# Discovering Usage Pattern for Web Recommendation System by using DHP Algorithm

Htet Htet San, Khin Mar Myo
*University of Computer Studies (Mawlamyine)*
*htethtetpyisone@gmail.com, kmmyo09@gmail.com*

## Abstract

*Today, Web becomes a large repository for knowledge discovery. However, how to find needed and related information from the Web is a big challenge for users. As a solution, Web personalization and recommendation technique have evolved. Web recommendation is considered as a process of identifying user's preference and adapting service to satisfy user's need based on referring the historical behavior of current user or others who share similar interest to this user. This paper describes how to develop a web recommendation system using Web Usage Mining. Firstly, e-library system is developed not only for recording log file but also for evaluating the effectiveness of recommendation system. Then the explanation of web log data preprocessing is followed. Secondly, how to discover the usage patterns from Web log files using Direct Hashing and Pruning (DHP) association rule mining algorithm is described. Finally, rule generation processes based on user input of threshold values is explained. These rules are used to provide co-occurrence pages as recommendation to user.*

## 1. Introduction

Since World Wide Web serves as a huge, widely distributed, global information service centre for every kind of information such as news, advertisements, consumer information, financial management, education, government, e-commerce, health services, and many other information service, it becomes more important to find the useful information from these huge amounts of data. Recommender system on Internet help people make decisions in this complex information space where the volume of information is available to them is very large [7].

Basically, Web recommendation is considered as the process of identifying user's preference and adapting service to satisfy user's need based on referring the historical behavior of current user or others who share similar interest to this user. There are some approaches and techniques commonly used in Web recommendation, namely content-based filtering, collaborative filtering, hybrid-based filtering, knowledge-based filtering and Usage-based systems.

Web Usage Mining (WUM) is an application of data mining to discover usage pattern from web log files and identify the underlying user functional interests that lead to common navigational activity and has become an active topic of research and commercialization.

Data Mining techniques such as association rules, classification, and clustering and attribute selection are considered very useful in web usage mining. Association rule find the correlations among the items in large data sets. Association rules generated the relationships among items in the data, because in the large number of such relationships can be established, confidence and support help analysts to filter out unwanted rules.

This paper describes how to use the Web Usage Mining in implementation of Web Recommendation system. This paper also describes how to conduct the usage mining using DHP algorithm, one of the association rule mining technique. The e-library system is developed to use as a web place for giving recommendation.

The rest of the paper is organized as follows. The next Section summarized the related work. Some knowledge concerning with the log file is describe in Section 3, because these access log are used as source for usage mining. The detail of the system design and system implementation is described in Section 4. In this session, how to implement based on e-library domain and how to conduct rule analysis and interpretation into recommendation system. Finally this paper is concluded in Section 5.

## 2. Related Work

Several data mining techniques can be used in web usage mining. Association rules in [4, 9] web logs are discovered interesting rule for web designers. Web usage mining technique is able to capture useful knowledge about user task pattern from usage data. According [8] proposed a Probabilistic Latent Sematic Analysis (PLSA) model to discover the navigation patterns. Using PLSA the hidden sematic relationships among users and web pages can be detected. Data preprocessing of web usage mining, there are several preprocessing tasks to the ready data for data mining algorithms to the data collected from server logs[2].Web access patterns can provide valuable information for web recommendation in making website-based

communication more efficient. To extract interesting or useful web access patterns, web data mining techniques which analyze historical web access log to mine the most interesting web access associations [3]. Association rules in data mining techniques are used to discover the pages which are visited together even if they are not directly connected, which can reveal associations between groups of users with specific interest [5]. Author [11] applied data mining techniques to discover the correlation between visited pages and generate the frequent patterns and analyze the association rules to produce strong associated travel path patterns from web access log.

## 3. Web Log Data for Usage Mining

According to [10], there are different levels of collections to collect the usage data. They are (i) Server Level Collection, (ii) Client Level Collection and (iii) Proxy Level Collection.

In Server Level collection, web server store information of each page requested by web visitor on a file called the web access log. The data recorded in Server Logs reflects the access of a web site by multiple users. Web Server log are plain text (ASCII) files, that is independent from the server platform. There are some distinctions between server software. But traditionally there are four types of server logs (1)Transfer log(2)Agent log (3)Error log(4)Referrer log. The first two types of log file are standard [12].

