Web Usage Mining

Stu Westin
University of Rhode Island, USA

INTRODUCTION

Research studies concerning the use of the World Wide Web (WWW) have become quite common in the MIS, education, marketing, and e-commerce literature. Increasingly, the research methodology employed in these studies involves some form of Web usage mining. This research technique seeks to uncover the Web user's access and navigation behaviors through analysis of real-time data artifacts of Web usage. These data artifacts are sometimes referred to as *mouse droppings* since each datum results from a specific user action involving the mouse. The so-called *click streams*, the sequence of URLs visited by a Web user, are often the focus of Web usage mining. These data can be supplemented with timestamp information to reveal page viewing time. Mouse click coordinates (i.e., X, Y location) can also be of interest, depending on the research question.

Studies that rely on Web usage mining can be experimental or observational in nature. The focus of such studies is quite varied and may involve such topics as predicting online purchase intentions (Hooker & Finkelman, 2004; Moe, 2003; Montgomery, Li, Srinivsan, & Liechty, 2004), designing recommender systems for e-commerce products and sites (Cho & Kim, 2004; Kim & Cho, 2003), understanding navigation and search behavior (Chiang, Dholakia, & Westin, 2004; Gery & Haddad, 2003; Johnson, Moe, Fader, Bellman, & Lohse, 2004; Li & Zaiane, 2004), or a myriad of other subjects. Regardless of the issue being studied, data collection for Web usage mining studies often proves to be a vexing problem, and ideal research designs are frequently sacrificed in the interest of finding a reasonable data capture or collection mechanism. Despite the difficulties involved, the research community has recognized the value of Web-based experimental research (Saeed, Hwang, & Yi, 2003; Zinkhan, 2005), and has, in fact, called on investigators to exploit "non-intrusive means of collecting usage and exploration data" (Gao, 2003, p. 31) in future Web studies.

In this article we discuss some of the methodological complexities that arise when conducting studies that involve Web usage mining. We then describe an innovative, software-based methodology that addresses many of these problems. The methods described here are most applicable to experimental studies, but they can be applied in ex-post observational research settings, as well.

BACKGROUND

Approaches to Web usage mining can be server-centric or client-centric. In the former case the data are harvested from a server machine. In some instances this approach requires no special software mechanisms since server logs are maintained routinely by server software. Client-centric approaches always require special data collection mechanisms because standard browsers do not document user actions. An example of this is the *PCMeter* usage mining software application. This software runs in the background on the client machine recording click-stream data as the research subject interacts with a Web browser (see Johnson et al., 2004, and Montgomery et al., 2004, for usage examples).

Server logs provide the most frequent data source for usage mining studies. This is because the data are readily available in a standard, machine-readable format, and preexisting Web sites can be used as long as the server log data can be procured for analysis. However, the literature is rife with criticism and complaints about the shortcomings of this data source (e.g., Bracke, 2004; Fenstermacher & Ginsburg, 2003; Huysmans, Baesens, & Vanthienen, 2004; Montgomery et al., 2004; Spiliopoulou, Mobasher, Berendt, & Nakagawa, 2003). The problems arise from such confounding elements as multiple server types (e.g., proxy servers, image servers, and application servers), server farms and load balancing procedures, caching activities, stateless nature of sessions, and so forth. In the words of Shahabi, Banaei-Kashani, and Faruque (2001, p. 1) "... usage data acquisition via server logs is neither reliable, nor efficient. It is unreliable due to the side effects of the network ... [it is] inefficient because of usage data requiring extensive preprocessing before it can be utilized."

Other server-centric data collection approaches based on server-side scripting (e.g., ASP, ASP.NET, etc.) can prove useful in some circumstances. Consider, for example, the situation where one wants to investigate the impact of download time as a factor affecting user satisfaction or Web site success. Using server-side scripting, a delay mechanism can be easily built into a Web page so that the server will delay serving the requested page to the client until some precise, predetermined time has passed. Different experimental treatment levels are accomplished by merely manipulating the delay time that is scripted into the Web page. Here, the

experimental subject, using an ordinary browser, will have the perception that the page is slow to download because of the delay between when the page is requested (e.g., by clicking a hyperlink) and when the page is available in the browser.

