

Title	A Graph-Based Web Usage Mining Considering Page Browsing Time
Author(s)	Mihara, Koichiro; Terabe, Masahiro; Hashimoto, Kazuo
Citation	
Issue Date	2007-11
Type	Conference Paper
Text version	publisher
URL	http://hdl.handle.net/10119/4098
Rights	
Description	The original publication is available at JAIST Press http://www.jaist.ac.jp/library/jaist-press/index.html , KICSS 2007 : The Second International Conference on Knowledge, Information and Creativity Support Systems : PROCEEDINGS OF THE CONFERENCE, November 5-7, 2007, [Ishikawa High-Tech Conference Center, Nomi, Ishikawa, JAPAN]



A Graph-Based Web Usage Mining Considering Page Browsing Time

Koichiro Mihara Masahiro Terabe Kazuo Hashimoto

Graduate School of Information Sciences, TOHOKU University
{mihara, terabe, kh}@aiet.ecei.tohoku.ac.jp

Abstract

With the increase of large web sites which have complex link structures, web access logs have caught attention as a clue for web site administrators to understand user's needs and demands. While conventional statistical analysis is used for most of the cases, web usage mining is an emerging attempt to apply data-mining based technique to web access log analyses. However, statistical and data-mining based analyses have been independently applied, and no method has been reported to correlate their results yet. This paper introduces a novel web usage mining method to combine the statistical analysis of page browsing time and the graph based data mining technique in order to extract users' typical browsing behaviors.

Keywords: Web access log analysis, Web Usage Mining, Graph Mining, Page browsing time

1 Introduction

Due to the growth of WWW related technologies, the number of web sites on the Internet increases rapidly, and human daily life is beginning to depend on such sites as shopping sites like Amazon.com, official sites of enterprises / organizations, promotion sites of events and so on. These web sites contain a variety of contents and complex link structures.

Web site administrators, who are constantly requested to improve the easiness of use of such large and complex web sites, need to analyze user's needs and demands in order to maintain their sites as easy to use as possible. Web access log has been recognized as a source of information to do this analysis. A web access log is a time-series record of users' requests which are sent to a web server when a user does some operation on a web page. Analyzing the logs is very useful for the administrators to understand users' behavior on the web site.



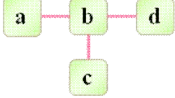
For web access log analysis, statistical methods, like Google Analytics, are widely used. The results of statistical analysis contain bounce rate, page views, page browsing time, and so on. Among them, analysis of page browsing time together with knowledge about the page gives administrators an immediate trigger to reorganize their sites in such a case when users rarely visit or stay at important pages (warning, caution, agreement, etc.) for very short period of time. It is a direct sign that users do not pay much attention to the message which the sites would like to convey to the users. Administrators should improve the accessibility or readability so that users will visit at reasonable frequency and stay at such pages for appropriate length of time.

On the other hand, Web usage mining (WUM) is an emerging attempt to apply data mining (DM) technique for web access log analyses. WUM can develop a pattern of users' navigation behavior in consideration of combination, order, etc. of the pages accessed by users. Examples of DM technique applied to the analysis are item-set mining, sequential pattern mining, and graph mining, etc. [1]. Particularly, graph mining can extract users' access patterns as a graph structure like the web site's link structure. It can handle the case where users browse more than one page at the same time. Tab browsers are gaining more popularity, which allow users to open several pages at a time. Graph mining will become an effective analysis means, especially for the web access log created with tab browsers.

Web site administrators analyze web access logs by using these statistical and WUM method independently. However, administrators have not yet correlated their results so far. We expect that administrators can analyze users' behavior more in detail with a WUM method considering quantitative information as obtained by a statistical method.

In this paper, we focus on page browsing time, and propose a novel WUM method which takes page browsing time into account, and extracts

Table 1. Example of the method used in web usage mining (In each pattern, a node intends a web page and an edge intends a users' transition path)

Pattern	Mining method	Features
	Item-set mining	The mined patterns can also be treated as an association rule. Useful to understand the relationships among the pages.
	Sequential pattern mining	This method treats only users' sequential transition, reconstructed by the requested order. It can't consider the <i>branched</i> transition to more than one page at the same time.
	Graph mining	This method can extract patterns, which structures are like the web site's link structure, considering the <i>branched</i> transition. Easy to identify the paths frequently used by users in the web site, and useful to reconstruct the web site's link structure etc.

users' transition patterns by graph mining. By doing this, we present one approach to correlate analyses of the statistical and WUM method. Moreover, by experiments which apply the proposed method to real web access logs, we explain the usefulness of our method.

