WEB USER NAVIGATION PATTERN MINING APPROACH BASED ON GRAPH PARTITIONING ALGORITHM

[1,2]Mehrdad Jalali, [3]Norwati Mustapha, [4]Ali Mamat, [5]Md. Nasir B Sulaiman

[1]PhD Candidate, Department of Computer Science, University of Putra Malaysia, Malaysia

[2]Department of Software Engineering, Islamic Azad University of Mashhad, Iran

[3]Asstt Prof., Department of Computer Science, University of Putra Malaysia, Malaysia

[4]Assoc. Prof., Department of Computer Science, University of Putra Malaysia, Malaysia

[5]Assoc. Prof., Department of Computer Science, University of Putra Malaysia, Malaysia

Email: {jalali,norwati,ali,nasir@fsktm.upm.edu.my}

ABSTRACT

We present a study of the web based user navigation patterns mining and propose a novel approach for clustering of user navigation patterns. The approach is based on the graph partitioning for modeling user navigation patterns. In order to mining user navigation patterns, we establish an undirected graph based on connectivity between each pair of the web pages. Moreover, we propose novel formula for assigning weights to edges of the graph. The experimental results represent that the approach can improve the quality of clustering for user navigation pattern in web usage mining systems. These results can be use for predicting user's next request in the huge web sites.

Keywords: *Web Usage Mining, User Navigation Pattern, Clustering, Graph Partitioning*

1. INTRODUCTION

Modeling user web navigation data is challenging task that is continuing to gain importance as the size of the web and its user-base increase. A web navigation behavior is helpful in understanding what information of online users demand. Following that, the analyzed results can be seen as knowledge to be used in intelligent online applications, refining web site maps, web based personalization system and improving searching accuracy when seeking information. Nevertheless, an online navigation behavior grows each passing day, and thus extracting information intelligently from it is a difficult issue. Web Usage Mining (WUM) is the process of extracting knowledge from Web user's access data by exploiting Data Mining technologies[1]. It can be used for different purposes such as personalization, system improvement and site modification.

In our system, user navigation patterns are described as the common browsing behaviors among a group of users. Since many users may have common interests up to a point during their navigation, navigation patterns should capture the overlapping interests or the information needs of these users. In addition, navigation patterns should also be capable to distinguish among web pages based on their different significance to each pattern.

In our paper, we propose a novel approach for clustering of user navigation patterns based on the graph partitioning for modeling user navigation patterns. For the clustering of user navigation patterns we create an undirected graph based on connectivity between each pair of web pages and we propose novel formula for assigning weights to edges in such a graph. The experimental results represent that our approach can improve the quality of clustering for user navigation pattern in web usage mining systems. These results can be use for predicting user's next request in the huge web sites.

The rest of this paper is organized as follows: In section 2, we review some researches that advance in navigation pattern mining. Section 3 describes the proposed method for the clustering of navigation pattern mining. Results of an experimental evaluation are reported in section 4. Finally, section 5 summarizes the paper and introduces future work.

2. RELATED WORKS

Recently, several Web Usage Mining algorithms have been proposed to mining user navigation behavior. In the following we review some of the most significant navigation pattern mining systems and algorithm in web usage mining area that can be compared with our system.

A partitioning method was one of the earliest clustering methods to be used in Web usage mining by Yan et al.[2]. They used an incremental algorithm that produces high quality clusters. Each user session is represented by an *n*-dimensional feature vector, where *n* is the number of Web pages in the session. The value of each feature is a weight, measuring the degree of interest of the user in the particular Web page. The calculation of this figure is based on a number of parameters, such as the number of times the page has been accessed and the amount of time the user spent on the page. Based on these vectors, clusters of similar sessions are produced and characterized by the Web pages with the highest associated weights. The characterized sessions are the patterns discovered by the algorithm. One problem with this approach is the calculation of the feature weights. The choice of the right parameter mix for the calculation of these weights is not straightforward and depends on the modeling abilities of a human expert.