These log files can be stored in various formats. Such as combined log and extended log format. Combine log format is a widely used format on Apache-Servers. A typical line in the logs looks as follows;

*192.162.218.155 – [27/Jun/2009: 00:01: 54+ 0200] " GET/ebook/software.htm HTTP/1.1" 200 38890 http://www.library.com "Mozilla/4.0 (compatible ; MSIE 6.0, Window 98)".*

The first part of the example log format, 192.162.218.155, specifies the *IP-address* of the client who made the request to the server. The second component indicates the time when the server completed the request. The third part of the line" GET/ebook/software.htm HTTP/1.1" is called the *request line* and consists of three parts.

GET-part specifies the *method* used to request the page. The second part of the request line indicates which file was requested. The final part of the request line specifies the *protocol* used to request the file.

This protocol is normally HTTP/1.0 or HTTP/1.1.The two numbers following the request line, 200 and 38890, are respectively a status code and the size of the returned file. The status code 200 indicates that the request was successfully completed. Other status codes indicate various types of errors, from which "error 404: Page not found", is probably the best known. The next part of the line designates the referrer. This is the page that refers to the requested page. Finally, the last component of the line specifies the browser.

## 4. System Design and Implementation

This session describes the detail of system design phase and detail of system implementation phase.

### 4.1 System Design

The design of the system is mainly based on web usage mining process; it mines the frequent patterns in web access log by using DHP algorithm. There are three main components in system design as shown in Figure1.

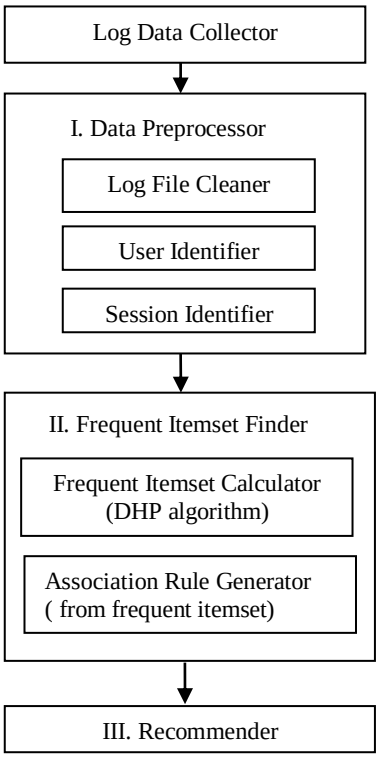


**Figure 1 System Design**

(i)Data Preprocessor

This component is responsible to load the raw log file and preprocess the log file. This phase contains sub processes of data cleaning, user identification and session identification.

(ii)Frequent Itemsets Finder

The log file getting from Step 1 is inputted to this phase to find the frequent itemsets. Here DHP algorithm is used to find the frequent itemsets. This component takes the advantage of DHP algorithm for

efficient generating of candidate set for large 2-itemsets and reducing the database scanning time.

This component also responsible to generate rules based on the user needs. Therefore the threshold values (minimum support and minimum confidence) must be inputted by users.

(iii)Recommender

The rules getting from second component is applied to provide as correlated and frequent access patterns to users.

## 4.2 System Implementation

To evaluate the effectiveness of system design, a web recommendation system is implemented in the application area of e-library. Firstly, e-library system is developed with various kinds of ebooks. When registered users make access to the ebooks, the server log stores the user access pattern with the Combine Log Format on Apache Server for usage mining. A sample raw web log data is illustrated in Figure 2.

195.162.218.155 - [27/Jun/2009:07:44:30+0200] "GET/ebooks/software.htm HTTP/1.1" 200 11223 " http://www.library.com " "Mozilla/4.0(compatible;MSIE6.0; window+NT+5.1;+hotbar+4.3.5.0)"

198.168.218.166 - [27/Jun/2009:00:01:57+0200] "GET/avatar.jpg HTTP/1.1" 200 334590 " http://www.library.com " "Mozilla/4.0(compatible;MSIE6.0;window 98)"

198.168.218.166 - [27/Jun/2009:00:01:57+0200] "GET/pictu.jpg HTTP/1.1" 200 322590 " http://www.library.com " "Mozilla/4.0(compatible;MSIE6.0;window 98)"

