International Journal of Information Science and Management

Integrating Web Content Mining into Web Usage Mining for Finding Patterns and Predicting Users' Behaviors

S. Taherizadeh
Group of Information Technology Engineering
Tarbiat Modarres University
Tehran, I. R. of Iran
Corresponding Author: Taherizadeh@modares.ac.ir

N. Moghadam
Department of Computer Science
Tarbiat Modarres University
Tehran, I. R. of Iran
email: charkari@modares.ac.ir

Abstract

With the increased confidence in the use of the Internet and the World Wide Web, the number of electronic commerce (e-commerce) transactions is growing rapidly. Therefore, finding useful patterns and rules of users' behaviors has become the critical issue for e-commerce and can be used to tailor e-commerce services in order to successfully meet the customers' needs. This paper proposes an approach to integrate Web content mining into Web usage mining. The textual content of web pages is captured through extraction of frequent word sequences, which are combined with Web server log files to discover useful information and association rules about users' behaviors. The results of this approach can be used to facilitate better recommendation, Web personalization, Web construction, Website organization, and Web user profiling.

Keywords: Association Rule, Clustering, Frequent Word Sequence, Web Content Mining, Web Usage Mining.

Introduction

Electronic commerce (e-commerce) is the use of computers and telecommunication technologies to share business information, maintain business relationships, and conduct business transactions. Currently, e-commerce depends mostly on the Internet as the basic platform. With the rapid growth of Websites and increase in e-commerce systems, maintaining Web log data has become very important for many of services. Thousands of people may visit a Website in a given period of time and these visits are stored as Web log data. Web log data provides important information about users' behaviors.

Data mining techniques are most used to find useful information from Web documents or Web services (Etzioni, 1996). A successful utilization of the Web data requires the exploit of data mining technologies, giving a rise to the area of Web mining. Realistically, Web mining is the application of data mining technologies in the Web environment and can help find useful patterns and rules of users' behaviors. Since

data mining technologies are being applied for a variety of analytical purposes in Web environment, Web mining could be further categorized into three major sub-areas: Web content mining, Web structure mining, and Web usage mining (Madria, Bhowmick, Ng, and Lim, 1999; Borges, and Levene, 1999). Web content mining attempts to discover useful information from Web contents. For example, the classification of web pages is a typical application of content mining techniques (Shen, Cong, Sun, and Lu, 2003). Web structure mining studies the Web linkage structure. Finally, Web usage mining focuses on the Web surfer's sessions and behaviors.

In this work, the main hypothesis is that Web page contents can be used to increase in quality of Web usage mining results. Therefore, the purpose of this paper is to propose a system to find useful association rules by integrating Web document analysis into Web usage mining. These association rules can be used to tailor e-commerce services in order to successfully meet the customers' needs and indicate the patterns and rules of users' behaviors that can be utilized to facilitate better analysis of user access behaviors, recommendation, Web personalization, Web construction, Website organization, and Web user profiling.

Overview

Cluster Analysis

Clustering analysis is widely used to establish object profiles based upon objects' variables. Objects can be customers, web documents, web users, or facilities (Chang, Hung, and Ho, 2007). Unlike classification, which analyzes class-labeled data objects, clustering analyzes data objects without consulting a known class label. The objects are clustered based on the principle of maximizing the intra-class similarity and minimizing the inter-class similarity. This means that clusters of objects are created for the objects within a cluster have high similarity in comparison to one another, but are very dissimilar to the objects of other clusters (Han and Kamber, 2006).

There exist many clustering algorithms, which can be classified into several categories, including partitioning methods, hierarchical methods, density-based methods, grid-based methods, and model-based methods (Han and Kamber, 2001). A partitioning method classifies objects into several one-level clusters, where each object belongs to exactly one cluster, and each cluster has at least one object. A hierarchical method creates hierarchical decomposition of objects. Based on how the hierarchy is formed, hierarchical methods can be classified into agglomerative (bottom-up) approaches and divisive (top-down) approaches. A density-based method is used to discover clusters with arbitrary shape, based on the number of objects in neighborhood – density. It typically regards clusters as dense regions of objects in the data space that are separated by regions of low density.

