A Tool for Web Usage Mining

Jose M. Domenech[1] and Javier Lorenzo[2]

[1] Hospital Juan Carlos I
Real del Castillo 152 - 3571 Las Palmas - Spain
`jdomcab@gobiernodecanarias.org`
[2] Inst. of Intelligent Systems and Num. Applic. in Engineering
Univ. of Las Palmas
Campus Univ. de Tafira - 35017 Las Palmas - Spain
`jlorenzo@iusiani.ulpgc.es`

Abstract. This paper presents a tool for web usage mining. The aim is centered on providing a tool that facilitates the mining process rather than implement elaborated algorithms and techniques. The tool covers different phases of the CRISP-DM methodology as data preparation, data selection, modeling and evaluation. The algorithms used in the modeling phase are those implemented in the Weka project. The tool has been tested in a web site to find access and navigation patterns.

1 Introduction

Discovering knowledge from large databases has received great attention during the last decade being the data mining the main tool to make it [1]. The world wide web has been considered as the largest repository of information but it lacks of a well defined structure. Thus the world wide web is a good environment to make data mining receiving the name of Web Mining [2, 3].

Web mining can be divided into three main topics: Content Mining, Structure Mining and Usage Mining. This work is focused on Web Usage Mining (WUM) that has been defined as "the application of data mining techniques to discover usage patterns from Web data" [4]. Web usage mining can provide patterns of usage to the organizations in order to obtain customer profiles and therefore they can make easier the website browsing or present specific products/pages. The latter has a great interest for businesses because it can increase the sales if they offer only appealing products to the customers although as pointed out Anand (Anand et al, 2004), it is difficult to present a convincing case for Return on Investment. The success of data mining applications, as many other applications, depend on the development of a standard. CRISP-DM, (Standard Cross-Industry Process for Data Mining) (CRISP-DM, 2000) is a consortium of companies that has defined and validated a data mining process that can be used into different data mining projects as web usage mining. The life cycle of a data mining project is defined by CRISP-DM into 6 stages: Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation and Deployment.

The Business Understanding phase is highly connected with the problem to be solved because they defined the business objectives of the application. The last

one, Deployment, is not easy to make automatically because each organization has its own information processing management. For the rest of stages a tool can be designed in order to facilitate the work of web usage mining practitioners and reduce the development of new applications.

In this work we implement the WEBMINER architecture [5] which divides the WUM process into three main parts: preprocessing, pattern discovery and pattern analysis. This three parts corresponds to the data preparation, modeling and evaluation of the CRISP-DM model.

In this paper we present a tool to facilitate the Web Usage Mining based on the WEBMINER architecture. The tool is conceived as a framework where different techniques can be used in each stage facilitating in this way the experimentation and thus eliminating the need of programming the whole application when we are interested in studying the effect of a new method in the mining process. The architecture of the tool is shown in Figure 1 and the different elements that makes up it will be described. Thus, the paper is organized as follows. Section 2 will describe the data preprocessing. In sections 3 and 5 different approaches to user session and transactions identification will be presented. Finally in sections 6 and 7 the models to be generate and the results are presented.
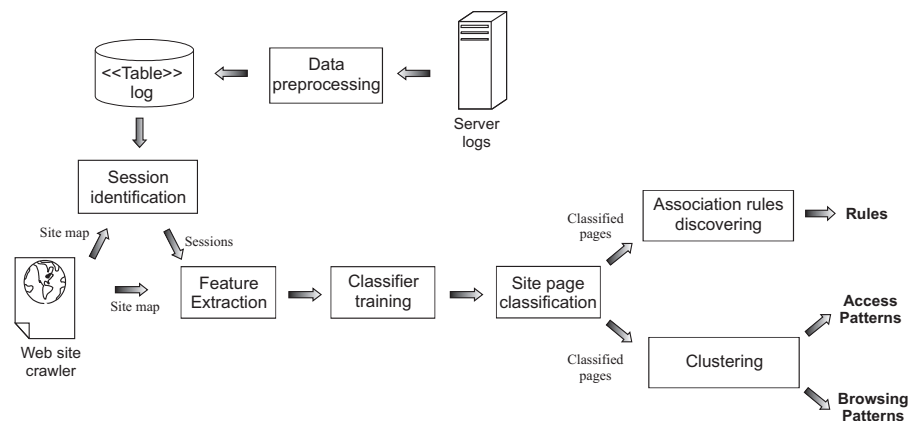

