A NOVEL WEB USAGE MINING METHOD
Mining and Clustering of DAG Access Patterns Considering Page Browsing Time

Koichiro Mihara, Masahiro Terabe and Kazuo Hashimoto
Graduate School of Information Sciences, Tohoku University, Sendai, Japan

Keywords: Web access log analysis, Web Usage Mining, Page browsing time, Closed Frequent Embedded DAG Mining, Clustering.

Abstract: In this paper, we propose a novel method to analyze web access logs. The proposed method defines a web access pattern as a DAG with page browsing time, and extracts the patterns using the closed frequent embedded DAG mining algorithm, DIGDAG. The proposed method succeeds in extracting as small number of patterns as necessary minimum, and enables more efficient analysis by clustering the extracted results.

1 INTRODUCTION

Due to the rapid spread of WWW techniques, web sites on the Internet have been so commonly looked up in the human daily life. Administrators of the web sites which have a variety of contents and complex link structures are constantly required to maintain their sites as easy to use as possible. Thus, administrators need to understand how their sites are used, guessing users' background needs and demands.

To do this, web access logs have been recognized as an important source of information. A web access log is a time-series record of users' requests, each of which is sent to a web server whenever a user does some operation such as clicking a link on a web page. Web access log analysis is very useful for administrators to understand users' behavior on the web site (Tec-Ed, 1999; Burton and Walther, 2001).

Statistical methods, such as Google Analytics (http://www.google.com/analytics/), are widely used to analyze the logs in terms of page views, page exit ratio, visit duration, etc.. Administrators analyze the features and tendency of usage for each page, based on these information. However, as the statistical method is limited to a local analysis of each page, the analysis results are likely to become fragmentary.

Web Usage Mining (WUM) is an emerging attempt to apply Data Mining (DM) techniques for web access log analysis (Raymond and Hendrik, 2000; Liu, 2006; Srivastava et al., 2000). WUM extracts regularities of users' access behavior as patterns, which are defined by combinations, orders or structures of the pages accessed in a session. WUM enables administrators to overview the features of web site usage with providing more comprehensive analysis than statistical methods. But, the results of WUM don't contain quantitative information as obtained by statistical analysis, and local analysis is difficult. Moreover, it is often the case that WUM extracts too many patterns for administrators to analyze.

This paper proposes a method to analyze web access logs by extracting the access patterns containing browsing time of each page, and associating local analysis of statistical methods with comprehensive analysis of WUM. The proposed method reduces the number of extracted patterns by closed pattern mining and clusters similar patterns for efficient analysis.

This paper is organized as follows. In the section 2, we clarify the target problem by reviewing the related works. After explaining the procedure of our proposed method in the section 3, we discuss the analysis with the proposed method in the section 4. Finally, in the section 5, we conclude the paper with an indication of future works.

2 BACKGROUND AND MOTIVATION OF OUR STUDY

2.1 Web Access Log Analysis

Web access log analysis is to analyze the patterns of web site usage and the features of users' behavior,

Mihara K., Terabe M. and Hashimoto K. (2008).
A NOVEL WEB USAGE MINING METHOD - Mining and Clustering of DAG Access Patterns Considering Page Browsing Time.
In *Proceedings of the Fourth International Conference on Web Information Systems and Technologies*, pages 313-320
DOI: 10.5220/0001528303130320

based on web access logs which are the records of users' access to the site (Tec-Ed, 1999; Burton and Walther, 2001). Web site administrators design their sites according to their intent and purpose. Getting users stay long time in the web site, see as many pages as possible, or buy some products is an example. Web access log analysis is a method to verify the achievement of the purpose (Nakayama et al., 2000). If the sites are used in different ways from what administrators have assumed, administrators have to recognize users' needs from the results, and redesign the sites structurally or visually to improve the usability.

In most cases, web access logs are stored as web server logs. On analysis of server logs, one of the challenging issues is the existence of the proxy server and cache. For example, when a user moves to a page by the *backward* button, the requested page is loaded from either proxy server or locally cached page of the user's client PC. In this case, no request is sent to the web server and recorded on the log. As a result, users' transition path cannot be reconstructed precisely only from the server logs. To cope with this problem, the solutions using Cookies or Remote Agents are proposed (Cooley et al., 1999). Writing scripts to gain the necessary information into web page files is another solution. These solutions will enable more precise analysis. But, web server logs remain to be an important source of information to infer users' behavior due to the handiness for administrators.

