International Journal of Research in Advent Technology, Vol.7, No.5, May 2019
E-ISSN: 2321-9637
Available online at www.ijrat.org

A Hybrid Approach to Discover Most Frequently Visited WebPages

N.Kurinjivendhan[1], Dr.K.Thangadurai[2]
Ph.D Research Scholar (Full Time) [1], H.O.D and Assistant Professor[2],
P.G. and Research Department of Computer Science,
Email: vendhancs1489@gmail.com[1], ktramprasad04@gmail.com[2]

Abstract - Extracting exciting data and information related to server logs is an interesting area and the web usage mining caters to the need of the website positioning and marketing strategies to the site owners. The web log server creates log files regarding information about the page, IP address of the user, name of the browser, and OS used and time/date stamp regarding browsing patterns and this raw data is mined to extract useful information using web usage mining. The primary objective of this paper is to find the frequently visited pages in a website in tandem using the server log files by combining clustering and sequential item mining techniques. This sequential item generator algorithm generates candidates using Apriori like approach and clustering technique discovers the user navigational behavior using the generated candidates effectively by employing equality distance. The proposed hybrid approach is compared with existing algorithms and a detailed comparison is carried out to prove the efficiency of the proposed algorithm.

Index terms-Hybrid, Frequent, Strict Clustering, Web Mining

1. INTRODUCTION

Drifting on an urbane innovative world, the business house's methodologies and advancements have changed drastically in the ongoing years since the significance of information in their business assumes a crucial task in the majority of their business exercises. Today PC has turned into a vital part of human life and the huge volume of information accessible over the globe gives some assistance to find helpful significant data to upgrade the business in different exercises and enhance the client service to a more prominent degree as for speed, exactness and anticipating some valuable data/information for the client with unmatched accuracy.

Web mining is the application of data mining to the web data and traces user's visiting behaviors and unearths the hidden data and interests using user patterns. Since this area is applicable in e-commerce and Web analytics directly, web mining has become one of the important areas in computer science. Web Usage Mining uses mining methods in log data to extract the behavior of users which is used in various applications like personalized services, adaptive web sites, customer profiling, creating attractive web sites.

Generally Web Page is considered as a nugget of information with a colossal volume of Web assets interconnected and networked. The web resource is habitually known by Uniform Resource Identifier (URI). A URI depiction pronounces a web resource as "Identifiable thing or object". The perception of Web Resource is the ultimate milieu in the web standards. That is considered as the foremost milieu which signifies the web and ensures that it is accessible to the consumers. Usually the web resources are traced by their Uniform Resource Locator (URL) primarily and basically assists initially to discourse the credentials and archives depicted in the webpages. This idea of URL has evolved considerably to include "entity" or "object" was perceived in the internet.

2. TYPICAL WEB LOG DATA

The web log file consists of many fields like IP address or hostname, User Agent, Referring URL, Method, Protocol, Path, Agent, Date, and Time. The web log file is preprocessed in such a way that it is ready to be used in the algorithm to fetch useful patterns. The usual web log file is shown in table 1 and this web log file is transformed into algorithmic log file as shown in table 2. The data is converted into records based on the IP address to identify the users browsing or navigational pattern. This preprocessed log file is fed as input to the proposed algorithm to test its workability.

3. SAMPLE SERVER LOG FILE

IP	Method	Protocol	Page	Agent	OS	Date	Time
192.125.86.3	GET	HTTP1.1	Page 1	chrome	Win7	15.12.18	22.10.13

192.125.86.3	GET	HTTP1.1	Page 3	chrome	Win7	15.12.18	22.11.29
192.125.86.3	GET	HTTP1.1	Page 9	chrome	Win7	15.12.18	22.17.43
192.125.86.3	GET	HTTP1.1	Page 5	Chrome	Win7	15.12.18	22.20.27
199.82.45.15	GET	HTTP1.1	Page 1	IE	Win7	15.12.18	22.17.04
199.82.45.15	GET	HTTP1.1	Page 3	IE	Win7	15.12.18	22.19.13

Table 1: Web server Log File

IP	Pages Visited
192.125.86.3	1,3,5,9
122.87.23.61	1,3,4
191.12.58.23	1,6,3,5,15,17
192.66.02.19	1,3,5,15,17
201.33.12.61	1,3,4,3,17
126.86.76.61	1,3,5,15

Table 2: Normalized web log file

4. SCOPE OF THE PAPER

This paper mainly focuses on the a new technique to find the browsing patterns or the navigational behavior of the users after mining the content of the server log files using a hybrid technique. The proposed algorithm uses sequential frequent itemset mining technique and clustering to unearth the patterns in the web log server data of websites. The most important objective of this paper is to find the hit pages from the web server data instead of just finding the individual hit pages as this paper focuses on finding the hit pages visited in tandem by the users.