2 Web Access Log Analysis

2.1 Statistical analysis

Web log analysis software, such as Google Analytics, gives a statistical analysis of web access log. It can output a variety of quantitative information such as bounce rate (Single page view visits divided by entry pages [2]), page views (The number of times a page was viewed [2]), page browsing time (the time during which users browsed the page), etc. These observed parameters show features and tendency of web page usage.

Among them, page browsing time is one of the most important parameters. About the pages which administrators want users to read carefully, such as an agreement page etc., it is desirable that user's browsing time is sufficiently long. On the other hand, about the pages, such as sitemap pages etc., should be passed soon, and their browsing time should be short. If the length of browsing time is different from what administrators expect, it suggests to improve the web page. Thus, page browsing time is very useful for administrators to reorganize the web site. We focus on page browsing time in the rest of the paper.

2.2 Web usage mining

Web usage mining, also called web log mining, includes analysis and prediction of users' behavior in the web site by extracting the access patterns from the page request records like web access logs, applying DM techniques.

For example, consider the case where, in some web site, a user browses the page *a*, and moves to the page *b*, then opens the page *c* on another window and moves from *b* to the page *d*, and finally browses *c* and *d* at the same time. In this case, users' access patterns extracted by WUM are as shown in the Table 1.

On shopping sites, which have been growing in recent years, users often browse more than one page at the same time to compare some commodities on the pages. Besides, tab browsers begin to prevail. In such situations, users' transition path becomes *branched*. Administrators who should improve the web site need to comprehend users' behavior including the *branched* transition like this. So, in what follows, we adopt graph mining which can treat *branched* transitions.

2.3 Graph mining algorithms

AGM [3], FSG [4], gSpan [5], and GASTON [6] are known as representatives of graph mining algorithms.

AGM is the first graph mining algorithm proposed by Inokuchi *et al.* AGM represents a graph structure as an *adjacency matrix*, and extracts frequent subgraphs based on an apriori-like algorithm with extension of the *adjacency*

matrix by node addition.

FSG uses an *adjacency matrix* like AGM, but differs in extending the *adjacency matrix* by edge addition.

gSpan represents a graph as a DFS (Depth First Search) tree, and extracts frequent subgraphs by right-most expansion of the tree.

GASTON extracts frequent patterns by using level-wise approach in which first simple paths are considered, then more complex trees and finally the most complex cyclic graphs. GASTON handles the very sparse data and extracts frequent subgraphs very fast.

2.4 Pattern mining method considering time interval

Hirate *et al* [7] proposed the sequential pattern mining method which extracts patterns taking into account the interval of the time each item occurred, by calculating and discretizing the difference from the time the first item occurred.

This approach treats the time as a distance of items, and in the case applying to WUM, it suits to evaluate the achievement of the conversion on marketing etc.

3 Proposed Method

3.1 Problem limitation

One of the challenging issues on WUM is the existence of the proxy server and cache. For example, when a user pushed the *backward* button, the requested page is loaded from either proxy server or local cached page of the user's client PC. In this case, no request is sent to the web server and recorded on web access logs. As a result, user's transition path cannot be reconstructed precisely only from the web access logs.

To cope with this problem, the solution using Cookies or Remote Agents is presented [8]. Writing scripts to gain the necessary information into web page files is another solution. These solutions will enable more precise analysis.

But even if the user behavior information inferred from web server log is not the accurate user behavior, web server log should not be denied as an important source of information. In this paper, we describe the way to apply the proposed method to the server logs.

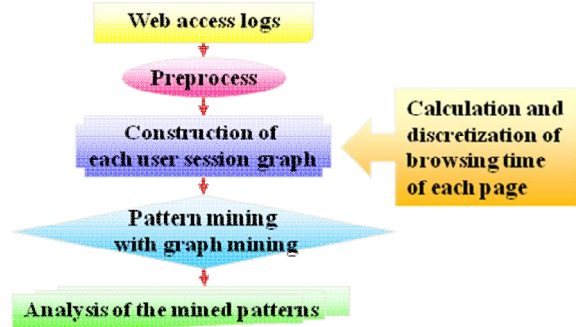


Figure 1. Process of the proposed method

3.2 Process of the proposed method

Figure 1 shows the process of our proposed method.