198.168.218.166 - [27/Jun/2009:00:01:57+0200] "GET/rurr.jpg HTTP/1.1" 200 31180 " http://www.library.com " "Mozilla/4.0(compatible;MSIE6.0;window 98)"

198.168.218.166 - [27/Jun/2009:00:01:57+0200] "GET/avatar.jpg HTTP/1.1" 200 334590 " http://www.library.com " "Mozilla/4.0(compatible;MSIE6.0;window 98)"

**Figure 2:Sample Raw Web Log Data**

### 4.2.1 Preprocessing

Before applying data mining algorithm, data preprocessing must be performed to convert the raw data into data abstraction necessary for the further processing. There are three steps for preprocessing in this system: they are (i) Data Cleaning,(ii) User Identification, and (iii) Session Identification.

**(i) Data Cleaning**

Many web log entries that are considered uninteresting for mining were removed in this cleaning step. Unnecessary portion of web log contents are removed by comparing with 10 arrays lists (.ico, .jpg, .jpeg, .gif, .js, .css, .png, .jpe, .jare) as shown in Table 1. For example, if the irrelevance entry of ".jpg" is found, then ".jpg" portion is removed from log file.

**Table 1 : Irrelevant Entries**

| No | Irrelevance entries | No | Irrelevance entries |
|---|---|---|---|
| 1 | .ico | 6 | .css |
| 2 | .jpg | 7 | .png |
| 3 | .jpeg | 8 | .jpe |
| 4 | .gif | 9 | .jar |
| 5 | .js | 10 | .jare |

After cleaning the irrelevant entries, several attributes are ignored because they were considered not necessary for the analysis. For example *www.library.com* is uninteresting attribute in the raw web log file. And then the clean logs records will be stored in the database. The data design of the clean web log data is illustrated in Table 2.

**Table 2: Clean Web Log Data**

| IP Address | Date Time | Method | URL address | Protocol |
|---|---|---|---|---|
| 195.162.218.155 | 27/jun/2009 :00:01:45 | GET | ebooks/ software.htm | HTTP/1.1 |
| 196.168.218.166 | 27/jun/2009 :00:01:55 | GET | ebooks/ ASP.htm | HTTP/1.1 |
| 195.162.218.155 | 27/jun/2009 :00:01:56 | GET | ebooks/ Asp.htm | HTTP/1.1 |

**(ii) User Identification**

This user identification step applies heuristics method to identify unique users. The content from IP attribute from cleaned web log file is used to identify the user. In this step, we first compared the IP address of each user request from Table 3 and separated them into same IP group of users.

**Table 3: URL Stem and IP from Web Log**

| URI stem | IP |
|---|---|
| ebooks/software.htm | 195.162.218.155 |
| ebooks/ASP.htm | 196.168.218.166 |
| ebooks/ASP.htm | 195.162.218.155 |
| ebooks/Java2.0.htm | 195.162.218.155 |
| ebooks/ASPWith C#.htm | 198.168.218.166 |
| ebooks/JavaGuide.htm | 196.168.218.166 |
| ebooks/software.htm | 196.168.218.166 |
| ebooks/ASPADOGuide.htm | 198.168.218.166 |
| ebooks/ASP.htm | 198.168.218.166 |
| ebooks./JavaGuide.htm | 198.168.218.166 |
| ebooks/software.htm | 195.166.216.145 |

After separating process, four user groups (User1, User2, User3 and User4) are obtained as shown in Table 4.

**Table 4: User Identification Based on IP Address**

| User1 | |
|---|---|
| **URL** | **IP** |
| ebooks/software.htm | 195.162.218.155 |
| ebooks/ASP.htm | 195.162.218.155 |
| ebooks/Java2.0.htm | 195.162.218.155 |

| User2 | |
|---|---|
| **URL** | **IP** |
| ebooks/ASP.htm | 196.168.218.166 |
| ebooks/JavaGuide.htm | 196.168.218.166 |
| ebooks/software.htm | 196.168.218.166 |

| User3 | |
|---|---|
| **URL** | **IP** |
| ebooks/ASPWith C#.htm | 198.168.218.166 |
| ebooks/ASPADOGuide.htm | 198.168.218.166 |
| ebooks/ASP.htm | 198.168.218.166 |
| ebooks./JavaGuide.htm | 198.168.218.166 |

| User 4 | |
|---|---|
| **URL** | **IP** |
| ebooks/software.htm | 195.166.216.145 |

**(iii) Session Identification**

This step identifies the user sessions. The goal of session identification is to divide the page accesses of each user into individual sessions. The simplest method of achieving this is through a timeout, where if the time between page requests exceeds a certain limit, it is assumed that the user is starting a new session.

