Clickstream Data Warehousing for Web Crawlers Profiling

Anália Lourenço and Orlando Belo

Abstract — **Web sites routinely monitor visitor traffic as a useful measure of their overall success. However, simple summaries such as the total number of visits per month provide little insight about individual site patterns, especially in a changing environment like the Web. In this paper it is described an approach to usage profiling based on clickstream data collected on several Web servers' sites and stored in a specialized clickstream data warehousing. We aim at providing valuable insights about common users, but also preventing unauthorised access to contents and any form of overload that might deteriorate site performance. Common crawler detection heuristics help to classify sessions, enabling the construction of site-specific profile training sets. Then, classification algorithms are used for building predictive models that can evaluate unseen sessions, namely their nature and potential site hazard, when they are still ongoing.**

Index Terms — **Data Warehousing, Clickstream Data, Web Housing, Web Usage Mining, Web Crawler Profiling.**

I. Introduction

TODAY, the World Wide Web is an universal information resource. The lack of centralised control and the permanent availability of new contents have converted it into a privileged environment for the exchange of information and services. In fact, most organisations are using it as a main access interface for their information systems. Besides research, or pure entertainment, the Web is now a huge, highly heterogeneous business centre that generates many millions of dollars per year. As it happens in any other business centre, the enhancement of user experience and the accommodation of new user requirements are mandatory. In the early days, simple measures of unique visitors served as indicators for site performance. But most administrators subsequently have learned that measures of visitor retention and loyalty are in need for capturing user behaviour and designing user attraction and retention strategies. The business implications of profiling site visits are huge for Webmasters, but especially for Web service providers such as portals, personalized content providers and e-tailers [2]. The evaluation of system's specifications and goals, the personalisation of contents, the improvement of system performance, the identification of business opportunities, and the design of new marketing strategies are guided by these insights. These same profiles also assist on another key area: privacy and security. The detection and containment of illegitimate and non-desired activities, such as server performance deterioration, privacy and copyrights violation and fraud, are ever more important on Web analysis. In particular, the detection and monitoring of Web crawler activities have become a challenge [13]. Web crawlers are commonly related to large-scale search engines and directories and specialised services such as investment portals, competitive intelligence tools, and scientific paper repositories [9]. However, many have been the times that Webmasters have reported malicious crawling over their sites. Scenarios such as server overload, unauthorised content access (e.g. email harvesting or illegal competitive advantage) and fraudulent behaviour (e.g. impersonation or click-through overrating) are often associated to crawling. Privacy violations and economic losses impel drastic containment actions whereas moderated site indexing by general and focused services is in the best interest of the sites. The ability to differentiated regular usage from crawler usage promotes a better understanding of each community as well as enhances the analysis of site overall metrics, reaching a compromise between desired visibility and privacy and security concerns. A lot of relevant research has been done in the area of Web Usage Mining [12], which directly or indirectly addresses the issues involved with the extraction and interpretation of Web navigational patterns. Main studies such as the improvement of site topology [10] the prediction of Web traversing/shopping behaviour [3] and the clustering of Web usage sessions [8] are primarily based on Web server logs, possibly supplemented by Web content or structure information [4]. In this paper, we present an end-to-end approach to differentiated Web usage profiling and containment. We propose an analytical approach to modelling Web user profiles that reveals the relevance of studying Web crawler activities in general clickstream analysis. By using the clickstream data recorded in Web server log files, we developed and evaluated an approach to generate site-specific Web session classifiers that are able to differentiate between regular and crawler usage and to alert about potential hazards when the sessions are still active. We constructed a clickstream data warehousing system that captures main aspects of browsing behaviour, handles the limitations of server log-file data and sustains common heuristic-based crawler detection. Empirical results showed that crawler visits are somewhat challenging to track down since they are getting ever more similar to regular visits to pass by undetected. Although no actual attempt of attack has been spotted in the period under analysis, most crawlers did not sustain self-identification nor took into consideration

Anália Lourenço is with the Institute for Biotechnology and Bioengineering, Centre of Biological Engineering, University of Minho, Campus de Gualtar, 4710-057 Braga, PORTUGAL (e-mail: analia@deb.uminho.pt).