First, in the preprocessing phase, irrelevant information is removed and user sessions are identified [8]. Second, in the graph construction phase, browsing time of each page is calculated and discretized, and each user session is represented in a graph structure. We will detail this phase in section 3.3. Then, by applying a graph mining algorithm, frequent subgraphs are extracted as users' access patterns. Finally, administrators analyze the mined patterns.

In the proposed method, a *frequent* pattern means that the pattern has the *support* (the percentage of the sessions which includes the pattern against all sessions) greater than or equal to the predefined the *minimum support*.

3.3 Construction of each user session graph

In the graph construction phase, first, browsing time of each page is calculated and discretized according to the length, and then, each user session graph is constructed.

Browsing time t_b of a certain page P is assumed to be the period of time with the longest time difference between the request time t_r of the request which includes P as a referrer (*Ref*, the page that originally generated the request for the current page [2]) and another t_r of the request which includes P as a requested page (*Req*), until P is requested next or until the end of the session if P is not requested again. If the request which includes P as *Ref* does not exist in the session after the P is requested, t_b is assumed to be *null*.

After that, the calculated browsing time is

discretized according to the length. In section 3.4, we will describe the reason why the discretization is needed. To discretize the browsing time, we introduce the *weighting function* $w(P, t_b)$. Let T_b be the minimum browsing time the administrators arbitrarily define, $f(t_b, T_b)$ a decision function to judge if the browsing time is shorter than T_b or not. The *weighting function* $w(P, t_b)$ is given by the equation (1).

$$w(P, t_b) = \begin{cases} 0 & \text{where } t_b \neq \text{null and } f(t_b, T_b) = \text{true} \\ 1 & \text{where } t_b \neq \text{null and } f(t_b, T_b) = \text{false} \\ 2 & \text{where } t_b = \text{null} \end{cases} \quad (1)$$

If t_b is not *null*, the *weight* 0 is given when $f(t_b, T_b)$ is true, and the *weight* 1 is given when $f(t_b, T_b)$ is false. If t_b is *null*, the *weight* 2 is given to the page which does not exist as *Ref* in the session (called an *exit page*). Browsing time of an *exit page* cannot be calculated, and the page has an important meaning that the user doesn't move to another page from the *exit page*. So, we give a special *weight* to this kind of pages to distinguish from the others. $f(t_b, T_b)$ can be arbitrarily defined depending on the web site management policies.

When constructing the graph representing each user session, a pair of a page and the *weight* is treated as a node. In the case that both the page and *weight* is the same respectively, the pairs are treated as the same node. And an edge connects each *Ref* and *Req*. If there are several requests that connect the same nodes, they are treated as only one edge. Thus, a constructed graph of each session includes a cyclic undirected graph.

3.4 Discretization problem

We assume to use the existing graph mining algorithm. However, the existing algorithm considers only the substructures, and cannot deal with the numerical values like browsing time. Therefore, we first discretize each browsing time by introducing the *weighting function* $w(P, t_b)$, label each node with discretized browsing time and corresponding page. This preprocessing enables us to apply the existing graph mining algorithm to the constructed user session graphs, still taking browsing time into consideration.

In this paper, we discussed the simplest ex-

ample of $w(P, t_b)$, where the *weight* is given only depending on whether each browsing time is longer or shorter than the threshold which administrators assumed. The proposed method will work flexibly for more complicated cases by introducing a more generalized decision function and *weighting function*.

3.5 Consideration about examples

In the example of the section 2.2, samples of the patterns, which are expected to be extracted by the proposed method, will be as shown in Figure 2. In the proposed method, even if the pages and subgraph structures included in the mined patterns are the same, by the difference of the length of page browsing time, it become possible to distinguish the patterns and extract as the different patterns.

From the patterns of Figure 2, for example, web site administrators can analyze as follows. In the pattern (1), it seems that browsing time of the page b is short, and users passed b and exited from the page d . In the pattern (2), it seems that browsing time of b is long, and users moved another page after having moved to d from b . If the administrators want users to move to the page further than d , one of the solutions is to improve b to give users more about the contents of b easily.