A partitioning graph theoretic approach is presented by Perkowitz and Etzioni [3], who have developed a system that helps in making Web sites adaptive, i.e., automatically improving their organization and presentation by mining usage logs. The core element of this system is a new clustering method, called cluster mining, which is implemented in the PageGather algorithm. PageGather receives user sessions as input, represented as sets of pages that have been visited. Using these data, the algorithm creates a graph, as signing pages to nodes. An edge is added between two nodes if the corresponding pages co-occur in more than a certain number of sessions. Clusters are defined either in terms of cliques, or connected components. Clusters defined as cliques prove to be more coherent, while connected component clusters are larger, but faster to compute and easier to find. A new index page is created from each cluster with hyperlinks to all the pages in the cluster. The main advantage of PageGather is that it creates overlapping clusters. Furthermore, in contrast to the other clustering methods, the clusters generated by this method group together characteristic features of the users directly. Thus, each cluster is a behavioral pattern, associating pages in a Web site. However, being a graph based algorithm, it is rather computationally expensive, especially in the case where cliques are computed.

Another partitioning clustering method is employed by Cadez et al. [4] in the WebCANVAS tool, which visualizes user navigation paths in each cluster. In this system, user sessions are represented using categories of general topics for Web pages. A number of predefined categories are used as a bias and URLs from the Web server log files are assigned to them, constructing the user sessions.

The Expectation-Maximization (EM) algorithm, based on mixtures of Markov chains is used for clustering user sessions. Each Markov chain represents the behavior of a particular subgroup. EM is a memory efficient and easy to implement algorithm, with a profound probabilistic background. The EM algorithm is also employed by Anderson et al. [5] in two clustering scenarios, for the construction of predictive Web usage models. In the first scenario, user navigation paths are considered members of one or more clusters, and the EM algorithm is used to calculate the model parameters for each cluster. The probability of visiting a certain page is estimated by calculating its conditional probability for each cluster. The resulting mixture model is named Naive Bayes mixture model since it is based on the assumption that pages in a navigation path are independent given the cluster. The second scenario uses a similar approach to [4]. Markov chains that represent the navigation paths of users are clustered using the EM algorithm, in order to predict subsequent pages.

Baraglia and Palmerini proposed a WUM system called SUGGEST, that provide useful information to make easier the web user navigation and to optimize the web server performance [6, 7]. SUGGEST adopts a two levels architecture composed by an offline creation of historical knowledge and an online engine that understands user's behavior. As the requests arrive at this system module it incrementally updates a graph representation of the Web site based on the active user sessions and classifies the active session using a graph partitioning algorithm. Potential limitation of this architecture might be: a) the memory required to store Web server pages is quadratic in the number of pages. This might be

a severe limitation in large sites made up of millions of pages; b) it does not permit us to manage Web sites made up of pages dynamically generated.

All of these works attempt to find architecture and algorithm to improve quality of clustering, but the quality still does not meet satisfaction. In our work we advance a model and propose novel graph partitioning algorithm for improving accuracy of user navigation clustering.

3. NAVIGATION PATTERNS MINING

Generally, several pretreatment tasks need to be done before performing web mining algorithms on the Web server logs. Data pretreatment in a web usage mining model (Web-Log preprocessing) aims to reformat the original web logs to identify all web access sessions. The Web server usually registers all users' access activities of the website as Web server logs. Due to different server setting parameters, there are many types of web logs, but typically the log files share the same basic information, such as: client IP address, request time, requested URL, HTTP status code, referrer, etc.

For this work, these include data cleaning, user differentiation and session identification. These preprocessing tasks are the same for any web usage mining problem. Navigation pattern modeling is done after preprocessing tasks.

3.1. Navigation Pattern Modeling

To model navigational patterns we use proposed algorithm in [7] but with some modifications. The main difference is in the definition of the undirected graph *M*. In our contribution the degree of connectivity in each pair of pages depends on two main factors: the time position of two pages in a session and the occurrence of two pages in a session.