Orlando Belo is with the Department of Informatics, School of Engineering, University of Minho, Campus de Gualtar, 4710-057 Braga, PORTUGAL (Phone: +351 253 604470, Fax: +351 253 604471, e-mail: obelo@di.uminho.pt).

ISBN: 978-988-18210-6-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

site's crawling policy.

II. Clickstream Data Warehousing

Clickstream data offer a wide range of opportunities for modelling user behaviour. By definition, the term clickstream denotes the path an user takes while visiting the Web site, i.e., it reflects a series of choices by identifying which pages were visited, how long the user stayed at each page, whether or not he made an online purchase or a software download and so on. Both site navigation (including dynamic personalization and pre-fetching pages) and E-commerce and recommendation (namely adaptive one-to-one marketing) involve building user profiles based on a chosen unit of analysis or level of aggregation [2].

A. Common Data Pre-Processing

Before describing data pre-processing it is important to establish a formalism. Let S1, S2, ..., SN be N user sessions and assume that in these data the number of unique users is M and users are identified by a uid -> {1, 2, ..., M}. Each session Si is defined as a tuple of the form <uid,Ci> where uid corresponds to the user in session Si and Ci is a set of tuples of the form <page, access_details>, where each tuple represents a Web page request. Depending on site's infrastructure (frame-based or not, template-based or not, etc), this request results in the recording of more than one hit (document request) in the server's log file. Page view details include standard information from http headers (such as time of access, IP address, referrer field, etc.), other information such as whether the user made a purchase or a download in that particular page and the approximated time spent on page viewing. Common Web data pre-processing aims at identifying individual users and sessions based on raw log file data [4]. In particular, log data are to be filtered and specific fields are to be processed, time-frame single-user requests are to be grouped into sessions and, whenever possible, sessions are to be completed based on the site topology and other relevant information (figure 1).

Data Filtering. The HTTP protocol requires a separate connection for every file that is requested from the Web server. Therefore, a user's request to view a particular page often results in several log entries since graphics and scripts are downloaded in addition to the actual HTML file. In most cases, only the log entry of the HTML file request is relevant and should be kept for the user session file. Since profiling aims at getting a picture of the user's behaviour, it does not make sense to include file requests that the user did not explicitly request. A default list of filename suffixes is enough to ensure the elimination of irrelevant data. For instance, all log entries associated to images, music and videos (such as gif, jpeg, mp3 or wav) or common scripts (such as cgi).

User Identification. As mentioned previously, this task is greatly complicated by the existence of local caches, corporate firewalls, and proxy servers. However, there are heuristics that can be used to help identify unique users. The basic element of user identification is the IP address of the machine making the requests. Attempts to identify proxy-related users are based on agent and referrer analysis. Within a given time-frame (typically 30 minutes period), if there are agent changes (browser or automatic programs, except ancillary applications such as PDF viewers) and/or referrer sequence presents inconsistencies (for example, it is not possible to access a given page from the last recorded page), it is reasonable to assume that different users are behind that IP address. Of course, this approximation is not bulletproof and involves additional log processing, namely the collection and analysis of information on referrer and site topology to construct browsing paths for each user.

Request and Referrer Identification. When processing server hits it is not always to determine the actual user request. In a static Web page, content is determined once, when the page is created. In a dynamic Web page, content varies based on user input and data retrieved from external sources. Detailed site topology may help to group together hits related to a given request and the analysis of user input may resolve some template-based requests. Moreover, these additional data can help on completing referrer information whenever there is no record and/or there are log gaps.

Session Reconstruction. The reconstruction of user sessions is affected by the stateless nature of the HTTP protocol and log gaps. The reconstruction can be either proactive or reactive [11]. Proactive mechanisms enforce session delimitation while sessions are still operational, using cookie and session identifiers generated by Web application servers. On the other hand, reactive heuristics perform session delimitation a posteriori, based on upper thresholds on total visit time or total page viewing time, considering a typical threshold of 30 minutes. Additionally, methods similar to those used for user identification can be used for path completion.

