Knowledge-Based Systems 49 (2013) 50–62

Contents lists available at SciVerse ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Effective web log mining and online navigational pattern prediction

Abdelghani Guerbas, Omar Addam, Omar Zaarour, Mohamad Nagi, Ahmad Elhajj, Mick Ridley, Reda Alhajj *

Department of Computer Science, University of Calgary, Calgary, Alberta, Canada
School of Information Technology, Bradford University, Bradford, UK
Department of Computer Science, Global University, Beirut, Lebanon

ARTICLE INFO

Article history:
Received 31 October 2012
Received in revised form 16 April 2013
Accepted 18 April 2013
Available online 3 May 2013

Keywords:
Web mining
Weblog mining
Pattern analysis
Prediction
Navigation
Indexing

ABSTRACT

Accurate web log mining results and efficient online navigational pattern prediction are undeniably crucial for tuning up websites and consequently helping in visitors' retention. Like any other data mining task, web log mining starts with data cleaning and preparation and it ends up discovering some hidden knowledge which cannot be extracted using conventional methods. In order for this process to yield good results it has to rely on some good quality input data. Therefore, more focus in this process should be on data cleaning and pre-processing. On the other hand, one of the challenges facing online prediction is scalability. As a result any improvement in the efficiency of online prediction solutions is more than necessary. As a response to the aforementioned concerns we are proposing an enhancement to the web log mining process and to the online navigational pattern prediction. Our contribution contains three different components. First, we are proposing a refined time-out based heuristic for session identification. Second, we are suggesting the usage of a specific density based algorithm for navigational pattern discovery. Finally, a new approach for efficient online prediction is also suggested. The conducted experiments demonstrate the applicability and effectiveness of the proposed approach.

1. Introduction

Organizations, companies and institutions are relying more and more on their websites to interact with clients. Retaining current clients and attracting potential ones push these organizations, companies and institutions to look for attractive ways to make their websites more useful and efficient. To achieve this goal, some auditing work needs to be done. Such an auditing task can be performed in at least two alternative ways. First, users of a specific website can be asked to evaluate their browsing experience. Then actions will be taken to improve the website's structure and/or content based on the feedback received from the clients who did provide a feedback. Second, clients' automatically recorded navigational history is analyzed and the website is tuned up accordingly. There is no doubt that the second option is the best. This is mainly because it does not rely on clients' manual voluntary input. Not only that, but also the analysis of navigational history of clients can be fully automated. This kind of analysis is referred to as web usage mining (WUM) and precisely in our case it is web log mining.

Web usage mining has many applications [10], e.g., personalization of web content, pre-fetching and caching and support to the design, recommendation systems [15], prefetching and cashing [14], among others. There are various benefits of web usage mining, especially in e-commerce. Clients can be targeted with appropriate advertisement. Also, relevant products can be suggested to clients in real-time while browsing the website. According to [7], the usage mining process can be divided into three steps. It starts first with data cleaning and pre-processing. Second, the pre-processed data is mined for some hidden and useful knowledge. Finally, the web log mining process ends by analyzing the mining results.

Like any mining task, web usage mining has also to deal with huge datasets. Therefore, issues related to space availability and running time are present and has to be faced. Besides these computational challenges, there exist some other ones that need to be dealt with as well. These issues are mainly due to the nature of the web log file itself [1]. Web server logs were not meant in the first place to be used for tracking users' navigational behavior. Their first purpose was to be used by server administrators to keep track of the server's bandwidth and capacity. The following are globally the problems a data miner faces when using server side collected data namely web access log to discover navigational patterns.

* Corresponding author at: Department of Computer Science, University of Calgary, Calgary, Alberta, Canada. Tel.: +1 403 210 9453.
E-mail addresses: alhajj@ucalgary.ca, rsalhajj@gmail.com (R. Alhajj).

0950-7051/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.knosys.2013.04.014

First, we need to distinguish between different visitors. The problem is that certain visitors may use proxy servers or share the same machine to browse the website. Therefore, using the IP address assigned to a user's computer as a unique identifier might lead to erroneous results. Second, users use backward and forward buttons of the browser and these actions are not recorded in the log. Therefore, we need to deal with missing information. Also, when a user requests a resource, the server will most likely log more than one entry. Many records might be added to the log for one single request. We have to get rid of the extra information the log collects. Third, we need to identify the different browsing sessions a user might have within a period of time. Forth, we need to estimate the time spent by a user on the last page he/she visited during a specific session. In addition to the aforementioned problems which are directly related to the web usage mining process itself, there are problems related of its applications such as online navigational pattern prediction. This prediction task has to be done in a timely manner with the best accuracy possible.

