Discovering Web Usage Patterns - A Novel Approach

K. Sudheer Reddy[1], Ch.N. Santhosh Kumar[2], V. Sitaramulu[2], and M. Kantha Reddy[3]

[1] Dept. of CSE,
Acharya Nagarjuna University, Guntur, AP, India
sudheercse@gmail.com
[2] Department of Computer Science & Engineering
Swarna Bharathi Institute of Science & Technology, Khammam, AP, India
vsitaramu.1234@gmail.com, santhosh_ph@yahoo.in
[3] IUCEE, India
kanthareddy_m@yahoo.com

Abstract. Pattern mining is one of the most pivotal steps in data mining; pattern mining immediately comes after the preprocessing phase of WUM. Pattern discovery deals with the sorted set of data items presented as part of the sequence. Pattern mining, users can recognize the web paths follow on a web site easily. The aim of this research discovers the patterns which are most relevant and interesting by using a Web usage mining process. The server web logs aids are the input to this process. Our target is to discover users' behavior, who has visited the web sites for less number of times. We have enlightened a method for clustering, based on the pattern summaries. We have conducted intense experiments and the results are shown in this paper.

Keywords: Web usage mining, preprocessing, pattern discovery, sequential patterns, clustering, patterns summary.

1 Introduction

Analyzing the behavior of the web users' is also known as Web Usage Mining (WUM). WUM is an active research area which entails in adapting the mining techniques to the records of access log files. These access web log files collect numerous types of data include IP address of the host, the requested URL. The date and other required information about the user navigation into web. The techniques of WUM provide most interesting knowledge about the numerous web user behaviors in order to excerpt relationships in the recorded data. Amongst the techniques available, the sequential patterns are predominantly well adapted to the web log study. Sequential patterns extraction on a web access log file, is theoretical to provide the thoughtful relationship: "On SRKREC Web Site, 23% of users visited the homepage consecutively, the available resources page, the RSC offers, the RSC missions and finally the past RSC competitive selection". Exhibiting this type of behavior is an assumption, because pattern extraction on a web access log file also infers, managing several problems, as listed below:

S.C. Satapathy et al. (Eds.): Proc. of Int. Conf. on Front. of Intell. Comput., AISC 199, pp. 181–188.
DOI: 10.1007/978-3-642-35314-7_21

- The number of records in the web server log file is lowered due to user's computer cache and the proxies.
- The entries of the log file can be reduced and also reduce the user navigations is possible with the aid of research engines. As a result of this, the user can directly access a definite portion of the web site.
- The number of portions visited on the site is compared to the entire site.
- The user's representativeness who navigates the web through that part is compared to the whole site users.

If the web caching problems are to be solved [5], the representativeness requires a sturdy study. To exemplify our goal, let's consider sequential patterns we are supposed to get. Due to the minor size of the "job offer" part of the web site, users requesting a page on that part represent only 0.3% of users on the entire web site. In the similar way, users navigating on the "research" part of the research assignment represent only 0.003% of all the users. So, the study of WUM on this type of site has to manage this specific representativeness in order to provide sufficient results. Our objective is to showcase that a classical pattern mining technique is unable to provide web users behaviors with such a weak support.

Furthermore, we present a unique method for discovering behavior of all web users of a Web site. We tag our test and experiments and then conclude the paper.

2 Principle

We propose a methodology and describe the outline as mentioned here: discovering the clusters of the web users (web users are typically grouped by the user's behavior) and then analyzing the user navigations by means of the sequential pattern mining process. Therefore, our methodology relies on two steps. The first step targets at splitting the web log into sub-logs, hypothetical to represent several separated actions. The second step targets at analyzing user behavior recorded in each sub-log.

The key principle involved in our method is described as given below:

1. Extracting the patterns on the original log.
2. These extracted patterns can be clustered.
3. Dividing the web log according to the clusters obtained, each sub-log encompasses user sessions from the original web access log, approving at least one of the user behavior of the cluster which permits to create this sub-log. A distinct sub-log is created then to collect the user sessions from the original sub-log which doesn't correspond to the cluster from an earlier step.
4. Apply the whole process recursively, for each sub-log.