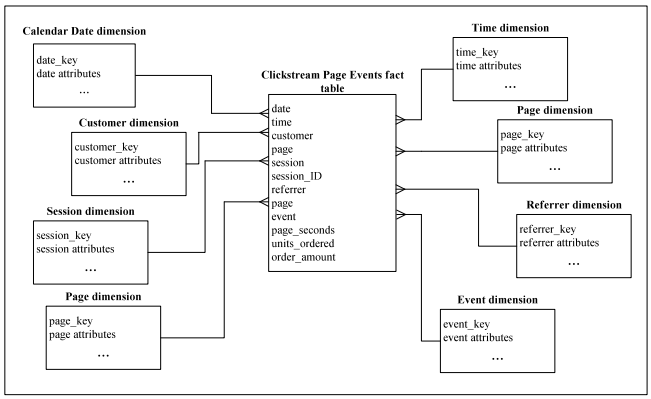

Fig. 1. Looking at page events dimensional schema.

B. Different Levels of Clickstream Processing

The grain or level of observation for clickstream analysis depends on site-specific analysis goals and clickstream data availability. Session and page view are the two most common grains of analysis [14] [6]. A simplistic page view star schema may include dimensions such as: the common Date and Time dimensions (universal and local); a Causal dimension for page view site-specific semantic interpretation; the Page dimension that identifies individual page; the Event dimension that describes what happened on the page view; the Session dimension that identifies session classes; the degenerate dimension Session_ID, used to roll up all page events of each session in an unambiguous way; and, the Product dimension when commercial activities are taking place. For each page view, the number of seconds

ISBN: 978-988-18210-6-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

elapsed before the next page event is recorded an. The units_ordered and the order_amount have an explicit semantic. These fields will only have non-empty value when the event in cause places the order. This situation will cause many zero or null field values for a great deal of records. Nevertheless, they are still considered relevant assets, because they tie all-important Web revenue to behaviour. Figure 2 illustrates a simplistic star schema where the grain was set to be one record for each completed session. In this sense, it may appear strange that the Page dimension is included in the design. However, in a given session, there is one particular page that is very interesting: the entry page. So, in this design, the dimension describes the page the session started with, enabling the study of how and why the customer accessed the site. In a more comprehensive design, other dimensions could also assist and enrich this particular fact table. For example:

- Universal Date and Universal Time, Local Date and Local Time. There could be two couples of date and time dimensions rather then one, standing for universal and local values. These elements are meant to deal with two conflicting requirements. First, the synchronisation of all session dates and times across multiple time zones could be attended. In order to achieve events synchronisation across multiple servers and processes, all session dates and times must be recorded in an uniform, single time zone (like GMT). On the other hand, the recording of the session according to the user's own clock could also be interesting.
- Causal dimension. This dimension focus broad market conditions affecting all possible products involved in a session. The idea is to assess how certain "causal factors" affected the users' interaction with the site. For instance, if an ad is running on the Web or on TV, it is interesting to know if it somehow influenced the shown interest in the site. However, it is must be highlighted that this particular Causal dimension focus on the overall and not on individual products. Otherwise, it would be inappropriate to have it included in this particular design.

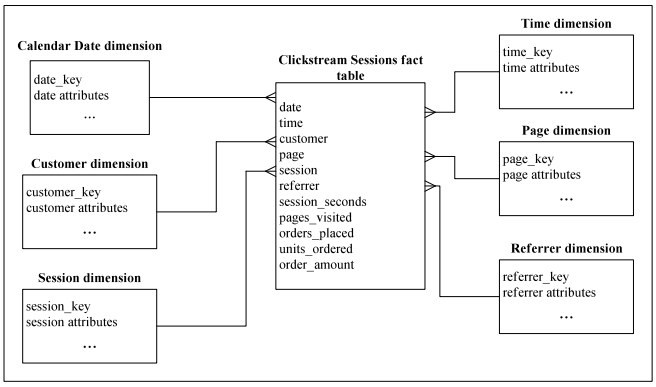

Fig. 2. Looking at session data dimensional schema.

