach for duplicate elimination in data cleaning. *Information Systems, 8*, 585–606.

Mannila, H., Toivonen, H., & Verkamo, A. I. (1995). Discovering frequent episodes in sequences. *Knowledge discovery & data mining*. Montreal, Canada: AAAI Press.

Mobasher, B., Dai, H., Luo, T., Sung, Y., & Zhu, J. (2000). Integrating web usage and content mining for more effective personalization. *Proceedings of the International Conference on E-Commerce and Web Technologies (ECWeb2000)*, Greenwich, UK.

Mobasher, B., Dai, H., Luo, T., & Nakagawa, M. (2002). Using sequential and non-sequential patterns in predictive web usage mining tasks. *International Conference on Data Mining*. Maebashi City, Japan: IEEE Computer Society.

Montgomery, A. L., & Faloutsos, C. (2001). Identifying web browsing trends and patterns. *Computer, 7*, 94–95.

Pabarskaite, Z. (2002). Implementing advanced cleaning and end-user interpretability technologies in web log mining. *Information Technology Interfaces ITI2002, Collaboration and Interaction in Knowledge-Based Environments*, Cavtat/Dubrovnik, Croatia.

Pabarskaite, Z. (2003). Decision trees for web log mining. *Intelligent Data Analysis, 2*, 141–155.

Pabarskaite, Z., & Raudys, A. (2002). Advances in web usage mining. *The 6th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2002)*, Florida, USA.

Padmanabhan, B., & Tuzhilin, A. (1999). Unexpectedness as a measure of interestingness in knowledge discovery. *Decision Support Systems*, 303–318.

Padmanabhan, B., & Tuzhilin, A. (2000). Small is beautiful: Discovering the minimal set of unexpected patterns. *International Conference on Knowledge Discovery and Data Mining; KDD 2000*. Boston, MA: Association for Computing Machinery.

Paliouras, G., Papatheodorou, C., Karkaletsis, V., Spyropoulous, C., & Tzitziras, P. (1999). From web usage statistics to web usage analysis. *IEEE International Conference on Systems Man and Cybernetics*: II-159–II-164.

Perkowitz, M., & Etzioni, O. (1997). Adaptive web sites: An AI challenge. *Proceedings IJCAI'97*, Nagoya, Japan.

Perkowitz, M., & Etzioni, O. (1998). Adaptive web sites: Automatically synthesizing web pages. *AAA/98*.

Perkowitz, M., & Etzioni, O. (1999). Towards adaptive web sites: Conceptual framework and case study. *Eighth International World Wide Web Conference*, Toronto, Ontario.

Peterson, T., & Pinkelman, J. (2000). *Microsoft OLAP Unleashed*.

Piatetsky-Shapiro, G., & Matheus, C. J. (1994). *The interestingness of deviations. Knowledge discovery in databases*. Seattle, WA: AAAI Press.

Pirjo, M. (2000). Attribute, event sequence, and event type similarity notions for data mining. *Department of Computer Science*, 199.

Pirolli, P., Pitkow, J., & Rao, R. (1996). Silk from a sow's ear: Extracting usable structure from the web. *Human factors in computing systems: Common ground; CHI 96*, Vancouver; Canada, New York.

Pitkow, J. (1997). In search of reliable usage data on the WWW. *The Sixth International World Wide Web Conference*, Santa Clara, CA.

Pitkow, J., & Bharat, K. (1994). WEBVIZ: A tool for world-wide web access log analysis. *First International World Wide Web Conference*, CERN, Geneva, Switzerland.

Pitkow, J., & Margaret, R. (1994). Integrating bottom-up and top-down analysis for intelligent hypertext. *Intelligent Knowledge Management*.

Pitkow, J., & Pirolli, P. (1999). Mining longest repeating subsequences to predict world wide web surfing. *Internet Technologies and Systems; USENIX Symposium on Internet Technologies and Systems*. Boulder, CO: USENIX Association.

Pitkow, J., & Recker, M. (1994). A simple yet robust caching algorithm based on dynamic access patterns. *First International World Wide Web Conference*, CERN, Geneva, Switzerland.

Roberts, S. (2002). Users are still wary of cookies. *Computer Weekly*, 24.

Sarukkai, R. R. (2000). Link prediction and path analysis using Markov chains. *Computer Networks*, 377–386.

Savola, T., Brown, M., Jung, J., Brandon, B., Meegan, R., Murphy, K., et al. (1996). *Using HTML*.

Schechter, S., Krishnan, M., & Smith, M. D. (1998). Using path profiles to predict HTTP requests. *Computer Networks and ISDN Systems* (1–7), 457–467.

Schmitt, E., Manning, H., Paul, Y., & Tong, J. (November, 1999). Measuring web successs. *Forrester Report*.

Shahabi, C., Zarkesh, A., Adibi, J., & Shah, V. (1997). Knowledge discovery from users Webpage navigation. *Research Issues in Data Engineering*, Birmingham, England.

Silberschatz, A., & Tuzhilin, A. (1995). On subjective measures of interestingness in knowledge discovery. *Knowledge discovery & data mining*. Montreal, Canada: AAAI Press.

Spiliopoulou, M. (1999). Managing interesting rules in sequence mining. *3rd European Conference on Principles and Practice of Knowledge Discovery in Databases PKDD'99*. Prague, Czech Republic: Springer-Verlag.

Srikant, R., & Agrawal, R. (1996). Mining sequential patterns: Generalizations and performance improvements. *Extending database technology.* Avignon, France: Springer.

Tan, P., & Kumar, V. (2002). Discovery of web robot sessions based on their navigational patterns. *Data Mining and Knowledge Discovery*, 9–35.

Tauscher, L., & Greenberg, S. (1997). How people revisit web pages: Empirical findings and implications for the design of history systems. *International Journal of Human Computer Studies, 1*, 97–138.

Wang, J. (1999). A survey of web caching schemes for the Internet. *Computer Communication Review, 5*, 36–46.

Weiss, S. M., & Kulikowski, C. A. (1991). *Computer systems that learn: Classification and prediction methods from statistics, neural nets, machine learning, and expert systems*. Morgan Kaufmann.

Wu, K.-L., Yu, P. S., & Ballman, A. (1998). SpeedTracer: A web usage mining and analysis tool. *IBM Systems Journal, 37*, 89–105.

Xiao, Y., & Dunham, M. H. (2001). Efficient mining of traversal patterns. *Data & Knowledge Engineering*, 191–214.

Xiao, J., & Zhang, Y. (2001). Clustering of web users using session-based similarity measures. *IEEE*.

Yang, Q., Wang, H., Zhang, W. (2002). Web-log mining for quantitative temporal-event prediction. *IEEE Computational Intelligence Bulletin, 1*, 10–18.

Yun, C. H., & Chen, M. S. (2000a). Using pattern–join and purchase–combination for mining web transaction patterns in an electronic commerce environment. *Compsac*, 99–104.

Yun, C.-H., & Chen, M.-S. (2000b). Mining web transaction patterns in an electronic commerce environment. *4th Pacific-Asia Conference on Knowledge Discovery and Data Mining*.