Fig. 1. WUM tool architecture

2 Web Log Processing

Data source for Web Usage Mining come from different sources as proxy, web log files, web site structure and even from sniffer packet logs. Normally, the most widely used sources are the web log files. These files record the user accesses to the site and there exists several formats: NCSA (Common Log Format), W3C Extended, SunTM ONE Web Server (iPlanet), IBM Tivoli Access Manager

WebSEAL or WebSphere Application Server Logs. The most of the web servers record the access using an extension of the CLF (ECLF). In ECLF basically the recorded information for each access is:

- *remote host*: Remote hostname. (or IP address number if DNS hostname is not available or was not provided)
- *rfc931*: The remote login name of the user. (If not available a minus sign is typically placed in the field)
- *authuser*: The username as which the user has authenticated himself. This is available when using password protected WWW pages. (If not available a minus sign is typically placed in the field)
- *date*: Date and time of the request.
- *request*: The request line exactly as it came from the client. (i.e., the file name, and the method used to retrieve it [typically GET])
- *status*: The HTTP response code returned to the client. Indicates whether or not the file was successfully retrieved, and if not, what error message was returned.
- *bytes*: The number of bytes transferred.
- *referer*: The url the client was on before requesting your url. (If it could not be determined a minus sign will be placed in this field)
- *user agent*: The software the client claims to be using. (If it could not be determined a minus sign will be placed in this field)

As said before, web server logs record all the user accesses including for each visited page all the elements that composed it as gif images, styles or scripts. Other entries in the log refers to fail requests to the server as "404 Error: Object not found". So a first phase in data preparation consists of filtering the log entries removing all useless entries. Others entries in the web log that must be removed, are those that correspond to search robots because they do not corresponds to a "true" user. To filter these entries it can be used the plain text file `Robot.txt`, the list of known search robots `www.robotstxt.org/wc/active/all.txt` and we have introduced an heuristic that is to filter those very quick consecutive requests because a characteristic of search robots is the short delay between page requests. So with a threshold of 2 seconds between two consecutive requests the entries that corresponds to robots can be eliminated.

The structure of the site has been used as another data source. This structure is obtained with a web crawler starting from the root, so all the pages that can be reached from the root will composed the structure of it. For non static sites the structure must be introduced by hand.

3   User Session Identification

Once the web log file is processed and all the irrelevant entries has been removed, it is necessary to identify the users that visit to the site. The visits are concurrent so in the log file the entries of different users are interlaced what makes us process it to collect the entries that belong to the same user.

A first approach to identify a user is to use the IP address and assign all the entries with the same IP to the same user. This approach exhibits some drawbacks. Some users access to internet through a proxy so many users will share the same IP. In other cases the same user has different IP because it has a dynamic IP configuration in its ISP. In order to minimize these effects some heuristics has been applied. A first heuristic is to detect changes in the browser or in the operative system fields of the entries that come from the same IP. Another heuristic makes use of the referer field and the map of the site obtained with the site crawler mentioned previously. Thus if a page is not directly linked to the pages previously visited by the user, it is an evidence that another user share the same IP and browser. With the explained heuristics we will get false positive, that is to consider only one user when actually are different users.

After identifying the users that have visited the site, the next phase is to obtain the user sessions. A session is made up of all the visited pages by a user. The technique is based on establishing a time threshold, so if two accesses take more than the fixed time threshold, it is considered as a new session [6, 7]. Many commercial products establish a threshold of 30 minutes. Catledge and Pitkow [8] define this threshold in 25.5 minutes based on empirical studies.

4   Web Page Classification

After data cleaning, the next stage in the data preparation phase is to compute the features of each page in the site. The following features has been defined:

- *Size*: Size of the page in bytes.
- *Num. incoming links*.
- *Num. outcoming links*.
- *Frequency*: Number of times the page was requested in a period of time.
- *Source*: Number of times the page is the starting point in a session.
- *Similarity*: Similarity of a page with its sibling pages based on a tbd computation.
- *Depth*: Average depth of the sibling pages. The depth of a pages is measured as the number of '/' in the URL.

From the previous features it can be obtained a model for diferent pages which avoid to the webmaster to annotate each of the page in the site. In this work we have defined the following pages of interest:

- *Home page*: It is the first visited page by the users.
- *Content page*: It contains a part of the site information.
- *Auxiliary page*: Users can use this page to visit other pages in the site.
- *Reference page*: Explain a concept or it has references
- *Personal page*: It contains biographic information of the organization staff.

To avoid the computational cost of training a classifier with the whole set of features, a previous feature selection stage is made. The initial feature set is

filtered using the GD measure [9], which is based on information theory concepts, in order to select the most informative features. This measure allows to rank the features according to the relevance with the concept and it also detects redundant features that can be removed from the initial feature set.