We proposed an algorithm for modeling the pages accesses information as an undirected graph *M= (V, E)*. The set *V* of vertices contains the identifiers of the different pages hosted on the Web server. We propose a weight measure for approximating the connectivity degree of each two web pages in sessions. First, let us introduce two concepts related to this measure, ''Time Connectivity'' and ''Frequency''.

''Time Connectivity'' measures the degree of visit ordering for each two pages in a session. In (1) we compute this measure with this novel formula:

$$TC_{a,b} = \frac{\sum_{i=1}^{N} \frac{T_i}{T_{ab}} \times \frac{f_a(k)}{f_b(k)}}{\sum_{i=1}^{N} \frac{T_i}{T_{ab}}} \quad (1)$$

Where T_i is time duration in *i*-th session, that containing both pages *a* and *b*, T_{ab} is the difference between requested time of page *a* and page *b* in the session. we consider $f(k)=k$ if web page appears in position *k* .However, A different form for *f* could be chosen .For instance it is also possible to increase the importance of the position of each two pages in a session by taking $f(k) = k^2$. The formula is also normalized so all values for time connectivity are between 0 and 1.

''Frequency'' measures the occurrence of two pages in each sessions. In (2) we compute this:

$$FC_{a,b} = \frac{N_{ab}}{Max\{N_a, N_b\}}$$

Where N_{ab} is the number of sessions containing both page *a* and page *b*. N_a and N_b are the number of session containing only page a and page b. and this formula also has the values between 0 and 1.

In our system, ''Time Connectivity'' and ''Frequency'' are considered two strong indicators of the degree of connectivity for each pair of web pages. Therefore, in the weight measure we devised, ''Time Connectivity'' and ''Frequency'' are valued equally. We use the harmonic mean of ''Time Connectivity'' and ''Frequency'' to represent the connectivity of each two pages, shown as (3).we take this formula for weight of each edges in the undirected graph.

$$W_{a,b} = \frac{2 \times TC_{ab} \times FC_{ab}}{TC_{ab} + FC_{ab}} \quad (3)$$

The data structure can be used to store the weights is an adjacency matrix *M* where each entry M_{ab} contains the value W_{ab} computed according to (3) .To limit the number of edge in such graph ,element of M_{ab} whose value is less than a threshold are to little correlated and thus discarded. This threshold is named as *MinFreq* in this contribution.

3.2. Clustering Algorithm Based On Graph Partitioning

We apply graph partitioning algorithm to finds groups of strongly correlated pages by partitioning the graph according to its connected components. Starting from a Vertex a Depth First Search (DFS) on the graph induced by *M* is applied to search for the connected component reachable from this vertex. Once the component has been found, the algorithm checks if there are any nodes not considered in the visit. If so, it means that a previously connected component has been split, and therefore, it needs to be identified. To do this the DFS is again applied by starting from one of the nodes not visited. In the worst case, when all the URLs are in the same cluster, the cost of this algorithm will be linear in the number of edges of the complete graph *G*. Before the clusters put into the navigational pattern profile, the clusters are ranked based on values store in the matrix *M*. The clustering algorithm is shown in Algorithm 1.

Input:

- Cleaned, filtered, and sessionized Log file.
- MinFreq.
- MinClusterSize.

Output:

- A list of Clusters C

```
L[p] = P ;    // Assign all URLs to a list of web pages.
for each (Pi, Pj) € L[p] do    // for all pair of web pages
        M (i,j)=WeightFormula (Pi, Pj);    //computing the weight based on (3)
        Edge (i,j)=M (i,j);
end for

//There is an undirected Graph (E, V)

for all Edge (u, v) € Graph (E, V) do  // removing all edges that its weight is bellow than MinFreq
        if Edge  (u ,v) < MinFreq then
                remove (Edge (u, v));
        end if
end for

for all vertices(u) € do
        Cluster [i]=DFS (u);          //doing the Depth first search algorithm
        if cluster[i] <MinClusterSize  // removing the cluster that its length is bellow than MinClusterSize
                remove (Cluster[i]);
        end if
        i=i+1
end for

return (Cluster);
```

Algorithm 1: The clustering phase

4. EXPERIMENTAL EVALUATION

Measuring the quality of the clustering in navigation patterns mining systems needs to characterize the quality of the results obtained.