Recently, web access log analysis has caught much attention on e-business (Draheim et al., 2005), and a lot of tools to analyze the logs are developed. There are various functions in each tool, but the methods adopted by such tools are mainly divided into two categories. One is statistical analysis, and another is data mining based analysis. Though administrators analyze web access logs by using both methods, the analyzed results of them are independent of and hard to relate to each other. In the following, we summarize the features and issues of each analysis method, and describe the motivation of our study.

subsectionStatistical Analysis Web access log analysis tools, such as Analog (http://www.analog.cx/) and Google Analytics (http://www.google.com/analytics/), are widely used to analyze the logs statistically. The statistical analysis estimates page views (the number of times a page was viewed), visit duration (the length of time in a session), page exit ratio (number of exits from a page divided by total number of page views of that page), bounce rate (single page view visits divided by entry pages) etc. (Burby et al., 2007). In addition, search words or referrers (the referrer is the page URL that originally generated the request for the current page view or object (Burby et al., 2007)) can be obtained.

The statistical analysis introduces a set of parameters (statistical indices) to describe users' access behavior. With those parameters, it becomes easy for administrators to define concrete goals for organizing their web sites and improve the sites according to the goals. For example, consider page browsing time (the length of time spent on a page) (Hofgesang, 2006). The pages, such as a sitemap, are to navigate users to another page quickly, so browsing time of such pages have to be short. By contrast, the pages, such as warning, caution, agreement, etc. have important information, and need to be read carefully. Thus, browsing time should be sufficiently long. When analyzing the logs, if browsing time of a page is different from what administrators have assumed, administrators can redesign the page based on the analysis results.

But there are two major issues to be concerned. One is that the statistical analysis produces only fragmentary information in the sense that the analysis results are independent page by page. Although the statistical analysis provides a variety of features of each page usage, it cannot say anything about the relationships among several pages from the viewpoint of users' browsing behavior. As for browsing time, if the browsing time of a page is different from what administrators have assumed, the reason may be found in another page. In such a case, the statistical analysis is not sufficient to solve the problem. Another issue is that the time-dependent variables like browsing time are treated as averaged values. As users' browsing behavior is expected to be different depending on the length of browsing time, more flexible treatment of browsing time should be introduced rather than an averaged value. Thus, by solving these issues, it becomes possible for administrators to take more efficient measures to improve their web sites.

2.2 Web Usage Mining

Web Usage Mining (WUM) includes the analysis and prediction of users' behavior in the web site, by applying Data Mining (DM) techniques to web access logs (Raymond and Hendrik, 2000). Various DM techniques, such as association rules, classification, sequential pattern mining, etc. are applied according to administrators' purpose (Liu, 2006; Srivastava et al., 2000; Bose et al., 2006).

The main feature of WUM is that WUM enables analysis on the regularity of users' access, by representing users' behavior as the combinations, orders or structures of the pages users accessed, and extracting frequent patterns from logs (Iváncsy and Vajk, 2006). A pattern *Ptn* is *frequent*, if the pattern ap-

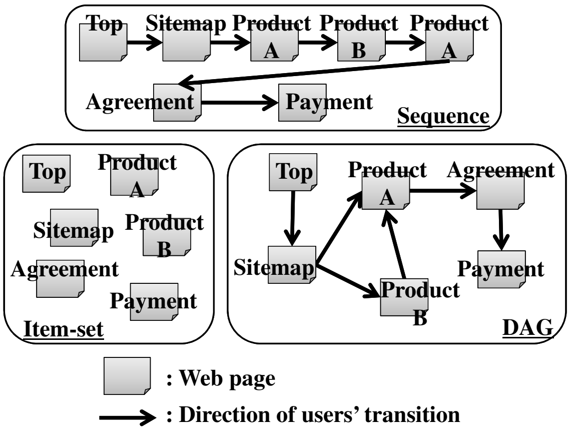

Figure 1: Examples of access patterns obtained by WUM.

pears in the logs with larger frequency than the minimum frequency ε(≥ 0) given by administrators in advance. In this case, the frequency is also called the *support*. There are many methods to obtain frequent patterns, such as item-set mining (Agrawal and Srikant, 1994; Su and Lin, 2004), sequential pattern mining (Srikant and Agrawal, 1996; Pei et al., 2001; Ayres et al., 2002; Zaki, 2001; Yang et al., 2005), tree mining (Asai et al., 2002; Asai et al., 2003; Chi et al., 2004), graph mining (Inokuchi et al., 2000; Kuramochi and Karypis, 2001; Yan and Han, 2002; Nijssen and Kok, 2004) and so on. As for tree mining and graph mining, the methods to extract *embedded* patterns are proposed, considering not only the parent-child relationship but also the ancestor-descendant relationship of vertices (Zaki, 2002; Termier et al., 2007). The definition of embedded patterns is mentioned in each paper.