In a small web site, pages can be manually tagged as home page, content page and so on, but in a medium or large web site this is not affordable. Therefore it is necessary an automatic or semi-automatic method to tag the pages. In this proposal a phase of page classification is include (Figure 1) based on a learned model for the different categories of pages and using the features defined above. Hwanjo et al. [10] propose to use SVM with positive samples to classify web pages. Xu et al. [11] also introduce the SVM to deal with the heterogeneous data that appear in a web page as link, plain text, title page or anchor text. Holden and Freitas [12] make use of the Ant Colony paradigm to find a set of rules that classify the web pages into several categories. The study of complex web page classification algorithms is out of the scope of this paper so two well known learning methods have been included: naive-bayes and C4.5.

In this tool, a supervised learning stage has been included. The user selects and manually tags a set of pages that makes up the initial training set. With this initial training set, the learning process is launched and the results are tested by the user. If there are bad classified pages, the user can introduce them into the learning set with the correct tag. After some cycles, a correct model is obtained and the pages of the site are classified.

5 Transaction Identification

A transaction is defined as a set of homogeneous pages that have been visited in a user session. Each user session can be considered as only one transaction composed of all the visited pages or it can divided into a smaller set of visited pages. The transaction identification process is based on a split and merge process in order to look for a suitable set of transactions that can be used in a data mining task.

Formally, a transaction is composed of an IP address, a user identification and a set of visited pages which are identified by its URL and access time.

$$t = <ip_t, uid_t, \{(l_1^t.url, l_1^t.time), ..., (l_m^t.url, l_m^t.time)\} >$$

$$For\ 1 \leq k \leq m,\ l_k^t \in L,\ l_k^t.ip = ip_t,\ l_k^t.uid = uid_t \tag{1}$$

To realize the split stage in the transaction identification there are different strategies.

Transaction Identification by Reference Length This strategy, proposed by Cooley et. al. [2], is based on the assumption that the time that a user spends in an auxiliary page is lower than a content page. Obtaining a time t

by a maximum likelihood estimation and defining a threshold C, the pages are added to the transaction if they are considered auxiliary-content:

$$1 \leq k \leq (m-1) : l_k^{trl} length \leq C \ and \ k = m : l_k^{trl} length > C \qquad (2)$$

While for only content pages transactions:

$$1 \leq k \leq m : l_k^{trl} length > C \qquad (3)$$

Transaction identification by Maximum Forward Reference This strategy is based on the idea proposed by Chen et al. [13]. A transaction is considered as the set of pages from the first visited page until the previous page where the user does a back reference. A back reference appears when the user accesses again to a previously visited page in the current session, while a forward reference is to access to a page not previously visited in the current session. So the maximum forward reference are the content pages and the path to the maximum reference is composed of index pages.

Transaction Identification by Time Window This strategy divides a user session into time intervals lower than a fixed threshold. In this strategy the last visited page normally does not correspond to a content page unlike the previous strategy. If W is the size of the time window, the accesses that are included to the transaction (1) are those that fulfill:

$$l_m^t time - l_1^t time \leq W \qquad (4)$$

This strategy is normally combined with the previous ones.

6 Model Generation

To characterize the visitors of the web site, it is interesting to detect the access patterns, that is, what type of pages are visited and also navigation patterns, that is, how the visitors browse the pages in the web site. Both patterns are of interest because they can help the web site designer to improve the usability or visibility of the site. To get these patterns a clustering stage is introduced into the tool and although many works have been proposed to deal with this problem [14–16], in the tool three well know methods have been used: Expectation Maximization, K-means and Cobweb. As input to the previous methods, both the identified sessions and the transactions are used.

Another information that is useful for the web site designer is to know if there exists any unusual relation among the pages that are visited by users. This information can be easily extracted from the transactions and user sessions by means of an association rule discovering module. The Apriori method proposed by Agrawal [17] has been used.