Regarding fact measures, we can say that: the session_seconds represents the total number of seconds the customer spent on the site during the session; the pages_visited stands for the number of pages visited during the session and can be used to analyse the promptness with which a user finds what he is looking for or to identify particular traverse purposes; the orders_placed, units_ordered and order_amount concern the number of purchase orders, the total number of purchased products and the amount of cash spent, respectively.

C. Web Usage Defferentiation: Different Purposes, Different Analysis

The relevance for Web usage differentiation is twofold: focused clickstream analysis to enhance decision making and, in particular, re-structuring, personalisation and marketing strategies; and, detection and containment of crawler activities that, intentionally or not, are inflicting some damage to the Web site. Due to the ever growing similarities between crawler and regular user patterns, usage differentiation challenges common Web data pre-processing and profiling [15]. Regular users, Web proxies and Web crawlers are representative of distinct usage: 1) Regular users are considered the primary target of profiling and the study of their patterns provide the guidelines for site re-structuring, marketing and personalisation, among other activities; 2) Web crawlers perform automatic visits aiming multiple purposes, not always regarded as useful or legitimate by Webmasters. They are not considered relevant for the before mentioned activities, but they detain privileged information about site indexing, contents popularity and security breaches; and 3) Web proxies aggregate regular and non-regular sessions into hard to analyse combined sessions and only through differentiation and careful processing it is possible to analyse each group of users properly. Web crawlers emerged at the earliest days of the World Wide Web and have been always related to the maintenance of search indexes [1]. Yet, for some time now, Web crawler have become involved in many specialised services such as investment portals, competitive intelligence tools, and scientific repositories, often retrieving focused time-sensitive information such as stock rates, shopping data and news. Furthermore, download managers or site rippers (offline browsing), email harvesters, hyperlink and Web page validation programs, statistics compilers and an ever growing number of crawlers with multiple, extendable purposes or whose purpose is unknown or dubious belong to the current population of the Web [16]. Web crawlers can be broadly defined as software programs that perform automatic information retrieval in order to meet a pre-defined set of user-specified topics or keywords [9]. In this work, we will not detail crawler implementation since regardless the implementation crawler behaviour is reflected on clicks.

D. Standard Identification Heuristics

The guidelines for crawler design and implementation, stated in the Robot Exclusion Standard [7] addressed both ethics and performance. Crawlers should not interfere with the normal functioning of Web servers nor be used to acquire privileges over Web contents. Moreover, Web administrators should use the "robots.txt" file to specify traverse boundaries. Depending on site administration concerns and previous crawler history, certain programs may be tagged as non-desired. Likewise, programs might be welcome only in areas that do not interfere significantly with server performance and particular site areas may be suggested as the most interesting to certain programs.

ISBN: 978-988-18210-6-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Assuming that crawler designers comply with these guidelines, standard detection heuristics were based on self-identification and crawler-alike navigation patterns.

User Agent Identification. At the same time, crawler self-identification is the most interesting and the less reliable heuristic. The user agent field of the Web server logs is expected to contain the identification of the programs accessing the site. A comprehensive identifier can provide invaluable information about the corresponding program and its current purpose. However, dubious identifiers make it very hard to track down the program and may even lead to false conclusions. Most well known browsers include the word "Mozilla" somewhere in their user agent field. The number and type of representative fields in the user agent string is neither constant nor ordered and often mozilla-alike user agents do not present a body standard, containing misleading and dubious identifiers. Agents such as "googlebot (larbin2.6.0@unspecified.mail)" and "htdig(NOTGooglebot!)'/3.1.6(twilliams@answerfinancial.com)" use the word "googlebot" in a way that should not be regarded as Google-related and may mislead further analysis. Finally, identifications such as "contype", "Java/1.4.1_02", "FDM1.x" or "snoopyv1.X" do not provide any insight about the undertaken activities.