The below Figure1 graphically illustrates the proposed method. Initially, sequential patterns are to be obtained and the same are to be clustered, it is shown from C1 to Cn in the figure. Then, the web access log is split into various sub-logs, from SLog1 to SLogn upon these clusters. To finish, a sub-log SLogn+1 is created for the several user sessions which cannot be consistent with a behavior from the web log. The quality of the results produced by our methodology will depend on the sub-log file. In detail, the initial sub-logs comprise the most represented categories of the web users. Hence, they are interesting, but the most interesting patterns discovery will derive from the study of the uncluster sessions of the sub-log SLogn+1. Seeing this sub-log as a new original web log, and recursively repeating the process will allow users to discover behaviors with the minimal support. To acquire reliable results, our method suitable on a “the quality of the split proposed for a log”. The split depends on on the clustering done on the discovered patterns in the original log. In the following section, we describe briefly the methodology that we have used in cluster patterns.

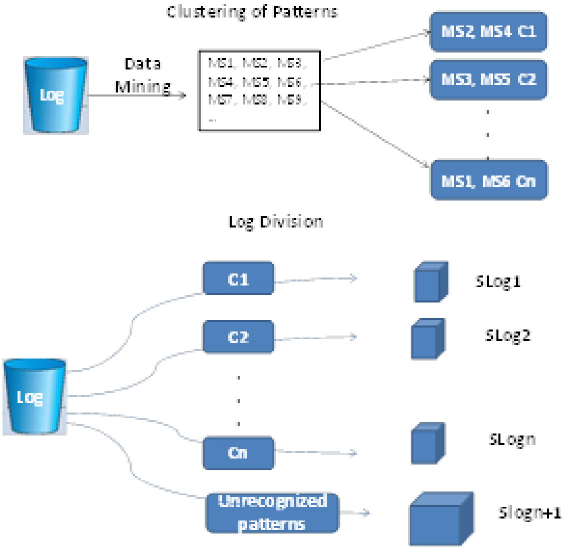

Fig. 1. The Principle of Discovery

3 Clustering Based on Pattern Generalization

We have deliberated several methods of clustering for sequential patterns discovery. We propose and describe here the utmost efficient method for sequential pattern clustering that we have used. The clustering approach which is used in this study is grounded on a method developed by [8] in 2000 for indexing the web sequences in the perception of Web-based recommender systems. The efficiency is based on the neural approach for such method and its effectiveness relies on usage of summarized descriptions for sequential patterns: these descriptions are based on Web access sequences generalization.

3.1 Neural Method

The proposed neural clustering method is based on [8] a framework for supporting the reuse of the various past experiences using the integrated object oriented organization. This approach has been successfully applied on browsing behaviors of thematic repertory, in huge organizations. This is purely based on hybrid model and composed from the connexionist part [5] and unadulterated flat memory compound of patterns’ groups.

A threshold TSi is linked to each prototype, which will be altered during the knowledge step. If a user pattern is introduced in the network, drops in the influence region of a prototype, then the prototype will be automatically activated. Such region is determined by set of input vectors acceptable a distance measure which is lower than the threshold. In case, if there is an inactivated prototype, a new prototype is created.

Hence, the structure of a prototype-based network is an evolutionary one in the sagacity that the numerous prototypes at the hidden level is not the priori fixed and might be enlarged during learning step. The prototype is characterized by using its influence region, reference vector, and a set of representing patterns.

4 Experiments and Results

The methods of extraction are written in object oriented language, C++ on a Pentium (3.2 Ghz) PC running a Linux system. The algorithm we have employed is the PSP algorithm [12] for extracting patterns. The neural method and GUI are grasped in Java. For the SRKREC's site, the data was composed over a period of 45 days, while for the other intranets, over a period of 70 days. The narrative of the characteristics (refer Figure 3) is: the number of lines in the web access log is indicated by N, S can be the number of user sessions, the number of filtered URLs is U, the average session length is denote as L, the average number of session URLs is SU. Through our experiments, we are able to bring into interval frequent behaviors, with a comparative representativeness getting feebler and weaker, depending on the sub-log's depth.

	www.srkrec.ac.in	**www.srkrec.ac.in/intranet**
N	12 57 24	17 167 81
S	287 493	437 648
U	46 218	61 398
L	3.5	2.9
SU	4.6	3.2

Fig. 2. Log file characteristics

C1: The user behavior listed here is related to the higher education prospects offered by the SRKREC. The users visit and read higher education page, and then the web page describing the competitive selection and lastly the web pages describing the education opportunities.

<(trv/higheredu/educon.html) (trv/higheredu/educon/oppot.html) (highedu/inplo/index.html) (highedu/inplo/listings/index.html) > (support:0.28%).

Fig. 3. C1 characteristics

C2: This behavior is based on a search for a security fleabag in the system. Generally, these web attacks are programmed once and further shared and used by different groups.

