Capturing User Access Patterns in the Web for Data Mining

I-Yuan Lin, Xin-Mao Huang, and Ming-Syan Chen
Electrical Engineering Department
National Taiwan University
Taipei, Taiwan, ROC
Email: mschen@cc.ee.ntu.edu.tw

Abstract—Existing methods for knowledge discovery in the Web are mostly server-oriented and approaches taken are affected by the use of specific web servers. As a result, it is difficult to capture individual Web user behavior from the current log mechanism. As an effort to remedy this problem, we develop in this paper methods for design and implementation of an access pattern collection server to conduct data mining in the Web. We also devise an innovative method, called page conversion, which converts the original Web pages to encrypted ones so that the devised data collection mechanism will not be deliberately bypassed. With the concept of page conversion, the methods we proposed involves a mechanism of software downloading to resolve the difficulty imposed by proxy servers and to effectively capture the Web user behavior. Using the devised mechanism, traversal patterns are generated and compared to those produced by the ordinary Web servers to validate our results. It is shown that the traversal patterns resulting from the devised system are not only more informative but also more accurate than those generated by ordinary Web servers, showing the importance and the usefulness of the mechanism devised.

I. Introduction

Capturing user behavior on the Web is essential for planning new services and applications. A typical approach in the field of web mining is to collect log files in a server and perform data mining on the collected data. However, this approach is not always practical. For example, in the case of a large enterprise, it may not be feasible to collect and analyze data from all the servers in the organization. In such cases, it is necessary to develop a method to collect data from a subset of servers. This paper describes such a method.

The basic idea is to use a middleware layer to intercept requests and responses between the web server and the client. The middleware layer can be implemented using a variety of technologies, such as proxies, load balancers, and application gateways. The key advantage of this approach is that it is relatively easy to implement and does not require any modification to the existing web servers. In addition, it allows for fine-grained control over the data collected, as the middleware layer can be configured to collect specific types of information, such as the user's IP address, the time the request was made, and the URL of the requested page.

The middleware layer intercepts HTTP requests and responses and modifies them in a way that makes it possible to collect data about user behavior on the Web. For example, the middleware layer can be configured to add a timestamp to each request, so that the time the request was made is recorded. Similarly, the middleware layer can be configured to add a user ID to each request, so that the identity of the user can be tracked.

The data collected by the middleware layer can be used to perform a variety of tasks, such as user profiling, content personalization, and advertising. For example, by analyzing the user's behavior on the Web, it is possible to identify groups of users with similar interests, and to tailor the content on the website to suit those interests. Similarly, by analyzing the user's behavior, it is possible to target ads to specific groups of users, which can be more effective than random targeting.

The middleware layer can also be used to collect data that is not available from the server logs. For example, the middleware layer can be configured to record the user's screen size and resolution, which can be useful for optimizing the layout of the website for different devices. Similarly, the middleware layer can be configured to record the user's browser type and version, which can be useful for testing the website on different browsers.

In summary, the middleware layer provides a flexible and powerful tool for collecting data about user behavior on the Web. By intercepting HTTP requests and responses, it is possible to collect data that is not available from the server logs. This data can be used to perform a variety of tasks, such as user profiling, content personalization, and advertising. The middleware layer is easy to implement and can be configured to collect specific types of information, making it a valuable tool for web mining.

References


[2] J. Han, J. Pei, and Y. Yin, "Mining Frequent Patterns without Candidate Generation," in Proc. 20th Int'l Conf. Very Large Data Bases, 1994, pp. 1-12.


[4] D. Lee, H. Zhang, and J. Han, "Mining Frequent Patterns without Candidate Generation," in Proc. 20th Int'l Conf. Very Large Data Bases, 1994, pp. 1-12.

[5] Y. Xiao and J. Han, "Mining Frequent Patterns without Candidate Generation," in Proc. 20th Int'l Conf. Very Large Data Bases, 1994, pp. 1-12.

[6] J. Han, J. Pei, and Y. Yin, "Mining Frequent Patterns without Candidate Generation," in Proc. 20th Int'l Conf. Very Large Data Bases, 1994, pp. 1-12.

[7] D. Lee, H. Zhang, and J. Han, "Mining Frequent Patterns without Candidate Generation," in Proc. 20th Int'l Conf. Very Large Data Bases, 1994, pp. 1-12.