7   Experiments

The tool was implemented in Java and the learning methods were the ones implemented in Weka [18] and by now we are only focused in the development of the framework it will allow us to introduce new learning methods. The appearance of the tool is shown in Figure 2.
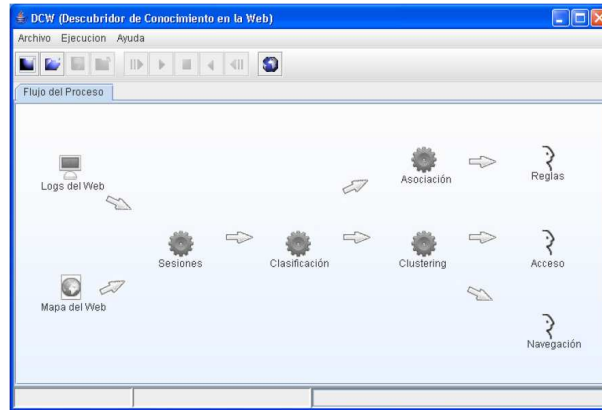

Fig. 2. DCW tool

To test the approach we select a log file corresponding to a week of accesses to the site of the Department of Computer Science (DIS) of the Univ. of Las Palmas de Gran Canaria (http://www.dis.ulpgc.es). The log file has 478512 entries that after the preprocessing phase (Sec. 2) it is reduced to 25538 entries.

Fixing a time threshold for session identification (Sec. 3) to 30 minutes, 9460 sessions were identified being the main page of the DIS the most visited pages with 1571 accesses. After the session identification the next stage is to train the classifier to tag the pages. In this experiment the pages where divided into two categories: content and auxiliary. The C4.5 algorithm was used to do induce a decision tree model and an accuracy of 85% was achieved.

Once the pages of the site are classified into auxiliary or content category, the pattern extraction is carried out. To get the access pattern of the visitors a clustering phase with EM algorithm is done. The results are the shown in Table 1. Two different clusters are obtained with correspond to users that visit mainly content pages while the other cluster represents the visitors that browse auxiliary pages. The first cluster could correspond to students and staff while the second one could correspond to "curious" visitors because they only browse auxiliary pages.

To get the access patterns, only sessions of 3, 4 o 5 accesses (pages visited) are considered. Table 2 shows the clusters obtained for 3 accesses sessions. The largest cluster corresponds to sessions that end up in auxiliary pages which

	Content pages	Auxiliary pages
Cluster 0	1	0.0323
Prob.=0.7012	D = 0.4463	D = 0.1768
Cluster 1	0	1
Prob.=0.2988	D = 0.4463	D = 0.4574

Log likelihood: -0.26076

Table 1. Access patterns results with EM clustering

means that the user abandons the site before reaching a page that gives useful information.

	access 0	**access 1**	**access 2**
Cluster 0	Auxiliary page	Auxiliary page	Auxiliary page
Prob.=0.6825	350(376.03)	374.97 (376.03)	375.03 (376.03)
Cluster 1	Content page	Content page	Content page
Prob.=0.3175	174.97(175.97)	174.97(175.97)	174.97(175.97)

Log likelihood: -0.8412

Table 2. Navigation patterns for 3 accesses sessions

Table 3 shows the results for the access patterns of session with 4 accesses and here it can be noted that the two largest clusters correspond to sessions that finish in content pages and only a small amount of sessions end up in auxiliary pages which can imply that the visitor does not find the information that was looking for.

	access 0	**access 1**	**access 2**	**access 3**
Cluster 0	Content page	Content page	Content page	Content page
Prob.=0.676	92.95(93.95)	92.9(93.95)	87.82(93.95)	86.71(93.95)
Cluster 1	Auxiliary page	Auxiliary page	Auxiliary page	Auxiliary page
Prob.=0.2601	34.47(36.51)	35.49(36.51)	35.51(36.51)	35.49(36.51)
Cluster 2	Auxiliary page	Auxiliary page	Auxiliary page	Content page
Prob.=0.0639	10.01(11.54)	8.46(11.54)	10.35(11.54)	9.27(11.54)

Log likelihood: -1.21079

Table 3. Navigation patterns for 4 accesses sessions

Table 4 shows the association rules that were obtained with Apriori algorithms. The rules do not contribute to generate new knowledge because they are very obvious. For example the first rule expresses that if the second visited page is the studies "Informatic Engineering", the first page was the root of the site.

As the aim of this work is to present the framework for a WUM tool, therefore a comparative of the results with other techniques has not been carried out

Rules	Support	Confidence
access1=/subject/index.asp?studies=ITIS =¿ access0=/	16	1
access1=/subject/index.asp?studies=II =¿ access0=/	14	1
access2=/staff/ =¿ access0=/	16	1
access2=/student/ =¿ access0=/	15	1

Table 4. Association rules

because they can be found in the literature. Comparing with other open source tools, we have found that the most similar is WUMprep [19] which only cover part of the Data Preparation stage and unlike DCW that has a GUI, WUMprep it is based on Perl script. In relation to the model generation and validation there are two well-know tools as Weka [18] and RapidMiner [20]. They are oriented to data mining in general and the previous stage of web log cleaning must be done with another tools.