The discovered user behaviors by employing our method cover more than 75 surfing goals on main SRKREC web site and more 130 goals on the intranet site of SRKREC. We stated three important goals here, from job opportunities requests to activities of hacking. Thus, these discovered behaviors demonstrate the success of our methodology in discovering the behaviors.

<(lscripts/root.exe) (c/winnt/system32/cmd.exe)
(..%255c../..%255c../winnt/system32/cmd.exe)
(..%255c../..%255c/..%c1%1c../..%c1%1c../..%c1%1c../winnt/system32/cmd.exe)
(winnt/system32/cmd.exe) (winnt/system32/cmd.exe)(winnt/system32/cmd.exe)>
(support: 0.04%).

Fig. 4. C2 characteristics

4.1 APRIORI ALGORITHM

Apriori is a classic and most sought after algorithm for learning association rules [6],[11] in data mining area is the Apriori algorithm. Apriori approach is designed to work on databases containing various transactions. Apriori uses breadth-first search (BFS) and a tree structure is applied to efficiently count the candidate item sets. The algorithm generates candidate item sets which are of length k from item sets of length k - 1. Then candidates that have infrequent sub patterns are to be pruned. According to the downward closure principle, the candidate set comprises all k-length frequent item sets. Afterwards, it scans transaction database to fix frequent item sets among the candidates.

The Apriori algorithm is an efficient for finding frequent item sets. A level-wise search being implemented using frequent item sets and can be further optimized.

The efficient Apriori algorithm we have introduced is based on Apriori algorithm but introduced efficiency while generating candidates.

The proposed algorithm has several distinctions from the traditional ones.

Unlike other techniques and algorithms, the proposed algorithm mainly focuses on discovering the frequent patterns. This algorithm has two main advantages when compared with the previous algorithms:

1. The linear time and liner space are being used in the algorithm for building and storing the sequences.
2. The new inclusions are continuously being added to the web logs and the sessions which are having removed files should be removed.

The modified Apriori algorithm is described in step-by-step as given below:

Table 1. Modified apriori algorithm – step-by-step process

Step 1	Split the database D into partitions of size n, these partitions are applied on apriori algorithm generation
Step 2	Use the apriori_generation module as mentioned in the algorithm for applying each partition of size n
Step 3	The candidate generations are being applied on each partition as performed in step1. Scan every partition for generating an itemset count. The output of this phase is finding the itemset count.
Step 4	For pruning the itemset, apply min_support module.
Step 5	This same process can be continued until there are no frequent items located in a partition. This process can be repeated from step 2 till step 4.

The steps given in the above matrix are transformed into a computational algorithm by using several procedures and recursive functions.

The following notations used in the algorithm listed below:

- D indicates the database transactions
- L1 denotes the frequent data item sets found in D
- Assuming K=2
- Ck : candidate itemset of size k
- Lk: frequent itemset of size k

```
Procedure Divide_npartitions ( D, size)
{       if Lk-1 • Ø then
        n_partitions = size / ksize;
            Divide_npartitions (D, n_partitions)
 // the database has divided into n no of partitions
        Divide_npartitions (D, size - n_partitions)
         Ck=apriori_generation (Lk-1 , n_partitions, D , size-n_partitions)
                For each partition p   D // scan D for partitions
                        Cp = subset (Ck , p)  // get the sub partition p i.e,
candidates
                        For each candidate c    Cp
                            c.count++;
                          Lk = { c    Cp / c.count • min_sup}
                              k++;
}Return L = UK Lk
Function apriori_generation (Lk-1: frequent (k-1) itemsets , n_partitions, D ,
size-n_partitions)
{       for each itemset $l_1$ Є $L_{k-1;}$
        for each itemset $l_2$ Є $L_{k-1;}$
        if $l_1[1] = l_2[1]$ ^ $l_1[2] = l_2[2]$……… $l_1[k-2] = l_2[k-2]$ ^ $l_1[k-1] = l_2[k-1]$
then
                        c = apply join on $l_1$ , $l_2$;
        for each partition D
                count the frequent itemsets (k-1) itemsets
                for each (k-1) subset s of c
                        if s does not contain k-1 then
                                delete c
                        else
                                add c to $C_k$
} Return $C_k$
End procedure Divide_npartitions
```

Fig. 5. Modified Apriori algorithm based on partitioning technique for improving the effectiveness

In our experiments, the web log files that we have used have been collected from SRKREC's Web server (www.srkrec.ac.in/intranet) during the months of January and February 2012. The size of the two log files is 2.8 MB. We had close to more than two thirds of web requests for the main Web site. We sketch the important characteristics of the initial dataset in the Table 2 given below.