Thus, with the patterns mined by the proposed method, administrators can correlate the patterns with browsing time, and analyze more in detail.

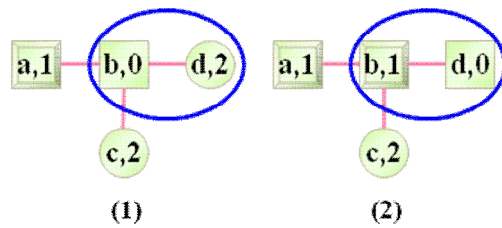


Figure 2. Samples of access patterns extracted by the proposed method (Each node indicates a web page and its *weight*, and each edge indicates a transition path. A square, double square, and circle nodes mean the *weight* 0, 1, and 2, respectively. The circled area is different from each other.)

4 Experiment and Evaluation

4.1 Contents and purposes

To evaluate the proposed method, we conduct the experiment on the real web access logs. About the cases where the *weight* is added to each page or not, we compare the number of the mined patterns, and confirm that it become possible to distinguish the patterns originally treated as the same, by weighting each page. Furthermore, through the inspection, we explain the patterns mined by the proposed method can give more information than the patterns mined by the existing method.

4.2 Environment and configuration

Table 2 shows the machine specification used for the experiment.

On the experiment, each session was identified by a timeout of 30 minutes.

About the *weighting function*, we defined $f(t_b, T_b)$ as follows;

$$\left. \begin{array}{l} T_b = 30 \text{ [sec]} \\ f(t_b, T_b) = \begin{cases} \text{true if } t_b \leq T_b \\ \text{false otherwise} \end{cases} \end{array} \right\} (2)$$

Under these conditions, we applied the proposed method to the real web access logs mentioned in section 4.3, while changing the *minimum support*. As the graph mining algorithm, we adopted GASTON.

4.3 Web access log data

On the experiment, we used the web access log data provided from Sendai City Industrial Promotion Organization, and KDDI R&D Laboratory.

The summary of each data is shown in Table 3.

4.4 Comparison of the number of mined patterns

We compared the number of the mined patterns.

First, about Sendai City Industrial Promotion Organization's web access log (Figure 3), in the case where the *minimum support* is 1% or less,

Table 2. A specification of the test machine

OS	Microsoft Windows XP Professional SP2
CPU	Intel Core 2 Duo 1.80GHz
Memory	1.99GB

Table 3. Web access log data used on the experiment

Sendai City Industrial Promotion Organization http://www.siip.city.sendai.jp/	
Term	01/Sep/2006 –01/Mar/2007
Original data size	About 1.94GB
Size after the preprocess	About 92.7MB
Number of all requests	9,440,870
Number of sessions	96196

KDDI R&D Laboratory http://www.kddilabs.jp/	
Term	01/Oct/2005 –27/Feb/2007
Original data size	About 827MB
Size after the preprocess	About 252MB
Number of all requests	5,057,390
Number of sessions	21189

more *weighted* patterns were extracted than *unweighted* ones. This is because, the pages, which are treated as the same node in the case the *weight* is NOT added, were distinguished by *weighting*, and the patterns which include these pages were extracted as different transition patterns. Also, in larger values of the *minimum support* less *weighted* patterns were mined because each page distinguished by *weighting* became unable to satisfy the large *minimum support*.

On the other hand, in the case of KDDI R&D Laboratory's web access log (Figure 4.), less *weighted* patterns were mined as a whole. This is why the cases, where all the pages distinguished by *weighting* became unable to satisfy the *minimum support*, increased.

Thus, by the comparison of the number of the mined patterns, it is shown that the proposed method can distinguish and extract the different transition patterns considering page browsing time.