Figure 1 shows the examples of access patterns obtained by WUM. From item-set patterns, regardless of the link structure, administrators can find out the relationships among the pages users accessed in a session. Sequential patterns of users' page transition path are useful for the prediction of the page expected to be accessed next. And especially, in the case where access patterns are extracted as graph structures, the graph patterns serve as a useful reference to reconstruct the web site, because the patterns are similar to the web sites' link structure. Besides, on a shopping site or auction site etc., there is the case where a user browse more than one page at the same time to compare several products. In such a case, the user's transition path is *branched*, and the case can be represented by a graph pattern. Thus, WUM can represent the relationships among several pages as access patterns to allow administrators to overview users' behavior.

However, WUM also has some issues to be fixed. One is that extracted access patterns don't contain quantitative information as obtained by the statistical analysis. As in the section 2.1, considering page browsing time, if users access the same page through the same path, the access patterns become the same, even if browsing time of the page is different. In other words, the patterns which should be distinguished are treated as the same, and thus, administrators cannot analyze the users' behavior accurately. Another issue is that WUM produces too many access patterns for administrators to analyze. This is because all the patterns which satisfy the minimum support are extracted as analysis results. The number of patterns can be reduced by setting the minimum support to a large value, but the risk of missing useful patterns increases. So it is reasonable to set appropriately smaller value to the minimum support, which inevitably causes a large number of patterns. Therefore, there is a need to assist the analysis of a large number of patterns efficiently.

2.3 Motivation and Goals

As mentioned above, there are issues that the statistical analysis cannot represent the relationships among several pages, or the time-dependent variables. On the other hand, there is an issue that the results of WUM don't include quantitative information such as page browsing time. For more detailed web access log analysis, it is essential but difficult to associate both results of statistical analysis and WUM, because the results of those methods are independent of each other. Thus, we define a goal, that is, to establish a method to extract the access patterns which contain quantitative information to combine the advantages of both methods. On this occasion, we focus on page browsing time. We have described above that a variety of improvements for web sites can be considered by page browsing time. By extracting the access patterns containing page browsing time, it become possible to analyze the users' page transition in association with the difference of browsing time. As a related work, about sequential pattern mining, the method which considers the time interval of items bas been proposed (Hirate and Yamana, 2006). But this method is specialized in sequential pattern mining, and cannot been applied to structured patterns like graph patterns.

Moreover, there is another issue on WUM that it extracts too many patterns to evaluate. As browsing time is introduced to the access patterns which are obtained by the existing WUM, the resulting patterns are segmentalized and increase in number. Some assistance measure should be taken for efficient analysis. So, we also add the following goals, 1) to minimize the number of patterns to be extracted, 2) to group similar access patterns.

To minimize the number of patterns, closed pattern mining methods have been proposed (Uno et al., 2004; Xia and Yang, 2005; Termier et al., 2004; Yan and Han, 2003; Termier et al., 2007). A pattern *Ptn* is *closed*, when there is no pattern Ptn' which contains *Ptn* as a sub-set for its support. By general frequent pattern mining, if *Ptn* is frequent, the sub-set of *Ptn* is also extracted as frequent patterns. But, *Ptn* is sufficient to analyze the mined results. Namely, closed pattern mining can omit redundant patterns.

To group similar patterns, clustering is useful. In some aspects, applying clustering to mined patterns means that the features of patterns similar to or different from other patterns are extracted. Although, when analyzing the mined patterns one by one, administrators have to find the features by themselves, clustering can improve the efficiency.

With the motivation mentioned above, we propose the method to realize these two purposes, 1) the extraction of the access patterns containing page browsing time, 2) the reduction of patterns by closed pattern mining and clustering of the mined patterns.