The experimental evaluation was conduced using DePaul University CTI Log file. The only cleaning step performed on this data was the removal of references to auxiliary files (e.g., image files). No other cleaning or preprocessing has been performed in the first phase. The data is in the original log format used by Microsoft IIS. The characteristics of the dataset we used are given in table 1.

Table 1. Dataset used in the experiments.

Dataset	Size (Mbytes)	Records (thousand)	Period (days)
CTI	260	1051	14

Table 2 shows some basic statistics on user and sessions after cleaning, filtering and sessioning the CTI dataset.

Table 2. Dataset after preprocessing.

Number of users	Size (Mbytes)	number of sessions	Number of repeat users
5446	10	20950	2734

All evaluation tests were run on a dual processor Intel CPU 1.8 GHz Pentium 4 with 2GBytes of RAM, operating system Windows XP. Our implementations run on .net framework 2.

Figure 1 shows the number of cluster Based on MinFreq. In Our evaluation *MinClusterSize*=1 it means the clusters lower than this threshold are discarded because considered not significant. In this figure we compare our work with latest contribution in clustering by graph partitioning[7] .As we represent when the graph becomes highly disconnected; the clusters found are smaller than *MinClusterSize* threshold and thus discarded. Therefore the total number of clusters found does not increase. Clustering by our algorithm can find more clusters based on *MinFreq*. It means the real connected components in our clustering are more than other contributions.

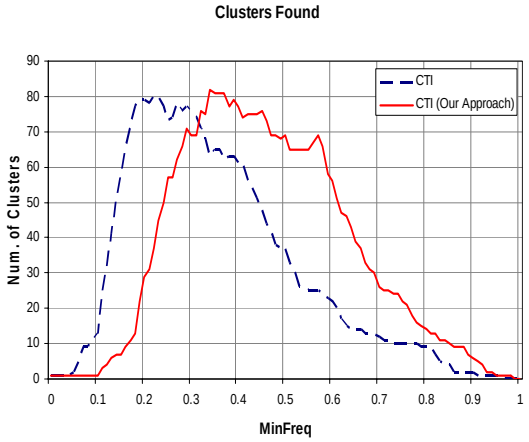

Figure 1. Number of cluster found.

Figure 2 shows the percentage of pages that do not belong to any cluster. In the other hand this figure shows the number of outliers and we can observe that as *MinFreq* increase the percentage of outliers also grows. In this work the number of outliers is below than other contributions.

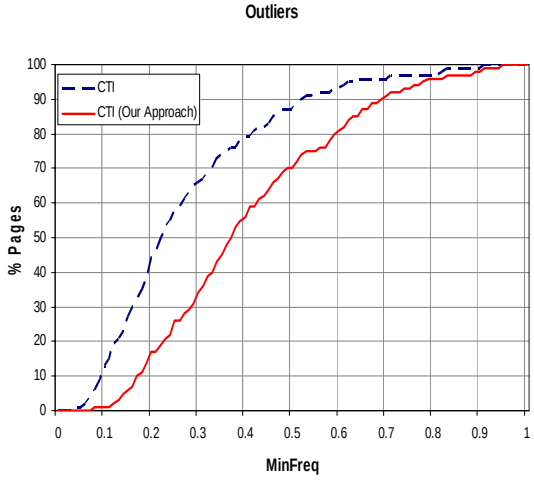

Figure 2. Percentage of outliers.

To evaluate the quality of the clusters produced by our approach we used the visit coherence index that allows quantifying a session intrinsic coherence. It measures the percentage of pages inside a user session which belong to the cluster representing the session considered.