Request of Robost.txt File. Programs should respect site traverse policy. The "robots.txt" file, located under the root directory of the Web server, provides the access policy for different user agents. Programs should check for particular disallow entries as well as general restrictions. Although requesting this file is usually associated to crawling, its containment abilities are far too limited. Even though its hyperlink is not visible, any person can access the file by means of a regular URL request. The presence of a valid "robots.txt" file does not actually protect the contents. Many crawlers disregard these guidelines (for performance reasons or intentionally) or use them to identify content-sensitive site areas.

Web Session Metrics. It is a common assumption that humans are too slow performing their visits when compared to Web crawlers. Therefore, detection heuristics take into account "so-called crawling patterns" such as: large session lengths (more than 1000 requests) over a short period of time (typically seconds); visits embracing many page views and do not providing any referrer information; visits focusing a single page; visits including many pages in a recurrent way; programs associated to a large number of sessions (over the site's average number of visits), which do not include the keyword "Mozilla" in the name and have never logged in. The main drawbacks of these heuristics relate to site-specific activity, i.e., they need to be adapted to particular site usage and require constant updating to keep up with site-specific crawler pattern evolving. Also, these rules are unable to deal with non-declared crawlers and may trigger a considerable rate of false positives. Crawler, proxy and anonymization agents tend to present certain similarities in terms of session metrics and this may cause unwanted and costly session containment.

III. A Joint Approach to Profiling and Server Monitoring

Our approach aims at delivering an integrated approach for enabling site-specific differentiated navigation pattern analysis and server monitoring. We consider that focused profiling, server performance monitoring and containment of unauthorised accesses and fraud are the main reasons for the enforcement of crawler identification. The adequate analysis of usage profiles leads to efficient content management while server monitoring guarantees quality of service and alerts for potential hazards. Our process of analyzing clickstream data and taking actions as a result of that analysis can be viewed as a circular process and involves three main components, namely: 1) a clickstream processing (ETL) component that supports the processing of standard server log contents towards the reconstruction of Web sessions and standard crawler detection aiming at differentiated profiling; 2) a machine learning component that studies user behaviour to build crawler identification models; and 3) a containment component that monitors user sessions, identifies Web crawlers and restraints visits when server performance or security are affected.

```
Let S be the set of reconstructed Web sessions.
Let C_BD be the crawler identification database.
Let ToInspect be the sessions that need manual inspection.

For each s in S Do
   selfIdentified<- robotsFileRequested (s.requests)
   foundMatch<- checkUserAgent(C_BD, s.userAgent)
   If (foundMatch or selfIdentified)
      s.agentType<- "Crawler"
   Else
      s.agentType<- "?"
      addSession(ToInspect, s)
   End If
End For
For each t inToInspectDo
   manualUserAgentCheck(t.userAgent)
End For
finalSessionLabelling(S, ToInspect)
```

Fig. 3. Web session labeling procedure.

A. The ETL Component

Common processing includes the resolution of IP addresses, the parsing of the requests and referrers and the identification of user agents. The component enables both proactive and reactive session reconstruction. Specifically, it enables the definition of proactive mechanisms based on site-specific analytical infrastructure and uses a reactive heuristic based on maximum page view, considering a threshold of 30 minutes. Web session labelling is a two-fold process (Figure 3): standard detection heuristics label well-known Web crawlers, and manual inspection allows further label reviewing. The catalogue of user agents supporting standard detection was gathered from well-known crawler lists. All user agents that match catalogue entries and agents that request the robots.txt file are labelled as crawlers. Ancillary agents such as PDF readers are labelled as applications. Unknown agents or agents whose session metrics are somewhat suspicious are labelled as unknown and will not take part of the data mining process. Only well-known browser-related agents presenting common session metrics are labelled as browsers. After usage differentiation, focused pre-processing is deployed over regular and crawler data streams. Agent-specific session reconstruction address parallel and multi-agent crawler activities and multi-agent sessions related to proxy servers or Web browsers, assisted by some application(s). Both differentiation-related and differentiated profiles, i.e., the single-machine and single-agent sessions that support the construction of differentiation decision support models and