Web Usage Mining

Web usage mining attempts to discover knowledge for the data generated by the Web surfer's sessions or behaviors (Cooley, Mobasher, and Srivastava, 1997). Web site servers generate a large volume of data from user accesses. This data may help to determine life time value of users, to improve Web site structure design, to evaluate the efficiency of Web services (Cooley et al, 1997), etc. Web usage data includes Web server access logs, proxy server logs, cookies, user profiles, registration data, user sessions, user queries, bookmark data, mouse clicks and scrolls and any other data as the results of interactions. The logfile analysis technologies include association rule mining (Lin, Alvarez, and Ruiz, 2000), sequential pattern mining (Zhou, Hui, and Chang, 2004), clustering and classification. The task of sequential pattern discovery is to find inter-transaction patterns, i.e. the presence of a set of items followed by another item, in the time-stamp ordered transaction set. Clustering and classification on Web server log file is a process that group the users, Web pages, or user requests on the basis of the access request similarities. Association rule mining task is to discover the correlation among a variety of issues like accesses to files, time of accesses, and identities who requested the accesses.

Association Rule Mining

Mining association rules among items in a large database is one of the most important problems of data mining (Han and Kamber, 2001). The task is to find interesting association or correlation relationships among a large set of data items. The typical rule mined from database is formatted as follows:

$$X \Rightarrow Y\ [Support, Confidence]$$

It means the presence of item X leads to the presence of item Y, with [*Support*]% occurrence of [X,Y] in the whole database, and [*Confidence*]% occurrence of [Y] in set of records where [X] occurred.

Web Content Mining

Web content mining focuses on the discovery of useful information from the web contents (Kosala and Blockeel, 2000). The Web contents consist of a large variety of different types of data. The two most important characteristics of WWW data are large volume and heterogeneity. Dealing with problems in these two aspects, Web content mining can be categorized into IR view (agent-based approach) and DB view (database approach).

In the DB view, Web content mining task is to integrate and organize the heterogeneous and semi-structured Web data into well-organized and high-level collections of resources, such as relational databases, to provide more sophisticated

queries instead keyword-based searches. Applications include finding DataGuide, schema, type hierarchy, useful/frequent sub-structure, as well as setting up multilevel databases (Kosala and Blockeel, 2000).

In the IR view, Web content mining tries to retrieve relevant information from a drawing source of Web data. It intends to develop more intelligent Information Retrieval systems by assisting and improving information finding and filtering. The applications include categorization, clustering, finding extracting rules, finding patterns in text, as well as user modeling. Text document is the most commonly used 'object' in Web content mining in the IR view. N-grams, bag of words, terms, phrases, concepts and ontologies are usually used to represent these documents. Document clustering is an important part of the Web content mining.

Document Clustering

Document clustering is the organization of a set of text documents into clusters on the basis of similarity. Indeed, text documents contained by a cluster are more similar to each other than those owned by a different cluster. 'Vector-space' model is widely used in document clustering. In the Vector-space model, each document is represented by a vector of the frequencies of 'features' (words, terms, or N-grams). But, in the Vector-space model, the position of the words in the text documents is not considered. Since the order of words in a text document is significant, the Vector-space model was not used in this study. Hence for overcoming to this problem, a new model to represent the document is proposed here. The sequential relationship between words in the document is preserved in the proposed algorithm and utilized for the document clustering. This clustering algorithm is the model based on the algorithm called 'CFWS' (Document Clustering Based on Frequent Word Sequences) proposed by Li, Chung, and Holt (2007).

System Architecture

We have proposed an experimental system to find useful association rules by integrating Web document analysis into Web usage mining. For our experiments, it is necessary to use a case that paves the way to analyze both its web log data and web pages. Our experiments have been conducted on an IIS server log access file from the Web server of a software provider company. In this case, the user name of visitors is anonymous. The data flow of the system is shown in Figure 1. Additionally, each component of the system is described with all the specifics in the following subsections.