3 PROPOSED METHOD

3.1 Process of the Proposed Method

The process of our proposed method consists of five phases. First is the preprocessing phase. Irrelevant information is removed by cleaning, and each user session is identified (Cooley et al., 1999). Next, in the user session DAG construction phase, browsing time of each page is calculated and discretized, and then a user's page transition in each session is represented as a Directed Acyclic Graph (DAG) structure. We describe the detail in the section 3.2. All sessions are represented as DAGs, then, in the pattern extraction phase, access patterns are extracted from the set of the DAGs. To mine the patterns, we adopt the Closed Frequent Embedded DAG (c.f.e-DAG) mining algorithm, DIGDAG (Termier et al., 2007). Fourth is the clustering phase. By clustering the set of extracted access patterns, similar patterns are grouped for the easiness of later analysis. We describe the mining and clustering of access patterns in the section 3.3. And the last is the pattern analysis phase, in which administrators analyze each pattern for each cluster.

3.2 Construction of Each User Session DAG

To construct user session DAGs considering page browsing time, what to do first is the calculation of browsing time of each page. As mentioned in the section 2.1, though, in the case where logs are obtained by the scripts written in each web page file, browsing time is measured more accurately, we consider the case of web server logs in this paper.

Browsing time t_{PgB} of a certain page Pg is assumed to be the period of time with the longest time difference between the request time t_{PgR} of the request which includes Pg as a referrer (Ref) and another t_{PgR} of the request which includes Pg as a requested page (Req), until Pg is requested next or until the end of the session if Pg is not requested again. If the request which includes Pg as Ref does not exist in the session after Pg is requested, t_{PgR} is assumed to be *null*. If the same page is requested several times, browsing time is calculated for each request.

After that, the calculated browsing time is discretized according to the length, and given to each page as the *weight* which denotes the length of browsing time. To do this, we introduce the weighting function, $w(Pg, t_{PgB})$. In the section 4.1, we describe the reason why the discretization is needed.

$$w(Pg, t_{PgB}) = \begin{cases} 0 & t_{PgB} \neq null \text{ and } t_{PgB} < T_{min} \\ 1 & t_{PgB} \neq null \text{ and } T_{min} \leq t_{PgB} \leq T_{max} \\ 2 & t_{PgB} \neq null \text{ and } T_{max} < t_{PgB} \\ 3 & t_{PgB} = null \end{cases} \tag{1}$$

The length of browsing time administrators assume is defined by T_{min} and T_{max}, and by applying the weighting function, $w(Pg, t_{PgB})$, the validity of browsing time t_{PgB} of a page Pg is evaluated. If t_{PgB} is shorter than the minimum browsing time T_{min} defined by administrators, the weight 0 which means browsing time is *too short* is given to Pg. T_{min} is given as a measure to judge a user didn't read and passed the page. By contrast, if t_{PgB} is longer than the maximum browsing time T_{max} also defined by administrators, the weight 2 which means browsing time is *too long* is given to Pg. T_{max} is defined as a measure to judge a user accessed but left the page. The weight 1 is given if t_{PgB} is within the length administrators assume and judged to be *valid*. And the weight 3 is given if Pg doesn't exist as Ref in the session. Pg given the weight 3 is called the *end page* in this paper. About the end page, browsing time cannot be calculated, and the end page indicates that users didn't move to any other page from the page. So the end page should be distinguished from the other pages. T_{min}, T_{max} can be defined arbitrarily before analysis.

Thus, after weighting each page according to browsing time, a DAG structure is constructed for each user session. The definition of DAGs is given in (Termier et al., 2007). In the user session DAGs, each vertex is labeled by a tuple of a page and its weight. That is, each vertex is represented by $(Pg, w(Pg, t_{PgB}))$. Each edge connects vertices in the direction from Ref to Req for each request. Edges show users' page transition, and only the direction is considered. If there is the cyclic structure, that is, the path which starts from $(Pg, w(Pg, t_{PgB}))$ and ends at the same $(Pg, w(Pg, t_{PgB}))$, another vertex which indicates $(Pg, w(Pg, t_{PgB}))$ is created and the cyclic structure is converted into the acyclic structure. By this procedure, user session DAGs containing quantitative information of browsing time can be constructed.