There are some basic approaches to evaluating the accuracy of clustering. One of these approaches is *K*-fold cross-validation. In this evaluation method we first split the data into *K* subset of equal size. Then each subset in turn is used for testing and the remainder for training. In our work we perform 10-fold cross-validation using the server logs from DePaul University web site (www.cs.depaul.edu).

In each of the 10 iterations, the data set divided into training (90%) and evaluation (10%) datasets. The training set is used to generate the models based on our clustering algorithm while the evaluation set is used to test the generated model. We use evaluation methodology introduced in [7].

To verify if the coherence hypothesis holds for every session *i* in evaluation dataset, we define a parameter β which measures the number of web pages in session *i* that belongs to representative cluster for that session.

$$\beta_i = \frac{\left|\{p \in S_i | p \in C_i\}\right|}{N_i} \quad (4)$$

Where p is a page, S_i is the *i*-th session, C_i is the cluster representiting *i*, and N_i is the number of pages in the *i*-th session. The average value for β over all N_s sessions contain inside the dataset partition treated is given by:

$$\alpha = \frac{\sum_{i=1}^{N_s} \beta_i}{N_s} \quad (5)$$

Figure 3 plots α as a function of *MinFreq*, in percent. For small values of *MinFreq* almost all pages in every session belongs to the same cluster. As we represented in this figure, our approach has higher degree of coherency according to *MinFreq*.

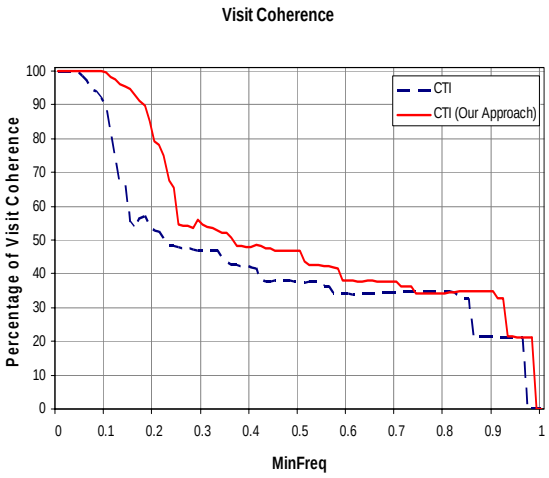

Figure 3. Coherence of visit.

5. CONCLUSION

In this paper, we advance an algorithm and proposed a novel contribution to clustering user navigation patterns. We used some evaluation methodology that can be used to evaluate the quality of the clusters found. The experimental results represent that our approach can improve the quality of clustering for user navigation pattern in web usage mining systems.

For the future, we would perfect the algorithm and apply some classification methods for classifying user request. This can be used in WUM based prediction systems.

REFERENCES

[1] B. Mobasher, R. Cooley, and J. Srivastava, "Automatic personalization based on Web usage mining," *Communications of the ACM,* vol. 43, pp. 142-151, 2000.

[2] T. W. Yan, M. Jacobsen, H. Garcia-Molina, and U. Dayal, "From user access patterns to dynamic hypertext linking," *Computer Networks and ISDN Systems,* vol. 28, pp. 1007-1014, 1996.

[3] M. Perkowitz and O. Etzioni, "Towards adaptive Web sites: Conceptual framework and case study," *Artificial Intelligence,* vol. 118, pp. 245-275, 2000.

[4] I. Cadez, D. Heckerman, C. Meek, P. Smyth, and S. White, "Visualization of navigation patterns on a Web site using model-based clustering," *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining,* pp. 280-284, 2000.

[5] C. R. Anderson, P. Domingos, and D. S. Weld, "Adaptive Web Navigation for Wireless Devices," *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence,* pp. 879–884, 2001.

[6] R. Baraglia and F. Silvestri, "Dynamic personalization of web sites without user intervention," *Communications of the ACM,* vol. 50, pp. 63-67, 2007.

[7] R. Baraglia and F. Silvestri, "An Online Recommender System for Large Web Sites," *Proceedings of the Web Intelligence, IEEE/WIC/ACM International Conference on (WI'04)-Volume 00,* pp. 199-205, 2004.