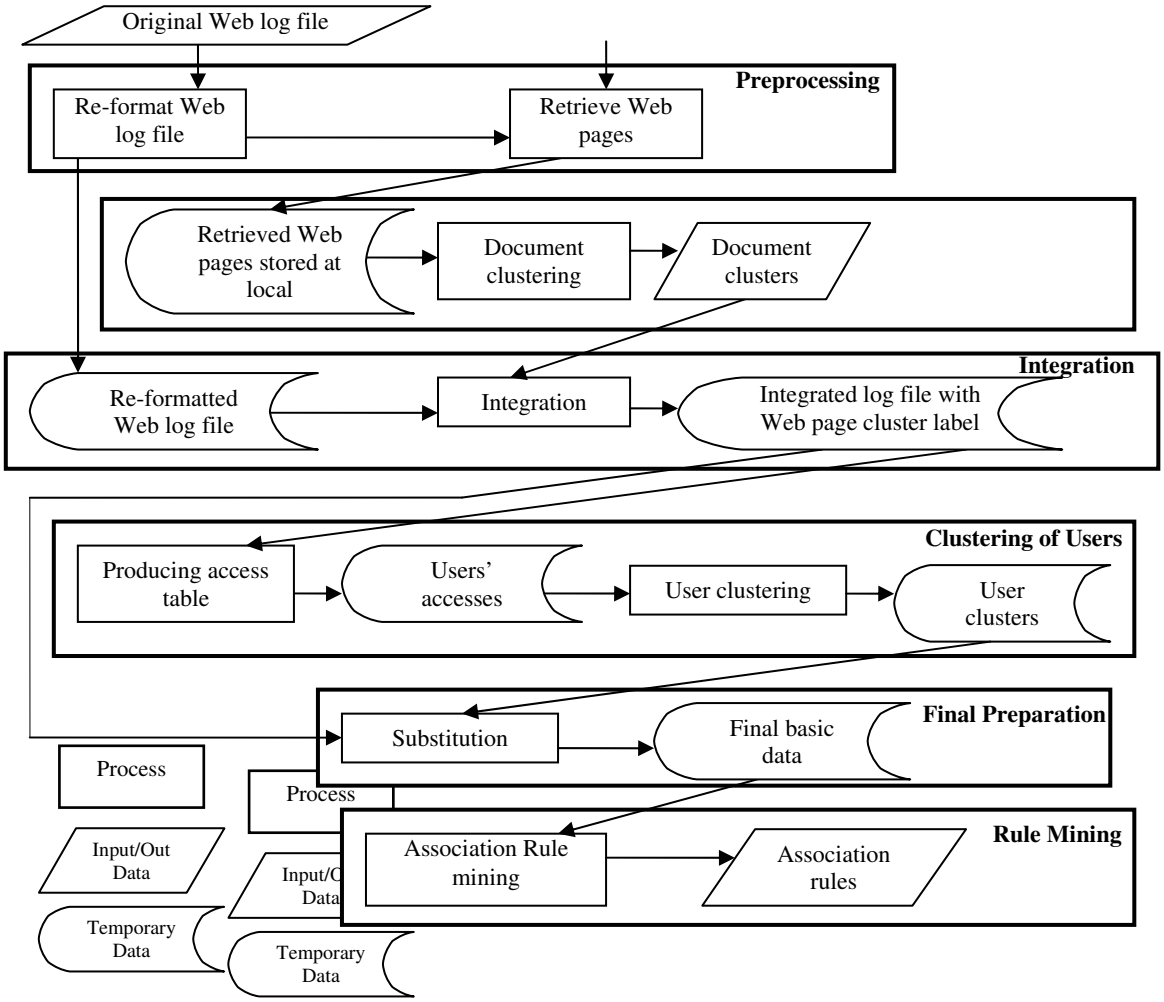


Figure 1. System dataflow diagram.

Preprocessing

In the preprocessing step, there are two major tasks: re-format the log file and retrieve Web pages to local space. Log file re-formatting involves revising the log file to an appropriate format for further steps. Web page retrieving involves reading the URL address part of the re-formatted log file, retrieving the Web pages accordingly, and storing them to local space.