[8] Y. Xiao and J. Han, "Mining Frequent Patterns without Candidate Generation," in Proc. 20th Int'l Conf. Very Large Data Bases, 1994, pp. 1-12.
Capturing User Access to Web Servers

A. Mining on Logs from Ordinary Web Servers

Users

User 1
(a, 1) (c, 4) (a, 6) (c, 9) (d, 11) (c, 12) (f, 15) (h, 17) (b, 18) (c, 19) (f, 20)

User 2
(b, 2) (c, 3), (d, 5) (c, 7) (f, 8) (h, 10) (f, 13) (c, 14) (a, 16)

- eVP Ya Lun e, rhr br x r Ur

Ho Ub Drf rse AcP h r Whfr d A Captiry y SurE y f Soft r bp DprFr EULm Uebd e Ur Wh F W Trh V y So e ur el Abd fo p f SUMP s fr bAr cusr blN fr e, rhr Isb hWhfr I - Sr. oumbolw Ur bp lru Phbr Prlr Urb, rW F S Trhbr e prsU - eW VP y - Smr Bu Ee, oulDp - e VP YZr br x Ur e OD sr e, rhr bp Du Sf r frN Bp p Dfr Sfr frsbr trbl h OE uO - Whfr H A Seb bYbn

vYR OVEEBEBEBEBEBEBEB EDER

@fr If sfr hlr, rfr Sfr Trhbr e WsAdI s fr bAr cusr blN fr eyr y Fl Durb F Whfr dp dfl DprFr EULm Uebd e Ur Wh F W Trh V y So e ur el Abd fo p f SUMP s fr bAr cusr blN fr e, rhr Isb hWhfr I - Sr. oumbolw Ur bp lru Phbr Prlr Urb, rW F S Trhbr e prsU - eW VP y - Smr Bu Ee, oulDp - e VP YZr br x Ur e OD sr e, rhr bp Du Sf r frN Bp p Dfr Sfr frsbr trbl h OE uO - Whfr H A Seb bYbn

I. INTRODUCTION TO APC SUPS FOR TRAVSORD AALDMORIVIAL

Capturing Usenet Act via Phe Bg WhpDs ns CFP D mHt jaiP. PUYMo ns ErH nub Lfg tigurirr, XiaH iub rct Cf D YhA Adt IPr tShPr Pug InamedIr rct ref Wealpy - Prtena an inuWg LIP Syte, y ur rt Cfr YhA

segp rec neginReME tUmte WoaIP R M Ut Lgtr mn bUH teFr ref Wealpy - Prtena lnH Tad Wt Aum P neginReM E tUmte WArct Late UtC PrAra Bt Yshbf in Pu iurtH YWig p PunteAPu rec inuop Prim us Lgtr itgrmir M u r ur intXgtg in rec Yb. C P etal YArct - Prtena guH eWig segp rec neginReME tUmte W Ybr IPu Ut tircete inuup - Yr tr ur intXf t A LaH hrA LeMr rec ut ulAaM lns Lnub rec Cfr o DateW,
A. Mining on Logs of Od a y

Capturing log files use Asaps tcp ip ps sm Ph et Wul s ur in f s B a c t u s D a m a n M i - Y a m L a i s, i n p t, f o a h u ti X L B H in o d R d i s
S = [(a1, u1), (b1, u2), (c1, u3), (d1, u4), (a1, u5), (u1), (b1, u2), (c1, u3), (d1, u4), (a1, u5), (u1)]

Capturing logs use Asaps tcp ip ps sm Ph et Wul s ur in f s B a c t u s D a m a n M i - Y a m L a i s, i n p t, f o a h u ti X L B H in o d R d i s

I. N T R O D U C T I O N

Capturing log files use Asaps tcp ip ps sm Ph et Wul s ur in f s B a c t u s D a m a n M i - Y a m L a i s, i n p t, f o a h u ti X L B H in o d R d i s

Acknowledgements

Capturing log files use Asaps tcp ip ps sm Ph et Wul s ur in f s B a c t u s D a m a n M i - Y a m L a i s, i n p t, f o a h u ti X L B H in o d R d i s

I. N T. R O. D