Table 2. Initial Log File description

Characteristic	www.srkrec.ac.in/intranet	Total
Log file size	2812 MB	2812 MB
Number of requests	3046	3046
% of requests	100%	100%

We have applied the data preprocessing methodology on the raw web log files, as discussed and deliberated in chapter 5. Once, the data preprocessing step is successful, the size of the structured web log file, which has user sessions and user visits are reduced to only 532 MB. This presents a total of 32578 visits, from which only 15,246 contained at least two pages. We have selected this visits set with at least two pages, an input for the frequent pattern mining applications.

Table 3. Characteristics of the Structured Web Log File

Characteristic	Value
Structured Log file size	532 MB
Number of sessions	3046
Number of visits	32578
Number of visits (length >-=2 pages)	15246

We have examined methods by extracting the frequent patterns with very low support from the dataset described before. The support value is varied from 0.01% to 0.001% and we have measured algorithm's response time as presented in 7.4. With this, we perceive that the proposed partitioning method turns as a complement for the traditional pattern discovery methods. For 0.02% to 0.06% very low support values, TANASA and WAP-mine are unable to extract any patterns. For the support value below 0.02%, the execution time for partitioning model grows exponentially.

5 Conclusion and Future Work

In this research paper, we offer a sophisticated method for extracting of the all users' behavior of a Web site. Our methodology has the distinguished feature to divide the log file recursively in order to discover the users' behavior and to characterize them as clusters (analogous behaviors are grouped into a cluster). For this perseverance, we have to offer a detailed clustering method, which is devoted to sequential patterns. The key benefit of our proposed method is to study the Web Usage Mining with minimal support as a composite problem that can be solved by succeeding partitions.

The problem therefore, moves from one single open problem to n number of problems. We can resolve and the problem that has to be recursively partitioned.(we can solve a problem, by the application of partitioning method recursively) By furthering in this approach, we could establish that the border between the data quantity and quality of results can occasionally be pushed vertebral by extracting user behaviors with a minimal representativeness.

References

[1] Benedek, A., Trousse, B.: Adaptation of Self-Organizing Maps for CBR case indexing. In: 27th Annual Conference of the Gesellschaft fur Klassifikation, Cottbus, Germany (March 2003)

[2] Fayad, U.M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R. (eds.): Advances in Knowledge Discovery and Data Mining. AAAI Press, Menlo Park (1996)

[3] Giacometti, A.: Modèles hybrides de l'expertise, novembre, PhD Thesis, ENST Paris (1992) (in French)

[4] Cooley, R., Mobasher, B., Srivastava, J.: Data preparation for mining world wide web browsing patterns. Knowledge and Information Systems 1(1), 5–32 (1999)

[5] Jaczynski, M.: Modèle et plate-forme à objets pour l'indexation des cas par situation comportementales: application à l'assistance à la navigation sur le web, décembre, PhD thesis, Université de Nice Sophia-Antipolis (1998) (in French)

[6] Malek, M.: Un modèle hybride de mémoire pour le raisonnementà partir de cas, PhD thesis, Universitẽ Joseph Fourrier (Octobre 1996) (in French)

[7] Masseglia, F., Poncelet, P., Cicchetti, R.: An efficient algorithm for web usage mining. Networking and Information Systems Journal (NIS) (April 2000)

[8] Srikant, R., Agrawal, R.: Mining Sequential Patterns: Generalizations and Performance Improvements. In: Apers, P.M.G., Bouzeghoub, M., Gardarin, G. (eds.) EDBT 1996. LNCS, vol. 1057, pp. 3–17. Springer, Heidelberg (1996)

[9] Tanasa, D., Trousse, B.: Web access pattern discovery and analysis based on page classification and on indexing sessions with a generalised suffix tree. In: Proceedings of the 3rd International Workshop on Symbolic and Numeric Algorithms for Scientific Computing, Timisoara, Romania, pp. 62–72 (October 2001)

[10] W3C. httpd- log files (1995), http://www.w3.org/Daemon/User/Config/Logging.html

[11] Masseglia, F., Cathala, F., Poncelet, P.: The PSP Approach for Mining Sequential Patterns. In: Żytkow, J.M. (ed.) PKDD 1998. LNCS, vol. 1510, pp. 176–184. Springer, Heidelberg (1998)