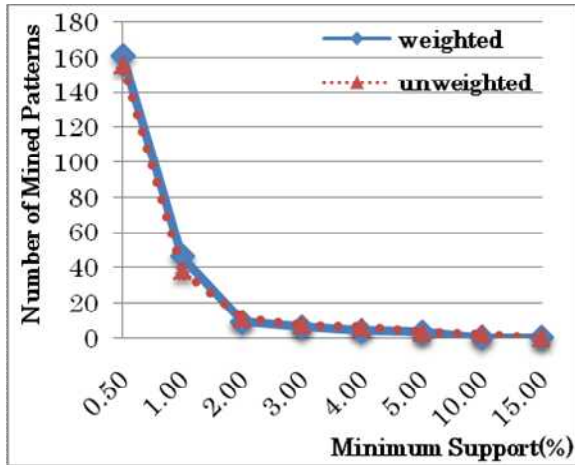


Figure 3. Comparison of the number of mined patterns (Sendai City Industrial Promotion Organization)

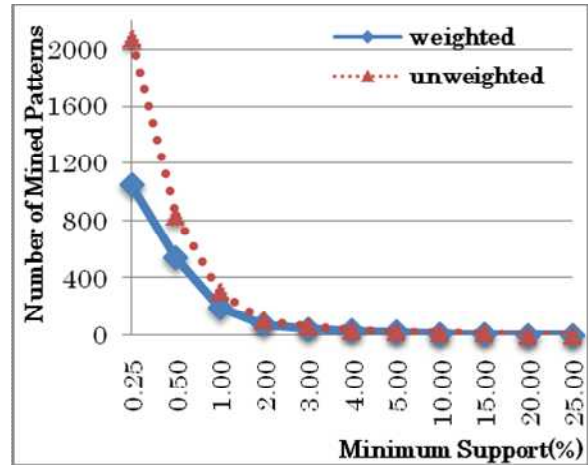


Figure 4. Comparison of the number of mined patterns (KDDI R&D Laboratory)

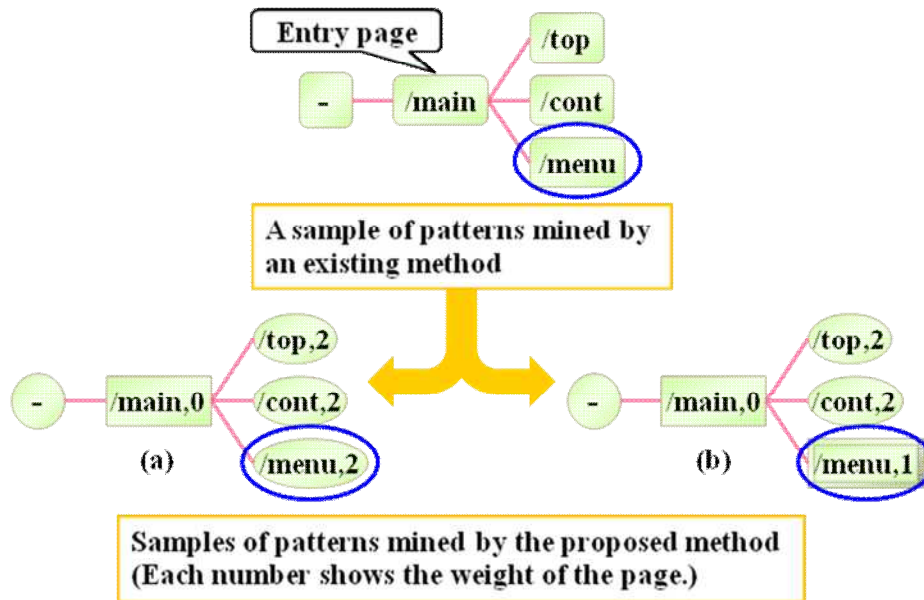


Figure 5. Comparison of the samples of mined patterns from KDDI R&D Laboratory (Each node indicates a web page and its *weight*, and each edge indicates a transition path. A square, double square, and circle nodes mean the *weight* 0, 1, and 2, respectively. The circled area is different from each other.)

4.5 Consideration of mined patterns

Actually extracted patterns are shown in Figure 5. These are the samples of the mined results from KDDI R&D Laboratory's web access log. The “-“ means that users accessed from the outside of the web site, that is, the page next to “-“ is the entry page (the first page in the visit regardless of

how the sessions are calculated [2]). In Figure 5, all the information administrators can obtain from the pattern mined by the existing method is that, the frequency of the access to each page and that of the use of each link in the pattern are high.