Re-formatting web log file

Each tuple of the log file is a record for one request from a certain user. For example, Figure 2 shows an IIS log file entry, as viewed in a text editor and Table 1 lists and describes the fields used in this work.

```
Y.Y.Y.Y, -, 03/20/07, 7:55:20, W3SVC2, SERVER, X.X.X.X, 4502, 163, 3223, 200, 0, GET, /Logo.gif, -,
```

Figure 2. An example of IIS log file entry.

Table 1

Description of the Used Fields Recorded in the IIS Log File Entry

Field	Appears as	Description
Client IP ddress	Y.Y.Y.Y	The IP address of the client.
Date	03/20/07	This log file entry was made on March 20, 2007.
Time	7:55:20	This log file entry was recorded at 7:55 A.M.
Target of operation	/Logo.gif	The user wanted to download the Logo.gif file.

The fields used for this study are client IP address, date, time and target of operation:

- *Client IP address:* indicating the IP address from which the user is accessing Web server, in format of four three-digit number eliminated by dot.
- *Date:* indicating the time of request.
- *Time:* indicating the time of request.
- *Target of operation:* indicating the target of operation such as a page and an image.

First, we build two list structures: suffix list and called robot list. The suffix list consists of the suffix of image files. The robot list includes robots' IP addresses. Each entry of web log file is checked whether the suffix of its URL matches any values of the suffix list or its IP address matches any values in the robot list. The matched entries of web log file are removed from the results.

The original format of log file needs to be modified according to the following reasons. Generally, there are too many distinct values for the fields such as IP address, day and time. For IP address, the values starting with same prefix can be treated the same user or a similar group of users. For example, many users utilize some commercial internet services that the IP addresses for these may also be dynamic. For the date part, we used the days of a week. That is, all Sundays are same, despite the specific date, so on and so forth. For the time part, we also converted them into sections of day (morning, afternoon, evening and night) according to the hours, in order to reduce the number of different values. An example of re-formatting Web log file for one entry is shown in Table 2.

Table 2

Web Log File Re-Formatting

Field	Original format	Re-formatted
IP address	213.176.78.10	213.176
Date	03/20/07	Tuesday
Time	7:55:20	Morning
URL	/default.htm	/default.htm

Retrieving web pages

This subsection involves reading the URL address part of the re-formatted Web log file, retrieving the Web pages accordingly, and storing them to local space. Only the page types that can be treated as text documents are kept and other types should be filtered. In this work, the following types of files are retrieved as resource of document clustering: .htm, .html, .shtml, .xml, .php, .cgi, .txt, .pdf, .doc.

Clustering of Documents

As mentioned in overview section, we used a document clustering algorithm based on the CFWS algorithm proposed by Li et al. (2007). An ordered sequence of at least two words is called a 'word sequence', such as $<w_1,w_2...,w_n>$, in which w_k is not certainly following w_{k-1} immediately ($k=1,2,...,n$). A document d supports a word sequence S, if there is at least the specified minimum number of occurrences of S in d. A word sequence S is a 'frequent word sequence' when there are at least the specified minimum number of documents supporting S. A 'frequent k-word sequence' is a frequent word sequence with length k, such as $<w_1,w_2...,w_k>$, and it has two frequent subsequences of length $k-1$, which are $<w_1,w_2...,w_{k-1}>$ and $<w_2,w_3...,w_k>$. Obviously in our algorithm, unlike the CFWS algorithm proposed by Li et al. (2007), multiple occurrences of a sequence in the same document is not counted as one because one document may refer to the specified sequence only a few times, whereas the topic is not relevant to the document.

Preprocessing of documents

In our algorithm, unlike the CFWS algorithm proposed by Li et al. (2007), before finding frequent 2-word sets, there are several preprocessing steps to take:

- Cleaning HTML, XML or SGML tags from the Web pages,
- Eliminating all punctuations like comma, full stop, quotation mark, etc., only except the underscore in-between words,
- Eliminating all digital numbers,
- Changing all characters to lower case and
- Eliminating the stops words, which are very common words such as 'an', 'and', 'from', 'is' and so on.