In this paper, we are interested in the process of web log mining and online navigational pattern prediction. The usage data we have selected to work on is the data logged by a web server in a file called access log file. Usage data can also be collected from visitors' computers by using adapted browsers or by using techniques such as cookies; we stayed away from that option simply because it might raise privacy concerns. Therefore, the first contribution of this paper is directly related to the access log data cleaning and preparation. The second contribution is the use of a density based clustering algorithm to mine for navigational patterns. The third contribution is a suggestion of a new efficient and accurate way in predicting navigational behavior online. Test results support the effectiveness of our method.

The first contribution of this paper is related to data cleaning and preparation. It starts by creating page views and ends up by generating sessions. Amongst the existing approaches for sessions' identification are the time-based heuristics. The idea of these time based heuristics is the use of a duration threshold to decide whether a session has ended or not. Our contribution in this area consists of the improvement of such a heuristic in order to get better quality results.

The second contribution is the use of a density based clustering algorithm namely DBSCAN, to mine for navigational patterns. We have opted for clustering instead of association rules discovery or sequential patterns mining because both of these two techniques require a user input parameter such as the minimum support. This is not the only issue about these two approaches but also they cannot detect low frequent and meaningful patterns unless the minimum support is too low which means too many rules and frequent patterns to generate. The clustering algorithm we have opted for requires one crucial input parameter from the user. DBSCAN is very sensitive to that input parameter. However, we have used a second algorithm that helps in finding the right value to use for that input parameter. Also, we have opted for DBSCAN because it detects outliers unlike some other clustering algorithms such as K-means for example. Outliers are closely related to erroneous results thus detecting and removing them is important [31].

The last contribution is the use of an inverted index built from all identified sessions from the log to narrow down and speed up the search for a k nearest neighbors for an online session, where k is positive integer specified by the user. We also cared about the accuracy of predicting the pattern of an online session in two ways. First, by taking into consideration not only the pages a user have viewed so far but also the order of those pages. For this we used generalized suffix trees. Second, we cared about the patterns where the order is not important and for this we used a binary matrix with pages as columns and sessions as rows. Details of these methods will follow in the related sections.

The rest of this paper is organized as follows. Section 2 is dedicated to related work. Section 3 includes a description of the proposed framework that integrates the three contributions mentioned above. Section 4 presents the experiments conducted and the results obtained. Section 5 is conclusions and future research directions.

2. Related work

In WUM in general it is not required to know about a user's identity; however, it is necessary to distinguish between different users [4,13,20,27]. If a website requires users to sign in before they can start browsing, it will be very easy not only to differentiate between users but also to identify each single user. The problem arises when a website allows visitors to anonymously browse its content, which is common place. In this case, relying only on what the web server records in the log to differentiate between visitors becomes challenging. In fact, it becomes more challenging if the web server is logging visitors' activities using common log format. The difficulty in distinguishing between visitors arises from the fact that some visitors' activities will be logged with the same IP address due to transiting by the same proxy server or by sharing an internet connection. Different ways to distinguish between users has been listed in [24].

By using cookies users can be identified. When a visitor connects for the first time to a website that uses cookies, the server sends a cookie to the web client along with the requested resource. The next time the same user requests another resource from the same website, the browser will send the cookie stored on the visitors computer along with the request if the cookie is still not expired. The server will be able to recognize the user if the request is coming with a cookie. The problem with this approach is that users can delete cookies stored on their computers and the server will not be able to recognize the coming back visitors.

A simple and straightforward approach in resolving the issue of having users with the same IP or domain name is the elimination of all requests coming from proxies and shared IPs. For example, all requests having the word proxy or cache in their domain name will be eliminated. The drawback of this approach is that navigational patterns of proxy/cache users will not be discovered. Also the dataset may become too small to conduct a valid analysis.

In order to mine for navigational patterns it is mandatory to know what visitors have looked at each time they have visited the website. Each time a visitor comes to the website is considered a session. Identifying users' sessions from the web log is not easy as it may seem. Logs may span long period of time during which visitors may come to the website more than once. Therefore, sessions' identification becomes the task of dividing the sequence of all page requests made by the same user during that period into subsequences called sessions. Many approaches have been used by researchers for sessions' identification. According to [24], the most popular session identification techniques use a time gap between requests.

It has been mentioned in [7] that many commercial products use 30 min as default time-out threshold. However, many thresholds can be found in the literature. These thresholds vary from 10 min [5,7,12,22] to 2 h. It has been also mentioned in [24] that the most widely used time gap is 25.5 min established based on empirical data. This threshold has been calculated as follows. The authors of [5] conducted a study that allowed them to calculate the mean inactivity time within a site. The value found was 9.3 min. After they added 1.5 standard deviations to the mean, the value of 25.5 min was obtained and was defined as a standard inactivity time. This was defined as a cutoff threshold to identify