This system assumes standard time is 30 for each session. Although Table 5 contains the same IP [195.162.218.155], user sessions may be differ base on time. Therefore the three contents (users) from Table 5 is split into another two Table 6 and Table 7 based on time. Because the access time of record 1 and 2 are within 30 minutes, so these two records are assumed as one session. The remaining record exceed 30 minutes .So, this is assumes other session.

**Table 5: Session Identification**

| URL stem | IP | Time |
|---|---|---|
| ebooks/software.htm | 195.162.218.155 | 08:22:30 |
| ebooks/ASP.htm | 195.162.218.155 | 08:26:45 |
| ebooks/Java2.0.htm | 195.162.218.155 | 09:55:12 |

**Table 6: Identified Session (1)**

| URI stem | IP | Time |
|---|---|---|
| ebooks/software.htm | 195.162.218.155 | 08:22:30 |
| ebooks/ASP.htm | 195.162.218.155 | 08:26:45 |

**Table 7: Identified Session (2)**

| URI stem | IP | Time |
|---|---|---|
| ebooks/Java2.0.htm | 195.162.218.155 | 09:55:12 |

After preprocessing the user identification and session identification, the preprocessed log data is still remaining long format (for example, the URL is “ebook/ASP.htm”).This long format is needed to replace with symbol/ ID number for more convenience to calculate. So we identify the right most part of URL content as book IDs. Table 8 is illustrated the mapping between URL and BookID.

**Table 8 : Mapping between URL and BookID**

| **URL** | **BookID** |
|---|---|
| Softwar.htm | 1 |
| Java2.0.htm | 2 |
| ASPWithC#.htm | 3 |
| JavaGuide.htm | 4 |
| ASP.htm | 5 |

By using the result of User Identification and Session Identification, the following transactions table is obtained. Table 9 is very useful to proceed the next phase of pattern mining.

**Table 9: Transaction Table Design**

| **Transaction IDs** | **List of book IDs** |
|---|---|
| 1 | 1,2,5 |
| 2 | 2,4 |
| 3 | 2,3 |
| 4 | 1,2,4 |
| 5 | 1,3 |
| 6 | 2,3 |
| 7 | 1,3 |
| 8 | 1,2,3,5 |
| 9 | 1,2,5 |

**4.2.2. Pattern Mining Using DHP**

In this step frequent access patterns are generated by DHP algorithm from the output of preprocessing step. The outputs of preprocessing step are transformed into transaction database. The first step of DHP algorithm counts each URL containing in the transaction database. The system scans the database for the first time to generate C1 and L1.From L1, 2-patterns are generated. Then hash table H2 is created using the hash algorithm. The step of algorithm illustrate in Figure 3. In the following example, total type of eBook pages containing is 5 and total users 9.

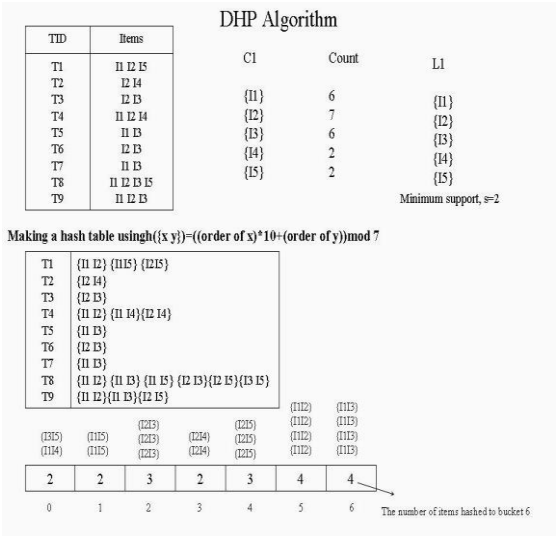


**Figure 3: Generate Candidate C1 and Creating H2**

C2 and L2 are generated from the H2. This process is shown in Figure 4. The same process is performed until no more reduced database can be created.