Example database: $D = \{d_1, d_2, d_3\}$

- d_1: Some amateur students study books!
- d_2: "Amateur students want to study books."
- d_3: <P>However, many students study same books</P>

Preprocessed documents:

- d_1': amateur students study articles
- d_2': amateur students want study books
- d_3': students study books

Finding members of all the frequent word sequences

Indeed, in this subsection, we try to find the members of all the frequent word sequences. We first find all the frequent 2-word sets. Then, all the words in these sets are put into a set called 'MW'. Since, on the basis of definition of frequent k-word sequence, each member of all the frequent word sequences exists in MW, MW contains all the members of the frequent word sequences.

"Specified minimum number of occurrences of S in one document supporting S" = 1
"Specified minimum number of documents supporting S" = 2
The set of frequent 2-word sets is {{students, study}, {amateur, students}, {students, books}, {amateur, study}, {study, books}}
MW = {amateur, students, study, books}

Eliminating all the words in the documents that are not in MW

We eliminate all the words in the documents that are not in WS. The purpose is to decrease the dimension of the database. The resulting documents are called 'compressed documents'.

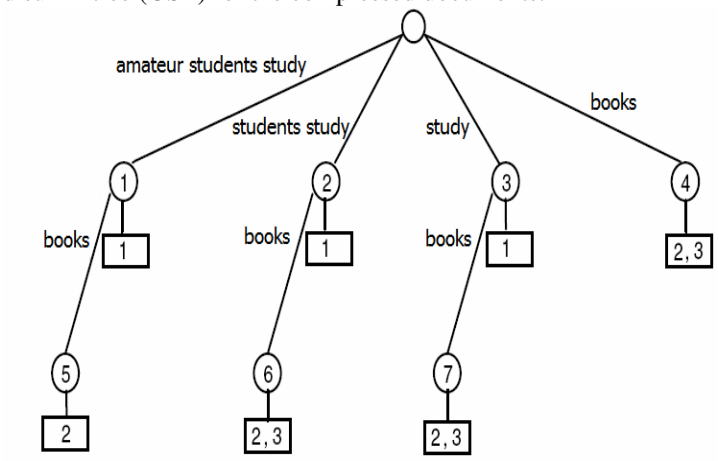
Compressed documents:

- d_1": amateur students study
- d_2": amateur students study books
- d_3": students study books

Creating a generalized suffix tree (GST)

In this subsection, the suffix tree (Weiner, 1973) is used to facilitate finding the all frequent word sequences. The suffix tree (ST) is undoubtedly one of the most important data structures in string processing (Abouelhoda, Kurtz, and Ohlebusch, 2004). This is particularly true if the sequences to be analyzed are very large and do not change such as static web pages. The ST of string $x[1..n]$ is the compacted tree of all non-empty suffixes $x[i..n]$ for $i = 1,..,n$. A GST is an ST that combines the suffixes of a set $\{S_1,\ldots,S_n\}$ of strings. We put all compressed documents into a GST. At the final GST, each suffix node has a box that contains the document id set of the suffix node.

Generalized suffix tree (GST) for the compressed documents:



Finding frequent word sequences of all length

We carry out the depth-first traverse upon the GST. Using the data put into the nodes of this GST, the information about all the frequent word sequences of the database can be obtained. After traversing the GST, via examination the support count and the length of the label of each node, all the frequent word sequences of the database can be found.

"Specified minimum length of the word sequence" = 2

Node No.	Word Sequence	Length of Word Sequence	Document Ids	Number of Document Ids	Frequent Word Sequence
1	amateur students study	3	1,2	2	Yes
2	students study	2	1,2,3	3	Yes
3	study	**1**	1,2,3	3	No
4	books	**1**	2,3	2	No
5	amateur students study books	4	2	**1**	No
6	students study books	3	2,3	2	Yes
7	study books	2	2,3	2	Yes

Producing cluster candidates

Each suffix node representing a frequent word sequence produces one cluster candidate that contains a set of documents supporting the same frequent word sequence (FWS).

Four cluster candidates can be collected:

Cluster Candidate No.	Frequent Word Sequence (**FWS**)	Document Id Set
1	amateur students study	{1,2}
2	students study	{1,2,3}
3	students study books	{2,3}
4	study books	{2,3}

Merging the cluster candidates based on the k-mismatch concept

The mismatches between the frequent word sequences are examined. Three types of mismatches exist:

• **Insertion:** through adding k words into the shorter FWS, it becomes the longer FWS.

