Integrating FS Tree with Clustering for Mining Frequent Web Access Patterns

Hnin Ei Ei Hlaing, May Aye Khine
University of Computer Studies, Yangon
hninei.ucsy@gmail.com

Abstract

Mining frequent patterns is an important component of many prediction systems. One common usage in web applications is the mining of users' access behavior for the purpose of predicting and hence pre-fetching the web pages that the user is likely to visit.

This paper presents web usage mining model for discovering frequent patterns in sequence databases that requires only two database scans. The first scan obtains support counts for subsequences of length. The second scan extracts potentially frequent sequences tree structure (FS-tree). Frequent sequence patterns are generated by mining the FS-tree. On the other hand, clustering methods are unsupervised methods, and normally are not used for classification directly. This paper involves incorporating clustering with FS-tree algorithm. The pre-processed data is divided into meaningful clusters then the clusters are used as training data for the FS-tree algorithm, to get higher accuracy.

1. Introduction

The ongoing increase of digital data on the Web has resulted in the overwhelming amount of research in the area of Web user browsing personalization and next page access prediction. For web applications, where users' requests are satisfied by downloading pages to their local machines, the use of mining techniques to predict access behaviors and hence help with prefetching of the most appropriate pages to the local machine cache can dramatically increase the runtime performance of those applications. These mining techniques analyze web log files composed of listings of page accesses (references) organized typically into sessions. These techniques are part of what is called Web Usage Mining.

This paper presents an algorithm for extracting frequent sequence patterns from web usage data. It is especially useful in correct predictions of next web access. Frequent sequences are also known as Traversal Patterns. There are several algorithms to generate the frequent sequences. Apriori based algorithms require multiple expensive scans of the database, one for each level, to determine which of the candidates are frequent. But FS-Tree alone does not provide good accuracy, and thus incorporating Clustering with FS-Tree algorithm increases the accuracy of the system. This paper presents Combination of Clustering with Frequent-Sequence Tree approach which avoids costly repeated database scans and candidate generation.

The organization of this paper is as follows: Section 2 describes the related work of the proposed system. Section 3 describes the Web Mining. Section 4 presents web usage mining and its processes. Section 5 illustrates the proposed system designs. In section 6, system implementation and experimental results are presented. Section 7 has the conclusion of the system.

2. Related Work

This paper presents generation of frequent access patterns by integrating FS Tree with Clustering. There are several approaches in web usage mining area. Nanpoulos et al. [6] proposed a method for discovering access patterns from web logs based on a new type of association patterns. They handle the order between page accesses, and allow gaps in sequences. They use a candidate generation algorithm that requires multiple scans of the database.

Yang et al. [7] presented an application of web log mining that combines caching and prefetching to improve the performance of internet systems. In this work, association rules are mined from web logs using an algorithm called Path Model Construction and then they are used to improve the GDSF caching replacement algorithm. These association rules assumes order and adjacency information among page references. Han et al. [4] proposed a technique that avoids the costly process of candidate generation by adapting a pattern growth method that uses a highly condensed data structure to compress

the database. The proposed technique discovers unordered frequent item sets. However, is does not support the type of sequences we are interested in. This work is similar to [4] in that it uses a condensed data structure and avoid expensive candidate generation. Yet our approach takes order among input items (page references) into consideration.

The proposed system uses the FS-Miner with Clustering approach. FS-Miner avoids the multiple scanning time and reduce counting step. Combining Clustering into FS-Miner makes the higher accuracy of the web usage mining process.

3. Web Mining

Web Mining can be defined as the use of data mining techniques to automatically discover useful knowledge from the Web. It is a converging area from several research communities such as Information Retrieval, Database, Machine Learning, and Natural Language Processing. According to the data type, Web mining is divided into three categories: Web content mining, Web structure mining, and Web usage mining.

1. Web Content Mining

Web content mining is the process of extracting useful information from the contents of web documents. Content data is the collection of facts a web page is designed to contain. It may consist of text, images, audio, video, or structured records such as lists and tables.

2. Web Structure Mining

The structure of a typical web graph consists of web pages as nodes, and hyperlinks as edges connecting related pages. Web structure mining is the process of discovering structure information from the web.

3. Web Usage Mining

Web usage mining is the application of data mining techniques to discover interesting usage patterns from web usage data, in order to understand and better serve the needs of web-based applications. Usage data captures the identity or origin of web users along with their browsing behavior at a web site.

4. Web Usage Mining

While Web structure mining and Web content mining exploit the real or primary data on the WWW, Web usage mining works on the secondary data such as Web server access logs, proxy server logs, browser logs, user profiles, cookies, user queries, and bookmark data. The information provided by the Web resources can be used to form different data abstractions, e.g., users' click streams and page views. Web usage mining aims at utilizing data mining techniques to discover the usage patterns from these secondary data and better fulfill the needs of Web-based applications. Typically, the usage mining is defined as a three-phase process: data preprocessing, pattern discovery, and pattern analysis. Web usage mining architecture is shown in Figure 1.