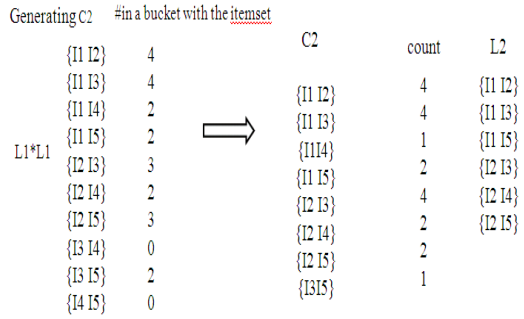


**Figure 4: Generate C2 and L2**

After getting the C2, the system is needed to calculate L2. In this example, six L2 itemsets possess predefined minimum support and two remaining C2 itemsets does not posses predefined minimum support. Therefore only 6 itemsets contains in L2.Then confidence is calculated based on L2 itemsets.

I1^I2 confidence =66% → Output for recommendation
I1^I3 confidence =66% → Output for recommendation
I1^I5 confidence =33%
I2^I3 confidence =57%
I2^I4 confidence =29%
I2^I5 confidence =29%

For example, if the minimum confidence threshold is 60% then rules produce as output .Except of lines first and second are output of this example.

#### 4.2.3. Rule Generation for Recommendation

In this system, rules are generated based on the user input of minimum support and minimum confidence.

If the rule is

I1 ^ I2 confidence 66%

Then the meaning for recommendation is

"You should visit/read I2 after I1".

## 5. Conclusion

This system mined frequent patterns from web log data and extracted association patterns by using DHP algorithm. By taking the advantages of DHP algorithm, this recommendation system performs effectively on large data sets and searches the correlated book links. The results of correlated book links are used to provide the users as related book information. By using this system, librarians can know association between books that are interested by most visited users, and users will get the recommendation which book should be read. This system can be extended as a more comprehensive recommendation system by adding other facilities such as rating-based recommendation.

## References

[1] R. Agrawal, T. Imielinski, and A. Swami."Mining association rules between sets of items in large databases". In Proc. of the ACM SIGMOD Conference on Management of Data, Washington, D.C., May 1993, 207-216.

[2] R.Cooley , B.Mobasher, and J.Stivastava, " Data perprocessing for mining world wide web browsing patterns", Department of Computer Science and Engineering University of Minnesota 4-192 EECS .

[3] L.S.L.Cheng, and J.Ford, "Mining the most interesting web access log associations", the Dartmouth Experimental Visualization Laboratory (DEVLAB), Department of Computer Science, Dartmouth College, Hanover NH 03755.

[4] M.S.Chen, J.S. Park, and P.S.Yu, " Data mining for path traversal pattern in a web environment".

[5] M. Eirinaki and M. Vazirgiannis, "*Web mining for web personalization*," *ACM* Trans. *Inter. Tech.*, vol. 3, no. 1, 2003, pp. 1-27.

[6] S.Giindiiz , M.T.Ozsu ,"Recommendation Models for User Accesses to Web Pages"(invited paper),Department of Computer Science, Istanbul Technical University Istanbul, Turkey.

[7] J.Huysmans, B.Baesens, J.vanthienen, Web mining for web personalization", ACM Trans.Inter.Tech, vol.3, no.1, 2003, pp.1-27.

[8] X.Jin, Y.Zhou,and B.Mobasher. "Web usage mining based on probabilistic latent sematic analysis," in KDD'04: Proceeding of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining. Patterns in Web Access log.

[9] J. Punin, M. Krishnamoorthy, and M. Zaki, "*Web usage mining: Languages and algorithms*," in Studies in Classification, Data Analysis, and Knowledge Organization. Springer-Verlag, 2001

[10] J.Srivasta, R.Cooley, M.Deshpande, P.Tan "Web Usage Mining: Discovery and Application of Usage Patterns from Web Data", ACM SIGKDD, vol.1,Jan 2000, Issue 2-page 12 .

[11] C.M.Win, K.A.Than, "Mining Frequent Patterns in Web Access log".

[12] M.Wahab, M.Mohd, H.Hanafi, M.Farhan, M.Mohsin "Data Preprocessing on Web Server Logs for Generalized Association Rules Mining Algorithm", proceedings of world academy of science, engineering and technology, vol.36, 2008, ISSN 2070-3740.