• **Deletion:** through removing k words from the longer FWS, it becomes the shorter FWS.

• **Substitution:** through replacing k words in first FWS (that supported by first cluster candidate), it becomes second FWS (that supported by second cluster candidate).

The Landau-Vishkin (LV) algorithm (Landau and Vishkin, 1989) is used to find all the *k*-mismatched frequent word sequences for the longest FWS for a given *k*. Then, the cluster candidate with the longest FWS is merged with all cluster candidates that have *k*-mismatch FWS of the longest FWS, into a new cluster. Subsequently, all the merged cluster candidates are removed from the set of all initial cluster candidates. These two steps are repeated until there are not any clusters in the set of all initial cluster candidates to merge.

k =1
Insertion: {FWS_i = "students have"; FWS_j = "students have books"}; by adding a word "books" in FWS_i, $FWS_i = FWS_j$;

Deletion: {FWS_i = "students have books "; FWS_j = "have books "}; by removing a word "students" from FWS_i, $FWS_i = FWS_j$;

Substitution: {FWS_i = "students have"; FWS_j = "teachers have"}; by replacing a word "students" of FWS_i with a word "teachers", $FWS_i = FWS_i$;

Producing overlap matrix (O)

After merging, some resulted clusters may have overlap between their document id sets considerably. The overlap of two clusters, $Cluster_i$ and $Cluster_j$, is estimated as follows:

$$O_{ij} = \frac{n(\text{Document Id Set}_i \cap \text{Document Id Set}_j)}{n(\text{Document Id Set}_i \cup \text{Document Id Set}_j)}$$

Combining the overlapping clusters

If the user specifies the number of final clusters, two clusters with the highest overlap value (O_{ij}) are combined into one cluster repetitively until the number of clusters becomes the specified number. On the other hand, if the number of final clusters is not specified, two clusters are merged only when their overlap value is larger than the specified overlap threshold value δ.

Creating final cluster

At the finish, those documents that are not in any cluster are gathered into one cluster.

Integration

The integration step is to integrate the Web document cluster information into log file. The result of this step is a table. For example, a part of this table is shown in Table 3.

Table 3

Integrated Log File with Web Page Cluster Label

IP address	Date	Time	URL	Document Cluster
213.208	Tuesday	Morning	/default.htm	DCluster1
213.208	Tuesday	Morning	/Persian/news/index.htm	DCluster9
213.208	Tuesday	Morning	/Persian/news/indexup.htm	DCluster9
140.193	Tuesday	Morning	/Persian/index.htm	DCluster3
140.193	Tuesday	Morning	/Persian/yesobkm.htm	DCluster10
213.176	Tuesday	Morning	/default.htm	DCluster1

User Clustering

This work bases user clustering on access to document clusters. Therefore, we, first, produce access table and then, use K-means algorithm for user clustering.

Producing access table

We can obtain the access matrix by using the result of previous step. For each IP address, there exists one row that shows the number of its corresponding accesses to each document cluster. For example, a part of this table is shown in Table 4.

Table 4

Part of Access Table

IP	DCluster1	DCluster2	DCluster3	DCluster4	…	DCluster18	DCluster19	DCluster20
213.208	3	0	2	0	…	0	0	0
140.193	1	0	1	0	…	0	0	1
213.176	1	0	0	0	…	0	0	0

For instance, the total number of accesses of the user with IP '213.208' across every page in the document cluster 1 is 3.

User clustering based on access to document clusters

K-means is a popular algorithm to solve the problem of clustering the data set into *k* clusters. The steps of k-means are as follows:

1. Partition objects into *k* nonempty subsets
2. Compute seed points as the centroids of the clusters of the current partition (the centroid is the center, i.e., mean point, of the cluster)
3. Assign each object to the cluster with the nearest seed point
4. Go back to Step 2, stop when no more new assignment