5. SEQUENTIAL PATTERN APPROACH

Consider I = {I_1, I_2, … , I_n} be a set of 'n' distinct attributes, T be transaction that contains a set of items such that T □ I, Ds be a database with different transaction records T. An association rule is an implication in the form of A □ B, where A, B□ I are sets of items called item sets, and A n B = ø. A is called antecedent while B is called consequent.

Assumption:

Sequential Web pages symbolizes that the antecedent Web page should be navigated or browsed before the descendant Web page.

This assumption is very crucial due to the fact that Web pages archived in the server log files are sequential in nature and the order of the viewed Web pages in a website is decisive in the prediction process of discovering the navigational patterns of the users [4][5].

Definition:

Itemsets discovered from the raw data without pruning the unpromising items is called candidates.

The proposed approach is to discover the navigational paths visited frequently by the users. To accomplish this, a new algorithm named "Hybrid Algorithm to find Navigational Behavior of the User (HANBU)" is proposed. The first step in the proposed approach is to clean the noisy data present in the weblog dataset to ensure accurate results by removing the noise. Next step is to generate sequential candidates to find the frequently visited pages in tandem by employing clustering. Next step is to cluster the pages to discover the frequently visited pages of the users. To accomplish these steps separate procedures are proposed and the working concept behind these procedures is enumerated with examples.

6. NOISE REMOVAL

This procedure helps to remove the false hits present in the weblog dataset as many of the pages will be repeated in succession in the transactional row of the dataset. This noise has to be removed to provide accurate results in the final stages.

PROCEDURE RemoveNoise(Dataset Ð)

INPUT: Sequential Dataset Ð
OUTPUT: Noiseless Dataset Ð

1. Find the total Transactional Rows Ř ∈ Ð
2. For all Row Ř ∈ Ð do
3. Find the Total Elements Ī ∈ Row Ř
4. For all Elements Ī ∈ Ř do
5. CHECK IF (Elements (Ī) = Elements (Ī +1)) then

```
    6. Remove Elements at (Ī +1)
    7. End IF
    8. End FOR
    9. Return Đ
END PROCEDURE
```

Figure 1: Pseudo code of RemoveNoise

Let us consider a transactional row T which contains 8 items {I1, I2, I3, I4, I5, I6, I7, I8} and the values corresponding to these items are {3, 2, 2, 2, 5, 5, 6, 8}. Here I2 is data and I3, I4 are noises which has to be cleaned before the candidate generation. The RemoveNoise procedure checks the I^{th} item with the $(I+1)^{th}$ item and if found to be equal, the (I+1) item will be removed from the transactional row. The resultant cleaned transactional row T= {3, 2, 5, 6, 8}

7. CANDIDATE GENERATION

PROCEDURE CreateRawCandidates (Dataset Đ)

Input: Dataset Đ = {T_1, T_2, T_3…… T_n }
Output: Sequential Itemsets SeqCand

1. Scan Dataset Đ
2. For all TransactionRow Ř ∈ Đ do
3. For all Elements Ī ∈ Ř do
4. Compute Elements count
5. While [SeqCand count ≤ Elements count] do
6. Combine Elements[Ī] ∪ Elements[Ī + 1] => SeqCand
7. End While
8. End For
9. End For
10. Return SeqCand

END PROCEDURE

Figure 2: Pseudo code of CreateRawCandidates

The candidates are created for each and every row. Let us consider the first row in the table 2,
T1= { 1, 3, 5, 9} Itemsets = { [1,3], [3,5], [5,9], [1,3,5], [3,5,9], [1,3,5,9]}

Table 3: Raw Candidate generated

ID	RAW CANDIDATES
T1	[1,3], [3,5], [5,9], [1,3,5], [3,5,9], [1,3,5,9]
T2	[1,3], [3,4], [1,3,4]
T3	[1,6], [6,3], [3,5], [5,15], [15,17], [1,6,3], [6,3,5], [3,5,15], [5,15,17], [1,6,3,5], [6,3,5,15], [3,5,15,17], [1,6,3,5,15], [;6,3,5,15,17], [1,6,3,5,15,17]
T4	[1,3], [3,5], [5,15], [15,17], [1,3,5], [3,5,15], [5,15,17], [1,3,5,15], [3,5,15,17], [1,3,5,15,17]
T5	[1,3], [3,4], [4,3], [3,17], [1,3,4], [3,4,3], [4,3,17], [1,3,4,3], [3,4,3,17], [1,3,4,3,17]
T6	[1,3], [3,5], [5,15], [1,3,5], [3,5,15], [1,3,5,15]

8. STRICT CLUSTERING

The process of clustering employed here is strict, (i.e.) instead of the calculating the distance between the elements and grouping with respect to the nearest distance, absolute distance (distance equality) is calculated to fetch accurate grouping of the frequently visited pages. The first step to accomplish this, the raw data is segregated according to the number of elements present in the candidates and the first procedure SegregateData initially groups the raw data with respect to the 2-items, 3-items and n-items present in the raw candidates.