8   Conclusions

In this work a tool for Web Usage Mining has been presented. It allows to realize all phases to get access and navigation patterns and also association rules. The implementation was done in Java and making use of the Weka inducers which allow to test new model induction algorithms. To test the tool, some experiments were carried out with a log file of more than 700.000 entries and they reveal some behaviors of the visitors that the designer of the web do not know and it can help them to redesign the web site to offer a better service to the students and staff of the Department of Computer Science of the ULPGC.

Future work is twofold. On the one hand, some elements of the proposed tool needs to be improved to tackle for example with dynamics web sites. One the other hand, other methods can be tested in the classification and clustering phases. In the page classification phase the computation of new features and the use of SVM as classifier.

References

1. Hand, D.J., Mannila, H., Smyth, P.: Principles of Data Mining. The MIT Press (2001)
2. R. Cooley, B. Mobasher, J.S.: Data preparation for mining world wide web browsing patterns. Journal of Knowledge and Information Systems **1** (1999) 5–32

3. Chakrabarti, S.: Mining the Web.Discovering Knowledge from Hypertext Data. Morgan-Kaufmann Publishers (2003)
4. Srivastava, J., Cooley, R., Deshpande, M., Tan, P.N.: Web usage mining: Discovery and applications of usage patterns from web data. SIGKDD Explorations **1** (2000) 12–23
5. Cooley, R., Srivastava, J., Mobasher, B.: Web mining: Information and pattern discovery on the world wide web. In: Proc. of the 9th IEEE International Conference on Tools with Artificial Intellegene (ICTAI'97). (1997)
6. Berendt, B., Mobasher, B., Nakagawa, M., Spiliopoulou, M.: The impact of site structure and user environment on session reconstruction in web usage analysis. In: WEBKDD 2002 - Mining Web Data for Discovering Usage Patterns and Profiles. LNAI 2703 (2003) 159–179
7. Spiliopoulou, M., Mobasher, B., Berendt, B., Nakagawa, M.: A framework for the evaluation of session reconstruction heuristics in web usage analysis. INFORMS Journal on Computing **15** (2003) 171–190
8. Catledge, L.D., Pitkow, J.E.: Characterizing browsing strategies in the World-Wide Web. Computer Networks and ISDN Systems **27** (1995) 1065–1073
9. Lorenzo, J., Hernández, M., Méndez, J.: Detection of interdependences in attribute selection. Lecture Notes in Computer Science **1510** (1998) 212–220
10. Yu, H., Han, J., Chang, K.C.C.: Pebl: Web page classification without negative examples. IEEE Transactions on Knowledge and Data Engineering **16** (2004) 70–81
11. Xu, Z., King, I., Lyu, M.R.: Web page classification with heterogeneous data fusion. In: Proceedings of the Sixteenth International World Wide Web Conference (WWW2007), Alberta, Canada (2007) 1171–1172
12. Holden, N., Freitas, A.: Web page classification with an ant colony algorithm. In et al., X.Y., ed.: Parallel Problem Solving from Nature - PPSN VIII, LNCS 3242, Springer-Verlag (2004) 1092–1102
13. Chen, M.S., Park, J.S., Yu, P.S.: Data mining for path traversal patterns in a Web environment. In Proceedings of the 16th International Conference on Distributed Computing Systems (1996)
14. Xiao, J., Zhang, Y.: Clustering of web users using session-based similarity measures. In: 2001 International Conference on Computer Networks and Mobile Computing (ICCNMC'01). (2001) 223–228
15. Bianco, A., Mardente, G., Mellia, M., Munafo, M., Muscariello, L.: Web user session characterization via clustering techniques. In: Global Telecommunications Conference, 2005. GLOBECOM '05. IEEE. (2005)
16. Chen, L., Bhowmick, S.S., Li, J.: Cowes: Clustering web users based on historical web sessions. In: 11th International Conference on Database Systems for Advanced Applications, Singapore (2006)
17. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: 20th Int. Conference on Very Large Data Bases. (1994) 487–499
18. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques. 2 edn. Morgan Kaufmann, San Francisco (2005)
19. Pohle, C., Spiliopoulou, M.: Building and exploiting ad hoc concept hierarchies for web log analysis. (In: Data Warehousing and Knowledge Discovery, Proceedings of the 4th International Conference, DaWaK 2002)
20. Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., Euler, T.: Yale (now: Rapidminer): Rapid prototyping for complex data mining tasks. (In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2006))