On the other hand, more information can be obtained from the patterns mined by the proposed method. For example, by the comparison of the pattern (a) and (b), administrators can see that, in

the case where browsing time of the /menu page is long, users move to another page. We confirmed that the /menu page gathered the links to the other pages. Generally, these types of page are provided to navigate users quickly, so browsing time of such pages should be short. However, the patterns of the results suggests that the /menu page wasn't able to navigate users to the other page, or spent longer time than expected. Thus, administrators can conclude that some improvement is needed to the /menu page.

Moreover, patterns containing exit pages as in Figure 5 can be treated in correlation with the page exit ratio (Number of exits from a page divided by total number of page views of that page [2]) obtained by statistical analysis. The proposed method can be used for the analysis of user's behavior exiting from the web sites.

As a result, by introducing the proposed method, administrators can analyze web access logs more in detail than by using the statistical and existing WUM method independently.

5 Conclusion and Future Works

This paper proposed the WUM method which *weights* each page according to the length of browsing time, and extracts access patterns by graph mining. Through the experiments with the two real web access logs, we confirmed that the proposed method can extract meaningful patterns by considering browsing time and make it possible for web site administrators to analyze users' behavior more in detail by integrating the statistical and existing WUM methods.

In our current proposed method, we constructed each user session as an undirected graph, but it is expected to mine more informative patterns by representing the session as a directed graph which shows the direction users moved between pages. Termier *et al* [9] reported the directed acyclic graph mining method. It would be possible to apply Termier's algorithm to extend our method.

In this paper, we assumed to use the existing graph mining algorithms and made it possible by the discretization of page browsing time with the *weighting function*. On this occasion, to simplify the problem, the same *weighting function* is applied to all pages. Actually, the features of each page differ from each other, and consequently it is useful to categorize the pages which have

similar features, and apply the *weighting function* suited to each category.

Since the goal of the proposed method is to assist web site administrators to improve their web site, it is desirable to choose more useful patterns for the administrators from all the mined patterns, and visualize the chosen patterns in easy-to-understand presentation.

Therefore, studies related to element technologies to realize this are needed. For example, the means that discover the patterns which has substructures similar to each other, or the metrics that estimate the usefulness of the patterns except for the *support* should be established. Moreover, a visualization method to make it easier to correlate the results of the statistical and WUM methods, which is the concept of our proposed method, is also needed. As future works, we studies to solve these challenges.

Acknowledgment

We are grateful to Sendai City Industrial Promotion Organization and KDDI R&D Laboratory for their support of web access log analysis.

References

- [1] R. Ivánczy and I. Vajk. Frequent Pattern Mining in Web Log Data. Acta Polytechnica Hungaria, Journal of Applied Sciences at Budapest Tech Hungary, Special Issue on Computational Intelligence. Vol.4, No.1, pp.77-99, 2006.
- [2] The Web Analytics Association (WAA). Web Analytics Definitions – Version 4.0. <http://www.webanalyticsassociation.org/>, 2007.
- [3] A. Inokuchi, T. Washio and H. Motoda. An Apriori-Based Algorithm for Mining Frequent Substructures from Graph Data. *Proc. of the 4th European Conference on Principles of Data Mining and Knowledge discovery*. pp.12-23, 2000.
- [4] M. Kuramochi and G. Karypis. Frequent Subgraph Discovery. *Proc. of the 1st International Conference on Data Mining*. pp.313-320, 2001.
- [5] X. Yan and J. Han. gSpan: Graph-Based Substructure Pattern Mining. *Proc. of the 2nd International Conference on Data Mining*. pp.721-724, 2002.
- [6] S. Nijssen and J. N. Kok. A Quickstart in Frequent Structure Mining Can Make a Differ-

ence. *Proc. of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. pp.647-652, 2004.

[7] Y. Hirate and H. Yamana. On generalizing of Sequential Pattern Mining with Time Intervals. *Proc. of Data Engineering Workshop*. 2006.

[8] R. Cooley, B. Mobasher and J. Srivastava. Data Preparation for Mining World Wide Web Browsing Patterns. *Knowledge and Information Systems*. Vol.1, No.1, pp.5-32, 1999.

[9] A. Termier, Y. Tamada, S. Imoto, T. Washio, and T.Higuchi. From closed tree mining towards closed DAG mining. *Proc. of the International Workshop on Data Mining and Statistical Science*, pp.1-7, 2006.