ISBN: 978-988-18210-6-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

the differentiated profiles are stored into the clickstream warehouse. Besides conventional date and time dimensions, the warehouse schema includes page, referrer, event and session dimensions. Page and event dimensions enable topological and semantic request tagging as the first identifies the request within site's topology and the latest associates the request to a meaningful action. The referrer dimension identifies both external and internal referrers, i.e., the page that somehow "initiated" the session and the pages traversed by the user till the current request. The session dimension places the request into a timeline sequence with a given IP address and user agent. Each page event is further characterised by the number of seconds spent in its viewing and the overall HTTP status, resulting from the retrieval of all associated documents.

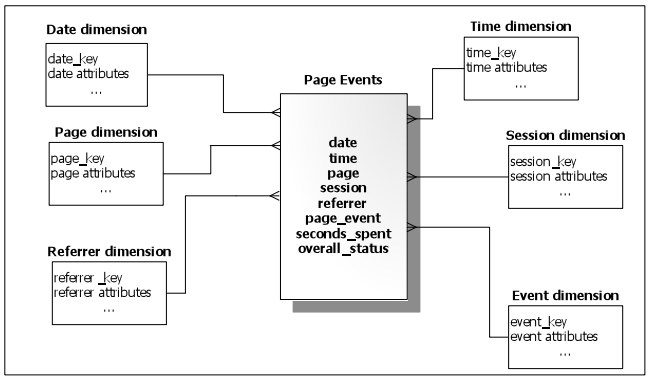

Fig. 4. The star schema of the clickstream data warehouse.

B. The Machine Learning Component

The Machine Learning component aims at building usage differentiation models and, in particular, models capable of identifying crawler sessions at their earliest stage. As such, it involves the selection of a supervised learning algorithm, the analysis of the minimum number of requests needed to perform classification based on error cost and the assessment of model validity, determining the best indicators for re-construction. In order to construct a site-specific user identification model, we use general session metrics such as session duration and length, weighted values for main document classes and HTTP request methods. General session metrics are computed during Web session reconstruction, while the goal attribute, i.e., the usage class, is outputted by the labelling semi-automatic schema. Model construction embraces one page request at a time and incrementally session information will be extended till the evaluation criterion is met (figure 5).

```
Let S be the set of reconstructed, labelled Web sessions.
Let n be the maximum length of session.
Let x be the accuracy threshold.
stopMining<- false
If (!stopMining)
        Fori=1, …, n Do
            subSessions_i<- gatherPageRequestsTill(S, i)
            decisionModel_i<- C45Mining(subSessions_i)
            If (i>1)
                If (|accuracy_i - accuracy_i | < x)
                    stopMining<- true
                End If
            End If
        End For
    End If
```

Fig. 5. Incremental model evaluation procedure.

Dataset 1 embraces session information up to one request. If a session contains more than one request, the dataset will only consider information till the first page request. Dataset 2 includes sessions with at least two page requests, ignoring shorter sessions and truncating sessions with more information. The procedure is repeated until a maximum number of page requests are reached or there are very few sessions left to support further datasets.

C. The Server Monitoring Component

Differentiating between authorised and unauthorised usage is a difficult problem that is often influenced by the domain under surveillance. Many conventional intrusion detection systems are based on attack signatures, i.e., patterns that match known or likely intrusions, assuming that the attacks keep a distinctive intrusive pattern over time. Decision models may be instructed to identify crawlers based on navigation patterns whilst server performance monitoring triggers containment actions whenever there is a service or security breach. Server monitoring is based in metrics such as: the number of requests received per minute, the distribution of requests per HTTP request method and document class type, the request of private contents and the number of status errors. Also, containment assessment takes into consideration previous crawling experience. Crawlers that cause a considerable load to the server and whose activities are not considered relevant should have a higher containment priority than, for instance, crawlers working for search engines and focused retrieval systems. In fact, session containment performance is gradual, issuing site close down only if server is severely affected.