As another scenario, consider the situation where the researcher wants to study the end user's Web search strategy by analyzing the click-streams (e.g., Chiang et al., 2004). Here again, server-side scripts in the Web pages could provide a simple data collection mechanism by logging each page request (page ID, server timestamp) in a server database. The advantages of this approach over relying on server logs are that the server-side scripts can be designed to capture the precise data objects in the particular format that is desired, and many of the aforementioned confounding items can be circumvented.

In considering these scripting approaches, it is obvious that client-side data collection mechanisms can be constructed just as easily. In most cases, Java applets, Java scripts, or VB scripts can be embedded into Web pages to handle the required tasks. The only difference in this client-side approach is that the data collection is being handled by the client rather than by the server machine. Neither approach provides any obvious benefits over the other, although in the client-side approach the Web pages for an experiment could be stored locally and thus WWW, or even network access, is not required. In all of the previous research settings, including those that harvest data from server logs, standard Web browser software such as Internet Explorer (IE) can be used in the research study.

One flaw in all of these research approaches (except, perhaps, the *PCMeter* tactic) lies in the fact that experimental access must be restricted to either (1) a limited set of Web pages that have been appropriately scripted for data collection, or (2) a specific set of servers from which log data can be procured ex-post. If the experimental subject is allowed to "wander" beyond this limited set of pages or sites (an activity that is quite fundamental to the nature of using the Web), then these actions will be unrecorded or inaccessible, and the validity of the research will be nullified. In the script-based approaches, a related complexity stems from the fact that all Web pages used in the experiment must be developed and maintained by the investigator—a task that can be quite labor intensive if a large number of pages are to be made available. Obviously, the experimental pages should usually be large in number and professional in appearance if external validity is to be preserved.

In some situations the research data can be collected without the use of client- or server-side scripting, or server logs. Click-stream data, for example, can often be gleaned through the use of standard network management software, or through *network sniffers* that can be configured to monitor Internet requests and/or page downloads. In this case the experimental subject can be allowed to roam beyond a predefined set of pages, and, again, standard browser software can be used on the client side. The problem here can be in the precision or in the format of the data, as the software was not designed for this purpose. Pages containing multiple frames, for example, may be logged as individual (frame) downloads in some circumstances and as a single page download in others. Client requests that are satisfied through the local cache may not be logged at all. Indeed, this approach suffers from many of the problems that plague server logs.

A problem with all of the data collection methodologies discussed thus far is that they suffer from a lack of experimental control. This lack of control comes from the fact that the instrument with which the experimental subject is interacting (a standard Web browser such as IE) was not designed to be used as a research tool.

Consider the situation in which we wish to study WWW use behavior through analyzing click-stream data. There are numerous ways of gathering data on page requests or page downloads, as noted previously. However, there are no means, short of direct visual observation, of recording *how* a particular page was requested. The page request could have come in the form of a click on a hyperlink, but the request could just as likely have been generated automatically through a dynamic action on the page (e.g., *meta refresh*), or through the *Back* or *Forward* buttons in the browser interface. Normal click-stream data will not distinguish between these circumstances, so the precise behavior or intentions of the experimental subject cannot be determined. This specific problem is noted by Montgomery et al. (2004, p. 580) as a shortcoming of the aforementioned *PCMeter* usage mining software: "However, the meter does not distinguish how the user navigates between pages (e.g., whether the user selects a hyperlink, a bookmark, or directly types in the URL to navigate to a page)."

Another problem has to do with the occurrence of multiple windows. Many hyperlinks open in new browser windows, and the user often has the option of requesting a new window at his or her discretion. The problem here is that the data collected cannot reflect which of the open windows is active when actions occur, or even that there are multiple windows in use (the opening and closing of windows are not logged). Again, the data cannot capture, or misrepresent the behavior in question; true *streams* cannot be traced.

A CLIENT-CENTRIC ALTERNATIVE

As noted earlier, the methodological problems, for the most part, stem from a lack of experimental control. Logic and research experience suggest that, for maximum experimental control, any experimental manipulations (treatments) and the data collection mechanisms should be as close to the experimental subject as possible. That is, they should be