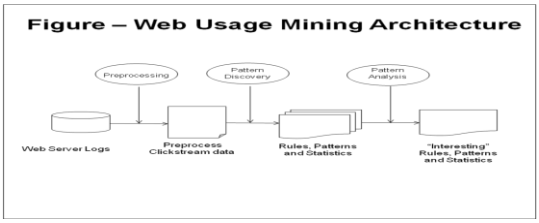

Figure 1: Web Usage Mining Architecture

4.1 Steps of Web Usage Mining

Web usage mining includes the following steps:

- **Data Preprocessing**: retrieves raw data from the Web resources, and automatically selects and preprocesses the retrieved data. It includes any kind of transformation of the original raw data. Figure 2 shows the steps of data preprocessing.
- **Pattern Discovery**: discovers knowledge from the pre-processed data. Machine learning and data mining procedures are carried out at this stage.
- **Pattern Analysis and Applications**: validates and post-processes the discovered patterns.

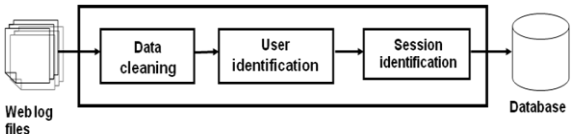

Figure 2: Preprocessing Step

5. Proposed System

The proposed system presents the web usage mining by combination of clustering and FS-Tree algorithm. It is the process of generating frequent sequences from the sequence database. Web log file consists of two steps: Preprocessing Step and Mining Step. In the Preprocessing Step, sequence database, is generated. Sequence database is clustered into groups of similar transactions using K-Mean clustering algorithm. Then FS-Tree is constructed from each sequence group and frequent sequences are generated from the FS-Tree. The proposed system desig is shown in Figure 3.

5.1 Clustering

Clustering is a pattern discovery algorithm in the Web usage mining stage of Web mining. It is defined as the classification of patterns into groups (clusters) based on similarity in order to improve common Web activities. Distance-based clustering involves determining a distance measure between pairs of data objects, and then grouping similar objects together into clusters. This system will use K-Means clustering algorithm.

5.1.1 K-Means Distance Measure

A common clustering algorithm is *k*-means clustering algorithm. It is distance based, unsupervised and partition. It is the simplest and most commonly used clustering algorithm, especially with large data sets. It involves:

1. Define a set of items (n-by-p data matrix) to be clustered.
2. Define a chosen number of clusters (*k*).
3. Randomly assign a number of items to each cluster.

The *k*-means clustering repeatedly performs the following until convergence is achieved:

1. Calculate the mean vector for all items in each cluster.
2. Reassign the items to the cluster whose center is closest to the item.

5.1.2 Cosine Similarity Algorithm

Cosine Similarity Algorithm is used in the calculation of (distance) Similarity between user sessions. User clickstream pages are converted into vectors. The process of cosine (distance) similarity is as follows:

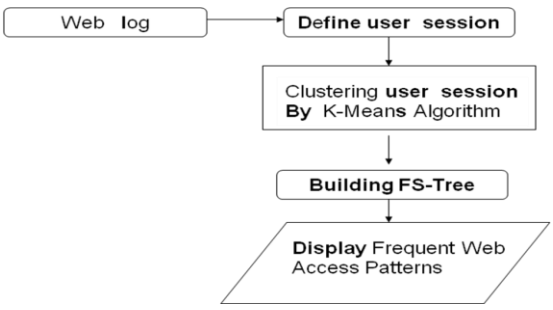

Figure 3: Proposed System Design

5.2. Mining by FS-Tree

Mining by FS-Tree consists of three main components-

1. Tree structure with a special root node R and a set of sequence prefix sub trees as children R.
2. A header table HT that stores information about frequent link by using appropriate minimum support value.
3. A non-frequent links table NFLT that stores information about non-frequent links.

5.2.1 FS-Tree Construction

In the frequent sequence mining algorithm, there are four main steps that are performed for only frequent links in the header table (HT):

- Extracting the derived Paths- For link, for each path in the FS-tree that contains, its path prefix that ends at this edge is extracted. These paths are called derived paths of the link.
- Constructing conditional sequence base- Given the set of derived paths of link extracted in previous step then the conditional sequence base is constructed for a link by setting the frequent count of each link in the path to the count of link.
- Constructing conditional FS-tree- Given the conditional base for link, a tree is created and inserted each of the paths from the conditional base of this link into it in a backward manner. Necessary nodes and edges are created and shared when possible (incrementing edges counts). This tree is called the conditional FS-tree for this link.
- Extracting Frequent sequences- Given a conditional FS-tree of a link, this system will perform a depth first traversal for that tree and return only sequences satisfying minimum support count.