The number of clusters *k* is necessary to be specified in advance as the input of the K-means algorithm. In order to identify the optimal number of clusters, K-means algorithm was used for user clustering with different values of *k*. Then, the evaluation function was used to compare the results of user clustering. Realistically, for each k value, we evaluated the quality of the clustering result using *internal quality* measure which is always the quantitative evaluation and does not rely on any external knowledge. In other words, an *internal quality* measure compares different sets of clusters without references to external knowledge. We applied an evaluation function proposed by Jo (2001):

$$EI = \frac{1}{mN}\sum_{j=1}^{N} E(t_j)\sum_{i=1}^{m} E_i(t_j)$$

where t_j refers to document cluster j, m is the total number of user clusters, N is the total number of users, $E(t_j)$ is *inter-cluster* entropy, and $E_i(t_j)$ is *intra-cluster* entropy:

$$E(t_j) = 1 - \frac{m_j - 1}{m - 1}\log_m \sum_{i=1}^{m} E_i(t_j)$$

$$E_i(t_j) = \frac{n_{ij} - 1}{n_i - 1} \log_{(f_{i\max}+1)}(\bar{f}_j + 1)$$

where n_i is the number of users in user cluster C_i, n_{ij} is the number of users including t_j in user cluster C_i, f_{imax} is the maximum frequency of t_j in user cluster C_i, $\bar{f}_j$ is the average frequency of t_j in user cluster C_i, and m_j is the number of clusters in which t_j appears.

Additionally, in order to aid specifying the optimal number of clusters, some *external quality* measures are also used. In fact, an external quality measure evaluates how well the clustering is working by comparing the groups produced clustering techniques to known classes. From the Figure 3, it can be seen that 14 is an appropriate value for cluster number *k*. Therefore, 14 became the optimal number of user clusters.

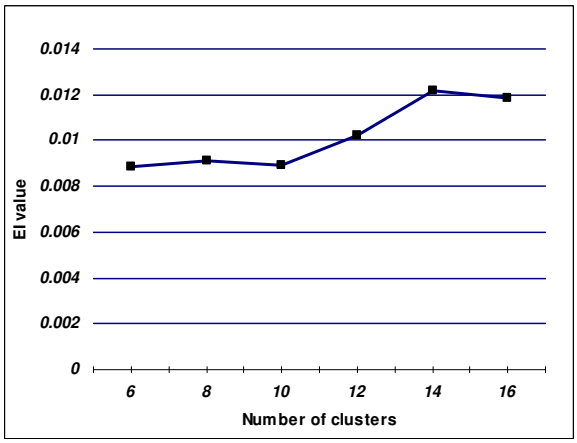


Figure 3. Comparison between results from user clustering.

Final Preparation

At this step, we substituted each IP address of the table resulted from integration step with its corresponding user cluster. The result of this step is a table that is input for association rule mining in the next step. For example, a part of this table is shown in Table 5.

Table 5

Final Basic Data

User cluster	Date	Time	Document cluster
UCluster6	Tuesday	Morning	DCluster1
UCluster6	Tuesday	Morning	DCluster9
UCluster6	Tuesday	Morning	DCluster9
UCluster4	Tuesday	Morning	DCluster3
UCluster4	Tuesday	Morning	DCluster10
UCluster11	Tuesday	Morning	DCluster1

Rule Mining

At this stage, Apriori algorithm was used. The Apriori algorithm was proposed by Agrawal and Srikant (1994) for mining association rules in large databases. Apriori is an influential algorithm for mining frequent itemsets for association rules (Han and Kamber, 2001). Two important attributes for Apriori algorithm are support threshold α and confidence threshold β. Considering the large volume of server log records, the support threshold can be set to be very low. On the other hand, the confidence threshold can not be very low, in order to get appropriate number of sufficient rules. In this work, we specified support threshold equal to $\alpha = 5\%$ and confidence threshold equal to $\beta = 50\%$.

Conclusion

We applyed Apriori algorithm on the dataset as shown in Table 5 from which a list of association rules was prepared. Table 6 displays some rules obtained from which we can get some interesting rules that are related to web page contents issues and provide information for user profiling and website evaluation and improvement. These rules also indicate when and from where the access queries occurred, who visited, and what kind of information was requested and may contribute to web personalization, web organization, web content distribution, web construction, analysis of user access behaviors and recommendation. Personalizing websites can attract new customers and retain existing customers. Web personalization technologies also benefit e-commerce applications and allow users to see and receive information based on the knowledge acquired from the users' previous actions. For example, rules such as "UCluster13 $\Rightarrow$ DCluster7" and "UCluster6 $\Rightarrow$ DCluster9" tell us about which users are interested in what kind of topics.