3.3 Mining and Clustering Patterns

To mine the patterns, we adopt the c.f.e-DAG mining algorithm, DIGDAG (Termier et al., 2007). DIGDAG replace the closed frequent DAG mining problem with the problem of closed frequent item-set mining on edges (Uno et al., 2004), with the restriction that all the labels of vertices in a DAG must be distinct. And by the reconstruction of DAG structures from the mined closed frequent edge set, closed frequent DAGs are obtained. DIGDAG also can extract the embedded DAGs based on not only the parent-child relationship but also the ancestor-descendant relationship of vertices. When mining the patterns, administrators give DIGDAG the user session DAG set and the minimum support $\varepsilon(\geq 0)$ as inputs. By doing this, access patterns are obtained as c.f.e-DAGs.

Then, the next step is clustering of the mined patterns. One of our purposes is to analyze users' behavior in association with browsing time of each page and page transition. So, when clustering, we don't consider the weight of each page, but focus on page transition and obtain the clusters for each feature of page transition. Thus, the patterns which include similar page transition are grouped, and it becomes easier to analyze the differences of users' behavior in association with browsing time. For clustering, the similarity of the patterns is to be estimated. The way to estimate the similarity of graphs based on the labels of vertices and the structure of edges is introduced (Bhattacharya and Getoor, 2006). Also a variety of clustering algorithms have been proposed (Berkhin, 2002), and a tool which implements them to cluster graphs is developed (Recupero and Shasha, 2007).

4 DISCUSSION

4.1 Discretization Problem

We assume to use the existing algorithm, DIGDAG. However, DIGDAG considers only the sub-structures, and cannot treat the numerical values like browsing time. Therefore, we discretize each browsing time by introducing the weighting function, $w(Pg, t_{PgB})$, and label each vertex with discretized browsing time and corresponding page. This process enables us to construct user session DAGs considering browsing time.

In this paper, we assumed the simplest form of the weighting function, $w(Pg, t_{PgB})$, such that the weight is given only depending on whether browsing time of each page is longer or shorter than the threshold that administrators give. This simplification is introduced in order to make it easy to compare extracted patterns with the expected access behaviors. There are some cases where administrators have to investigate the association of browsing time and access patterns in finer granularity. In such a case, the weighting function should be redefined by including additional parameters other than T_{min}, T_{max}.

Also, because browsing time depends on the number of contents on the page to some extent (if users read all the contents on a page, it is obvious that the more the number of contents are, the longer browsing time is), T_{min} and T_{max} are supposed be defined for each page. But on the large web site it is very difficult, so some compromises are necessary. One solution is, by categorizing the pages in advance based on the number or the properties of the contents etc., to define T_{min} and T_{max} for each category.

4.2 Objective Patterns

Let us compare the patterns of the cases where browsing time is considered or not. Figure 2 shows the examples assuming web access logs of a shopping site. The pattern (a) is that of the case where browsing time is NOT considered, and the pattern (b) contains browsing time.

All the information administrators can see in the pattern (a) is that users moved to each page through the paths contained in the pattern. This information is not sufficient for administrators to recognize whether the sitemap page can navigate users quickly, or whether users purchased the product A after reading the contents of the agreement page carefully, etc.

On the other hand, with the pattern (b), administrators can see whether browsing time is valid or not, by the weight of each page. For example, browsing time of the sitemap page is short (the weight is 0), so

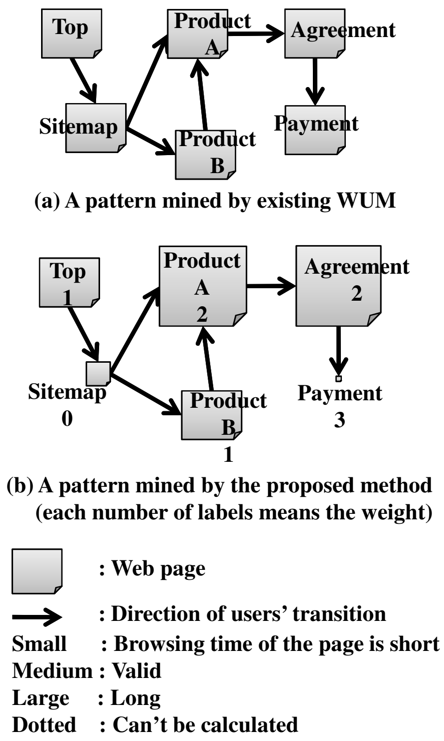
Top
Product A
Agreement
Sitemap
Product B
Payment
(a) A pattern mined by existing WUM
Top 1
Product A 2
Agreement 2
Sitemap 0
Product B 1
Payment 3
(b) A pattern mined by the proposed method (each number of labels means the weight)
: Web page
: Direction of users' transition
Small : Browsing time of the page is short
Medium : Valid
Large : Long
Dotted : Can't be calculated

Figure 2: Comparison of the cases in which page browsing time is considered or not.

the page works well to navigate users. Also, it is recognizable that the agreement page is read carefully by users when the product A is bought.