6. System Implementation

Web usage mining by combining of FS-Tree and Clustering is implemented using Java programming language. Jdk version 1.5 is used. It consits of three main steps, Preprocessing, Clustering and Mining by FS-Tree. Web log used in this system is obtained from web log files of web proxy server. The example scenario is shown in the following section. After preprocessing step, we got the user session as in Table 1.

Table 1: User Session

SID	User Session
S0	P0, P1, P2, P3, P6, P8
S1	P1, P2, P3
S2	P2, P3, P4

S3	P2, P3, P4, P5, P0, P1, P4
S4	P1, P3, P4, P7, P8
S5	P5, P1, P2, P6, P8, P4
S6	P3, P8, P4
S7	P0, P8, P4, P7
S8	P1, P3, P4, P7
S9	P3, P6, P8
S10	P4, P5, P0, P1, P2

To clusters, the similarity of the user session is calculated by using cosine similarity algorithm. K-means algorithm involves:

- Define a set of items (n-by-p data matrix) to be clustered.

Table1.1 :(n-by-p data matrix)

Session / page	P0	P1	P2	P3	P4	P5	P6	P7	P8
S0	1	1	1	1	0	0	1	0	1
S1	0	1	1	1	0	0	0	0	0
S2	0	0	1	1	1	0	0	0	0
S3	1	1	1	1	2	1	0	0	0
S4	0	1	0	1	1	0	0	1	1
S5	0	1	1	0	1	1	1	0	1
S6	0	0	0	1	1	0	0	0	1
S7	1	0	0	0	1	0	0	1	1
S8	0	1	0	1	1	0	0	1	0
S9	0	0	0	1	0	0	1	0	1
S10	1	1	1	0	1	1	0	0	0

Similarity matrix is obtained as in Table 1.2.

Table1.2: Similarity matrix calculate between user sessions

	s0	**s1**	**s2**	**s3**	**s4**	**s5**	**s6**	**s7**	**s8**	**s9**	**s10**
s0	**1**	**.71**	**.47**	**.54**	**.55**	**.67**	**.47**	**.41**	**.41**	**.71**	**55**
s1	**.71**	**1**	**.67**	**.58**	**.52**	**.47**	**.33**	**0**	**.58**	**.33**	**.52**
s2	**.47**	**.67**	**1**	**.77**	**.52**	**.47**	**.58**	**.29**	**.29**	**.33**	**.52**
s3	**.54**	**.58**	**.77**	**1**	**.60**	**.68**	**.58**	**.50**	**.67**	**.19**	**.87**
s4	**.55**	**.52**	**.52**	**.60**	**1**	**.55**	**.77**	**.67**	**.89**	**.52**	**.40**
s5	**.67**	**.47**	**.47**	**.68**	**.55**	**1**	**.47**	**.41**	**.41**	**.47**	**.73**
s6	**.47**	**.33**	**.58**	**.58**	**.77**	**.47**	**1**	**.58**	**.33**	**.67**	**.26**
s7	**.41**	**0**	**.29**	**.50**	**.67**	**.41**	**.58**	**1**	**.25**	**.29**	**.45**
s8	**.41**	**.58**	**.29**	**.67**	**.89**	**.41**	**.33**	**.25**	**1**	**.29**	**..45**
s9	**.71**	**.33**	**.33.**	**.19**	**.52**	**.47**	**.67**	**.29**	**.29**	**1**	**.54**
s10	**.55**	**.52**	**.52**	**.87**	**.40**	**.73**	**.26**	**.45**	**.45**	**.54**	**1**

- Define a chosen number of clusters (*k*)

S1 and S7 are two initial sessions for 2 different clusters.

They are selected because their similarity value is minimum (0).

Therefore, starts from the beginning of the sessions except S1 and S7, sessions are assigned to clusters as follows:

For S0, sim (S1, S0) = 0.71 and sim(S7, S0) = 0.41, therefore, S0 is more similar to S1 and it is assigned to Cluster 1.

Then calculate the mean for the Cluster 1according to K-means algorithm.

Session/ pages	P0	P1	P2	P3	P4	P5	P6	P7	P8
Cluster1									
S1	0	1	1	1	1	0	0	0	0
S0	1	1	1	1	0	0	1	0	1
mean	.5	1	1	1	0	0	.5	0	.5

Then,calculate Cosine Similarity Algorithm .

After clustering with number of cluster (k) is set to 2, we got 2 clusters as in Table 2.

Table 2: Clusters of User sessions in Table 1

Cluster 1	S1, S0, S2, S3, S5, S8, S9, S10
Cluster 2	S7, S4, S6

FS-Tree miner is applied to each cluster. Cluster 1 will be used in the following section to apply in FS-Tree miner. User Sessions, HT and NFLT shown in Table 3(a), (b) and FS-Tree Construction is shown in Figure 4.