Table 6

Some Association Rules Obtained

{User Cluster=UCluster13} => {Document Cluster=DCluster7}
{Date=Thursday} => {Document Cluster=DCluster7}
{Time=Morning} => {User Cluster=Ucluster12}
{Time=Afternoon, Document Cluster=DCluster8} => {User Cluster=UCluster3}
{User Cluster=UCluster6} => {Document Cluster=DCluster9}
{User Cluster=UCluster10, Time=Morning} => {Document Cluster=DCluster18}
{Date=Friday, Time=Afternoon} => {Document Cluster=DCluster6}

This system will benefit web usage mining when website is not organized based on web page contents. The system is also feasible to extend to multi-server analysis, as

long as both the server log files and the web pages are accessible.

References

Abouelhoda, M. I., Kurtz, S., & Ohlebusch, E. (2004). Replacing suffix trees with enhanced suffix arrays. *Journal of Discrete Algorithms*, *Elsevier*, 53-86.

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules in large databases. *Proceedings of the 20th International Conference on Very Large Databases*. Santiago, Chile, 487-499.

Borges, J., & Levene, M. (1999). Data mining of user navigation patterns. *Proceedings of the WEBKDD'99 Workshop on Web Usage Analysis and User Profiling*, 92-111.

Chang, H. J., Hung, L. P., & Ho, C. L. (2007). An anticipation model of potential customers' purchasing behavior based on clustering analysis and association rules analysis. *Expert Systems with Applications, Elsevier*, 753–764.

Cooley, R., Mobasher, B., & Srivastava, J. (1997). Web mining: Information and pattern discovery on the world wide web. *Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97)*, 558-567.

Etzioni, O. (1996, November). The world-wide web: Quagmire or gold mine?. *Communications of the ACM*, *39*(11), 58-65.

Han, J., & Kamber, M. (2001). *Data mining: Concepts and techniques* (1st ed.). Morgan Kaufmann Publishers.

Han, J., & Kamber, M. (2006). *Data mining: Concepts and techniques*. (2nd ed.). Morgan Kaufmann Publishers.

Jo, T. C. (2001). Evaluation function of document clustering based on term entropy. *Proceedings of 2nd International Symposium on Advanced Intelligent System*, 95-100.

Kosala, R., & Blockeel, H. (2000). Web mining Research: A survey. *ACM SIGKDD*, *2*(1), 1-15.

Landau, G. M., & Vishkin, U. (1989). Fast parallel and serial approximate string matching. *Journal of Algorithms*, *10* (2), 157-169.

Li, Y., Chung, S. M., & Holt, J. D. (2007). Text document clustering based on frequent word meaning sequences. *Data & Knowledge Engineering*, doi: 10.1016/j.datak.2007.08.001.

Lin, W., Alvarez, S. A., & Ruiz, C. (2000). Collaborative recommendation via adaptive association rule mining. *WEBKDD2000 – Web Mining for E-Commerce – Challenges and Opportunities, Second International Workshop,* Boston, MA, USA.

Madria, S. K., Bhowmick, S. S., Ng, W. K., & Lim, E. P. (1999). Research issues in web data mining. *Proceedings of Data Warehousing and Knowledge Discovery, First International Conference*, DaWaK'99, 303-312.

Shen, D., Cong, Y., Sun, J. T., & Lu, Y. C. (2003). Studies on Chinese web page classification. *Proceedings of the 2003 International Conference on Machine Learning and Cybernetics*, 1, 23–27.

Weiner, P. (1973). Linear pattern matching algorithms. *Proceedings of the 14th Annual Symp. on Foundation of Computer Science*, 1–11.

Zhou, B., Hui, S. C., & Chang, K. (2004). An intelligent recommender system using sequential web access patterns. *Proceedings of the 2004 IEEE Conference on Cybernetics and Intelligent Systems*, Singapore, 1-3.