Thus, the access patterns obtained by the proposed method are more informative than that without considering page browsing time and make it possible to analyze users' page transition in association with browsing time of each page.

4.3 Pattern Clusters

Our proposed method clusters the mined patterns based on the similarity of users' page transition. Examples of clusters expected to be created are as shown in Figure 3. Actually, as the support of each pattern is given, it is possible to analyze the patterns taking into account the difference of the frequency. But in this section, we would focus on the analysis using the clusters. As access patterns in each cluster share the same features, by recognizing the features, administrators can understand the meanings of each pattern more easily.

For instance, the feature of the patterns in Cluster (1) is that users started from the top page, moved to the pages of the product A and B by using the sitemap page, and finally purchased the product A. Browsing time of the pages of the product A and B is long to some extent, so it is assumed that users compared the two products and chose the product A.

Cluster (2) is the cluster of the patterns in which users accessed the pages of the product A and another product from the top page, but purchased nothing. Considering that browsing time of both the top page and the page of the product A is short, it is likely that users visited the web site to purchase the product A, but eventually stopped it on the agreement page. Thus, it is assumed that there is some factor making users hesitate to buy products on the agreement page, and administrators can redesign the page to improve.

And in the patterns included in Cluster (3), users accessed the page of the product A through the page of the product C, and bought the product A. Comparing Cluster (3) with Cluster (1) and (2), it seems that users buy the product A when accessing the page of the product A through the page of the product B or C. Therefore, it is expected that the purchase of the product A is facilitated by making easier the comparison of the product A with the other products.

Administrators have to analyze the patterns respectively and it is so time-consuming, if the extracted patterns are simply presented without any processing. But as we mentioned in this section, by analyzing the features of each cluster and the differences from the other clusters, administrators can recognize the meanings of each access pattern more deeply. Thus, administrators can find out the problem of their web sites, develop courses to enrich the contents, reconstruct the link structures, and improve the visual design more efficiently.

5 CONCLUSIONS AND FUTURE WORKS

In web access log analysis, it is important to combine the analysis based on quantitative and local information by the statistical method with the comprehensive analysis of the relationship among several pages by WUM. In this paper, we proposed the method to analyze web access logs in detail by mining the DAG access patterns containing the information on browsing time of each page. The proposed method takes advantages of both the statistical analysis and WUM. Also, for efficient analysis on a large number of the extracted patterns, we proposed the reduction of patterns by closed pattern mining, and the clustering of similar patterns. And we showed the examples of access patterns and clusters expected to be obtained by our proposed method. Through the description of those examples, we showed that web site administra-

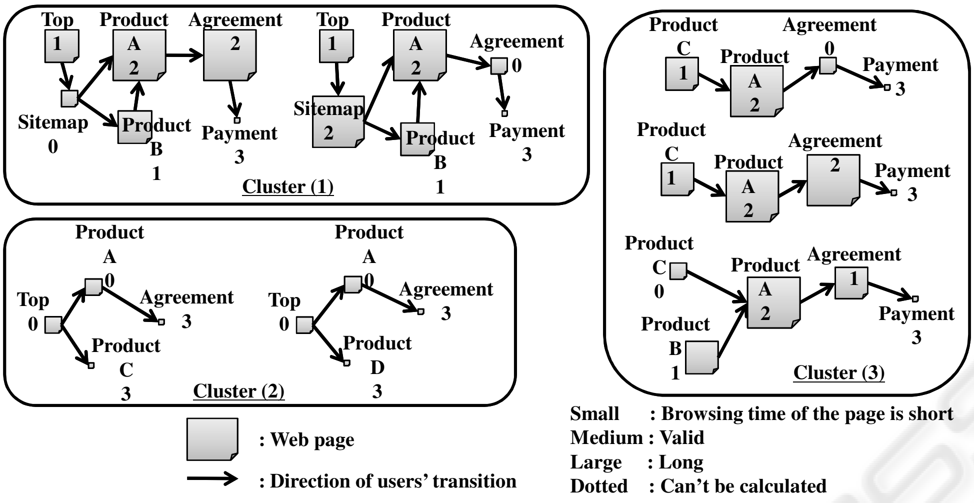

Figure 3: Clusters of mined patterns (each number of labels means the weight).

tors could find out the problems of their web sites efficiently, and develop concrete plans to improve their sites more easily.