IV. Experiments over a non Commercial Web Site

Natura is a *Natural Language Processing* (NLP) research project (at http://natura.di.uminho.pt) focused on Portuguese language. Its Web site contains project-related information, general NLP information and homepages of some of the project's members. The experiments were performed over six month data Web server logs and included a large variety of user agents and different navigation patterns for particular user agents, i.e., accounting for new and changing behaviour (Table I). Although this is a non-commercial site, the diversity of its contents is quite appealing in terms of differentiated usage profiling. Scientific publications, academic events, software and other NLP resources are mainly visited by students and researchers. Yet, the music repository embracing poems, lyrics, accords, music scores and karaoke files, attracts regular users as well as general and focused retrievers.

TABLE I
Web Sessions Statistics

Month	Total	Crawler	Regular
january	166490	59879	98875
february	175192	66091	103163
march	256649	120829	130126
April	222203	102445	115041
May	339413	135151	196535
June	318937	151324	161511

Month	Application	Unknown
january	6592	1144
february	4670	1268
march	4187	1507
april	3376	1341
may	5621	2106
june	4067	2035

ISBN: 978-988-18210-6-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

TABLE II
PERFORMANCE EVALUATION OF THE LEARNING ALGORITHMS OVER THE FIRST MONTH

	ZeroR				J48				NaiveBayes				DecisionTable			
	Acc	FP	FN	F1	Acc	FP	FN	F1	Acc	FP	FN	F1	Acc	FP	FN	F1
1	62.28	0	1	0	96.26	0.04	0.03	0.95	92.26	0.09	0.06	0.9	96.24	0.04	0.04	0.95
2	80.45	0	1	0	97.75	0.02	0.04	0.94	94.97	0.06	0.02	0.88	97.01	0.03	0.05	0.93
3	89.72	0	1	0	98.32	0.01	0.07	0.92	96.83	0.02	0.16	0.85	97.54	0.01	0.12	0.88
4	90.03	0	1	0	98.58	0.01	0.08	0.93	96.22	0.02	0.17	0.81	97.63	0.01	0.12	0.88
5	89.48	0	1	0	98.43	0.01	0.08	0.93	96.33	0.03	0.13	0.83	97.42	0.01	0.14	0.87
6	88.08	0	1	0	98.13	0.01	0.08	0.92	96.19	0.04	0.04	0.86	97.27	0.02	0.11	0.89

All data mining activities were supported by WEKA, one of the most popular open-source collections of machine learning algorithms for data mining. 0-R classsifier (Zero-R) is used as baseline reference while evaluating WEKA's implementation for Quinlan's C4.5 entropy-based algorithm, a simple decision table majority classifier, and a standard probabilistic Naive Bayes classifier. The 10-fold stratified cross-validation technique, which is the evaluation method of choice in most practical limited-data situations, supported the prediction of the error rate of the learning algorithms. Dataset construction was taken up to a maximum number of 6 page requests and the mining algorithm was selected based on the performance metrics outputted for the first trimester. As expected J48, NaiveBayes and DecisionTable outfitted ZeroR results at all times (Table II). The three algorithms exhibited an acceptable performance, although DecisionTable and J48 were found better with the latest presenting slightly smaller F1-measure values. When analysing the performance of the incremental J48 classifiers, classifiers built using information up to 3 page requests and 4 page requests present the best trade-off accuracy/cost. In the present scenario, the performance of the decision model over the next five months did not suffer a significant change. Accuracy loss was 2% or less and the false positive rate increase was almost null. Table 6. Performance evaluation of the learning algorithms over the first three months