Table 3 (a): User Sessions

SID	**User Session**
S1	P1, P2, P3
S0	P0, P1, P2, P3, P6, P8
S2	P2, P3, P4
S3	P2, P3, P4, P5, P0, P1, P4
S5	P5, P1, P2, P6, P8, P4
S8	P1, P3, P4, P7
S9	P3, P6, P8

S10	P4, P5, P0, P1, P2

After pruning with minimum support 2, HT and NFLT tables will be as follows:

Table 3(b): Header Table and NFLT

Header Table			**NFLT**	
Link	**Count**		**Link**	**Count**
P1-P2	4		P1-P4	1
P2-P3	4		P5-P1	1
P0-P1	3		P2-P6	1
P3-P6	2		P8-P4	1
P6-P8	3		P1-P3	1
P3-P4	3		P4-P7	1
P4-P5	2			
P5-P0	2			

Then after building the HT and NFLT, FS-Tree is constructed. Links from HT table is populated in the FS-Tree. Figure 4 presents above HT table is construted into FS-Tree.

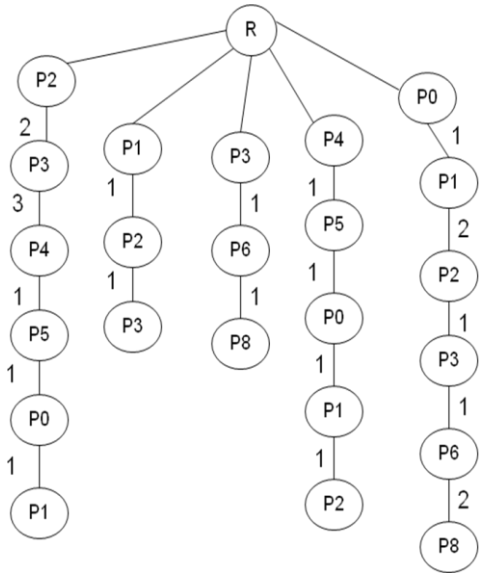

Figure 4: FS-Tree constructed from Table 1 (a) and (b)

6.1 Experimental Result

This system is tested with several web log files. We have set number of cluster (k) value into different values. In the FS-Miner step, threhold value (minimum support) is also set to different values. Web logs used in this system are obtained from web log files of web proxy server. Web log file name is epa.http. For the web log file with 12000 log entries, in the user sessions, we have got 584 user sessions. The experimental results of the system with different k values and different minimum support values are shown in Table 4.

Table 4: The experimental results of the system

No.	No. of cluster (k)	Min support	Accuracy
1	4	2	89.3%
2	4	3	91.4%
3	4	4	90.46%
4	5	2	90.57%
5	5	3	94.02%
6	5	4	93.08%

7. Conclusion

This system implements to generate frequent sequence pattern by constructing FS-Tree after applying into clustering. Our system constructs a compressed data structure (FS-Tree) that stores potentially frequently sequences and uses that structure to discover frequent sequences. Our approach also allows interactive response to changes to the system minimum support. K-Mean Clustering increases the accuracy of the system. Our system will predict and pre-fetch web pages that the user is likely to visit.

8. References

[1] Borges, J. and Levene, M. "A Dynamic Clustering-Based Markov Model for Web Usage Mining", May 26, 2004.

[2] Britos, P., Martinelli, D., Merlino, H and García-Martínez R, "Web Usage Mining Using Self Organized Maps", In proceeding of IJCSNS International Journal of Computer Science and Network Security, VOL.7 No.6, June 2007

[3] El-Sayed, M. Ruiz, C. and Rundensteiner E. A. "FS-Miner: Efficient and Incremental Mining of Frequent Sequence Patterns in Web logs", WIDM'04, November 12–13, 2004, Washington, DC, USA.

[4] Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In SIGMOD, pages 1–12, May 2000. [8] Hettich, S. and Bay, S. D. The UCI KDD Archive. Irvine.

[5] Mustapha, N., Manijeh Jalali and Mehrdad Jalali, "Expectation Maximization Clustering Algorithm for User Modeling in Web Usage Mining Systems", European Journal of Scientific Research, ISSN 1450-216X Vol.32 No.4 (2009), pp.467-476, EuroJournals Publishing, Inc. 2009.
http://www.eurojournals.com/ejsr.htm

[6] Nanopoulos, D. Katsaros, and Y. Manolopoulos. Effective prediction of web-user accesses: A data mining approach. In WEBKDD Workshop, San Francisco, CA, Aug. 2001.

[7] Q. Yang, H. H. Zhang, and I. T. Y. Li. Mining web logs for prediction models in WWWcaching and prefetching. In KDD, pages 473–478, 2001.