As future works, we consider as follows. First is the experiment to evaluate our method on real web access log data. By applying the procedure of our proposed method as mentioned in the section 3.1, we need to confirm that expected patterns and clusters can be obtain. Second is the detail examination of the weighting function, $w(Pg, t_{PgB})$. In this paper, to simplify the problem of discretization and analysis of browsing time, we defined the weighting function as Eq. 1 of the section 3.2. For finer discretization of browsing time, however, the number of the contents of each page should be considered as well as T_{min}, T_{max}. We should examine the different definition of the weighting function. Third is the visualization of the mined patterns and clusters. The purpose of our study is to assist administrators in analyzing web access logs as a clue to organize their web sites. Therefore, on the actual analysis, the obtained patterns and clusters need to be presented for administrators in an easy form to analyze. Addressing the logs of dynamic web sites which have been more and more popular recently, or acquiring more accurate browsing time are also important issues.

ACKNOWLEDGEMENTS

We would like to thank Prof. Takashi Washio and Assistant Prof. Akihiro Inokuchi in Osaka University for their kind advices on graph mining technologies. We also would like to thank Dr. Alexandre Termier in Université Joseph Fourier, for the introduction of his latest work on DIGDAG. And this work was partially supported by the Ministry of Education, Science, Sports and Culture, Grand-inAid for Scientific Research (B) 19360181, 2007.

REFERENCES

Agrawal, R. and Srikant, R. (1994). Fast algorithms for mining association rules. In *The 20th International Conference on Very Large Data Bases (VLDB)*, pages 487–499.

Asai, T., Abe, K., Kawasoe, S., Arimura, H., Sakamoto, H., and Arikawa, S. (2002). Efficient substructure discovery from large semi-structured data. In *SIAM International Conference on Data Mining*.

Asai, T., Arimura, H., Uno, T., and Nakano, S.-I. (2003). Discovering frequent substructures in large unordered trees. In *Discovery Science*, pages 47–61.

Ayres, J., Flannick, J., Gehrke, J., and Yiu, T. (2002). Sequential pattern mining using a bitmap representation. In *The 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '02)*, pages 429–435. ACM.

Berkhin, P. (2002). Survey of clustering data mining techniques. Technical report, Accrue Software.

Bhattacharya, I. and Getoor, L. (2006). *Entity Resolution in Graphs*, chapter Mining Graph Data (L. Holder and D. Cook, eds.). Wiley.

Bose, A., Beemanapalli, K., Srivastava, J., and Sahar, S. (2006). Incorporating concept hierarchies into usage mining based recommendations. In *WebKDD 2006: KDD Workshop on Web Mining and Web Usage Analysis, in conjunction with the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '06)*.

Burby, J., Brown, A., and WAA Standards Committee (2007). Web analytics definitions - version 4.0.

http://www.webanalyticsassociation.org/. Web Analytics Association.

Burton, M. C. and Walther, J. B. (2001). The value of web log data in use-based design and testing. *Computer-Mediated Communication*, 6(3).

Chi, Y., Yang, Y., and Muntz, R. R. (2004). Hybridtreeminer: An efficient algorithm for mining frequent rooted trees and free trees using canonical forms. *The 16th International Conference on Scientific and Statistical Database Management (SSDBM '04)*, 00:11.

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.

Draheim, M.-D., Hanser, C., and von Seckendorff, C. (2005). E-business case studies web log analysis: testberichte.de seminar paper.

Hirate, Y. and Yamana, H. (2006). Sequential pattern mining with time intervals. In Ng, W. K., Kitsuregawa, M., Li, J., and Chang, K., editors, *The 10th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD '06)*, volume 3918 of *Lecture Notes in Computer Science*, pages 775–779. Springer-Verlag New York, Inc.

Hofgesang, P. I. (2006). Relevance of time spent on web pages. In *WebKDD 2006: KDD Workshop on Web Mining and Web Usage Analysis, in conjunction with the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '06)*.