V. CONCLUSIONS

Most of the times, Web profiling is focused on regular usage, aiming at contents popularity and visibility, i.e., assessing if contents are able to attract the desired audience and if such audience finds it easy to navigate site's structure. However, Web crawler analysis delivers additional and equally relevant information. The identification of programs related to search engines and focused indexing programs provides further acquaintance about site popularity and visibility. Crawlers do not share regular user interests nor have the same impact over Web sites. Their activities have to be analysed according to their application, evaluating whether they are important to the site and if they can somehow compromise it. In this work, we developed a joint approach to differentiated profiling and server monitoring. Our clickstream data warehousing component enables common server log processing and standard heuristic-based crawler detection. Our machine learning component assists on differentiated profiling and, more important, provides the means to build automatic site-specific usage differentiation models. Server monitor alerts will be triggered by the deterioration of server performance and any violations of site's privacy policy. By using the differentiation model, the monitor is able to identify which crawler-related sessions should be terminated to ensure server and site integrity. A prototype system was developed to check the feasibility of the approach and to conduct experiments to examine its effectiveness. In our dataset we found that models that incorporated sequence or path information doubled the hit rates over those that did not. We also have shown that our model has reasonable predictive power with regards to understanding which users are likely to make a purchase or not. We can predict those users that are likely to purchase with 42% accuracy with as few as six viewings.

REFERENCES

[1] Baeza-Yates, R. (2003). Information retrieval in the Web: beyond current search engines. International Journal of Approximate Reasoning, 34(2-3), 97-104.
[2] Bucklin, R.E., Lattin, J.M., Ansari, A., Gupta, S., Bell, D., Coupey, E., Little, J.D.C., Mela, C., Montgomery, A. and Steckel, J. (2002). Choice and the Internet: From Clickstream to Research Stream. Marketing Letters, 13(3), 245-258.
[3] Cho, Y.H. and Kim, J.K. (2004). Application of Web usage mining and product taxonomy to collaborative recommendations in E-commerce.Expert Systems With Applications, 26(2), 233-246.
[4] Cooley, R. (2003). The use of web structure and content to identify subjectively interesting web usage patterns. ACM Transactions on Internet Technology (TOIT), 3(2), 93-116.
[5] Cooley, R., Mobasher, B. and Srivastava, J. (1999).Data preparation for mining World Wide Web browsing patterns.Knowledge and Information Systems, 1(1), 5-32.
[6] Kimball, R. and Merz, R. (2000). The Data Webhouse Toolkit: Building the Web-Enabled Data Warehouse. Wiley Press.
[7] Koster, M. (1993). Guidelines for robots writers.
[8] Mobasher, B., Dai, H., Luo, T. and Nakagawa, M. (2002).Discovery and Evaluation of Aggregate Usage Profiles for Web Personalization.Data Mining and Knowledge Discovery, 6(1), 61-82.
[9] Pant, G., Srinivasan, P. and Menczer, F. (2004). Crawling the Web. In M. Levene and A. Poulovassilis (Eds.), Web Dynamics: Adapting to Change in Content, Size, Topology and Use.
[10] Pierrakos, D., Paliouras, G., Papatheodorou, C. andSpyropoulos, C.D. (2003). Web Usage Mining as a Tool for Personalization: A Survey. User Modeling and User-Adapted Interaction, 13(4), 311-372.
[11] Spiliopoulou, M., Mobasher, B., Berendt, B. and Nakagawa, M. (2003).A Framework for the Evaluation of Session Reconstruction Heuristics in Web-Usage Analysis.INFORMS Journal on Computing, 15(2), 171–190.
[12] Srivastava, J., Cooley, R., Deshpande, M. and Tan, P.N. (2000). Web usage mining: discovery and applications of usage patterns from Web data. ACM SIGKDD Explorations Newsletter, 1(2), 12-23.
[13] Stassopoulou, A. and Dikaiakos, M.D. (2006). Crawler Detection: A Bayesian Approach. International Conference on Internet Surveillance and Protection (ICISP'06).
[14] Sweiger, M., Madsen, M. R., Langston, J. and Lombard, H. (2002).Clickstream Data Warehousing. Wiley Press.
[15] Tanasa, D. and Trousse, B. (2004). Advanced data preprocessing for intersites Web usage mining. IEEE Intelligent Systems, 19(2), 59-65.
[16] Thelwall, M. and Stuart, D. (2006). Web Crawling Ethics Revisited: Cost, Privacy, and Denial of Service. Journal American Society for Information Science and Technology, 57(13), 1771-1779.

ISBN: 978-988-18210-6-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)