Inokuchi, A., Washio, T., and Motoda, H. (2000). An apriori-based algorithm for mining frequent substructures from graph data. In *The 4th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD '00)*, pages 13–23. Springer-Verlag New York, Inc.

Iváncsy, R. and Vajk, I. (2006). Frequent pattern mining in web log data. *Acta Polytechnica Hungarica, Journal of Applied Science at Budapest Tech Hungary, Special Issue on Computational Intelligence*, 3(1):77–90.

Kuramochi, M. and Karypis, G. (2001). Frequent subgraph discovery. In *The 2001 IEEE International Conference on Data Mining (ICDM '01)*, pages 313–320. IEEE Computer Society.

Liu, B. (2006). *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data (Data-Centric Systems and Applications)*. Springer-Verlag New York, Inc.

Nakayama, T., Kato, H., and Yamane, Y. (2000). Discovering the gap between web site designers' expectations and users' behavior. *Comput. Networks*, 33(1-6):811–822.

Nijssen, S. and Kok, J. N. (2004). A quickstart in frequent structure mining can make a difference. In *Tthe 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '04)*, pages 647–652. ACM.

Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., and Hsu, M.-C. (2001). PrefixSpan mining sequential patterns efficiently by prefix projected pattern growth. In *The 17th International Conference on Data Engineering (ICDE '01)*, pages 215–226. IEEE Computer Society.

Raymond, K. and Hendrik, B. (2000). Web mining research: A survey. *SIGKDD Explor. Newsl.*, 2(1):1–15.

Recupero, D. R. and Shasha, D. (2007). GraphClust. http://cs.nyu.edu/shasha/papers/GraphClust.html.

Srikant, R. and Agrawal, R. (1996). Mining sequential patterns: Generalizations and performance improvements. In Apers, P. M. G., Bouzeghoub, M., and Gardarin, G., editors, *The 5th International Conference on Extending Database Technology (EDBT)*, volume 1057, pages 3–17. Springer-Verlag New York, Inc.

Srivastava, J., Cooley, R., Deshpande, M., and Tan, P.-N. (2000). Web usage mining: Discovery and applications of usage patterns from web data. *SIGKDD Explorations*, 1(2):12–23.

Su, J.-H. and Lin, W.-Y. (2004). CBW: An efficient algorithm for frequent itemset mining. *The 37th Hawaii International Conference on System Sciences (HICSS '04)*, 3:30064.3.

Tec-Ed (1999). Assessing web site usability from server log files.

Termier, A., Rousset, M.-C., and Sebag, M. (2004). DRYADE: A new approach for discovering closed frequent trees in heterogeneous tree databases. In *The 4th IEEE International Conference on Data Mining (ICDM '04)*, pages 543–546. IEEE Computer Society.

Termier, A., Tamada, Y., Numata, K., Imoto, S., Washio, T., and Higuchi, T. (2007). DIGDAG, a first algorithm to mine closed frequent embedded sub-DAGs. In *The 5th International Workshop on Mining and Learning with Graphs (MLG '07)*.

Uno, T., Kiyomi, M., and Arimura, H. (2004). LCM ver. 2: Efficient mining algorithms for frequent/closed/maximal itemsets. In *Proceedings of the IEEE ICDM '04 Workshop on Frequent Itemset Mining Implementations (FIMI '04)*.

Xia, Y. and Yang, Y. (2005). Mining closed and maximal frequent subtrees from databases of labeled rooted trees. *IEEE Transactions on Knowledge and Data Engineering*, 17(2):190–202.

Yan, X. and Han, J. (2002). gspan: Graph-based substructure pattern mining. In *IThe 2002 IEEE International Conference on Data Mining (ICDM '02)*, pages 721–724. IEEE Computer Society.

Yan, X. and Han, J. (2003). Closegraph: mining closed frequent graph patterns. In *The 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '03)*, pages 286–295. ACM.

Yang, Z., Wang, Y., and Kitsuregawa, M. (2005). LAPIN: Effective sequential pattern mining algorithms by last position induction. Technical report, Tokyo University.

Zaki, M. J. (2001). SPADE: An efficient algorithm for mining frequent sequences. *Machine Learning*, 42(1/2):31–60.

Zaki, M. J. (2002). Efficiently mining frequent trees in a forest. In *The 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '02)*, pages 71–80. ACM.