PROCEDURE SegregateData (Candidates C)

Input: Candidates C
Output: Segregated Candidates

1. Scan Raw Candidates C
2. For all Row Ř ∈ C do
3. For all Elements Ī ∈ Ř do
4. Compute Elements count
5. Group candidates according to the element count→ RES
6. End For
7. End For
8. Return RES

END PROCEDURE

Figure 3: Pseudo code of SegregateData

The segregated data obtained from the procedure is clustered to discover the frequently visited pages in tandem. The segregated candidates make it easier for the strict clustering or grouping process to be carried out to find the behavior of the user in certain websites. The candidate values which does not fit in any cluster is considered as outlier or low hit web pages which is also found as the admin will restructure the web design and make the pages with low number of audiences visible well to everyone to increase the hits. The procedure to cluster the segregated data is shown in the figure 4.

PROCEDURE StrictClustering (SegregatedCandidates C)

Input: SegregatedCandidates C
Output: Cluster

1. Scan C to find the number of segregation
2. For each segregation in C do
3. While [Row ≠ ϕ]
4. For all Elements Ī ∈ Row do
5. Compute Equality among all the element w.r.t other Row
6. If [Equality = TRUE]
7. Add to the corresponding Cluster
8. Else
9. Form New Cluster
10. End if
11. End For
12. End While
13. End For
14. Return Clusters

END PROCEDURE

Figure 4: Pseudo code of StrictClustering

9. EXPERIMANTAL EVALUATION

Table 4: dataset used

Dataset Name	Sequence count	Distinct items	Average Length	Data Type
MSNBC	989818	17	5.7	Click stream
FIFA	20450	2990	34.74	Click stream
BMS	59601	497	2.51	Click stream
Snake	163	20	60	Protein

The proposed algorithms were implemented in java platform on a personal computer with 2.66GHz Intel Pentium I3 processor, 2GB RAM running on windows 7 ultimate. The evaluations were performed on benchmarked datasets shown in the table 4.

The proposed algorithm is executed on the benchmarked datasets and compared with state of the art algorithms to find the efficiency of the proposed algorithm and the accuracy, speed and the memory consumptions are the important metrics used to gauge the performance of the proposed algorithm.

The algorithms compared are K-Medoid algorithm is an adaptation of K-Means algorithm. Rather than calculating the mean of data points in each cluster, medoid is chosen for each cluster during iterations.

[Shelokar et al.] [8]described an ant colony optimization methodology for data clustering (ACOC). It mainly relies on pheromone trails to guide ants to group data points according to their similarity and on a local search that randomly tries to improve the best iteration solution before updating pheromone trails.

[Kumar et al.] [9] developed a modified harmony search based clustering (MHSC) technique. Here cluster center based encoding scheme is used. Each harmony vector contains K cluster centers, which are initialized to K randomly chosen data points from the given dataset.

The accuracy levels is calculated by utilizing various distance schemes like, Euclidean, Canberra and Bray-Curtis distances along with the distance equality method and the accuracy of the proposed with respect to the distances are shown in the table 5

.

FIFA dataset				
Distance	**Algorithms**			
	KMD	**AOCC**	**MHSC**	**Proposed**
Eucl	**0.836**	0.751	**0.825**	0.805
Cann	0.786	0.789	0.819	0.821
Bray	0.728	**0.835**	0.732	**0.889**
Equality	0.778	0.812	0.766	**0.932**
BMS dataset				
Distance	**Algorithms**			
	KMD	**AOCC**	**MHSC**	**Proposed**
Eucl	0.619	**0.881**	0.817	0.739
Cann	0.756	0.725	**0.838**	0.821
Bray	**0.791**	0.718	0.718	**0.913**
Equality	0.771	0.667	0.726	**0.955**
MSNBC dataset				
Distance	**Algorithms**			
	KMD	**AOCC**	**MHSC**	**Proposed**
Eucl	**0.867**	0.737	**0.815**	0.883
Cann	0.769	**0.829**	0.779	**0.891**
Bray	0.746	0.765	0.711	0.819
Equality	0.661	0.676	0.701	**0.966**

Table 5: Calculated Accuracy for the algorithms

From the table 5, it is quite evident that the proposed algorithm performed extremely well when the equality of the distances is calculated and for some datasets, the proposed performed well by applying existing distances.

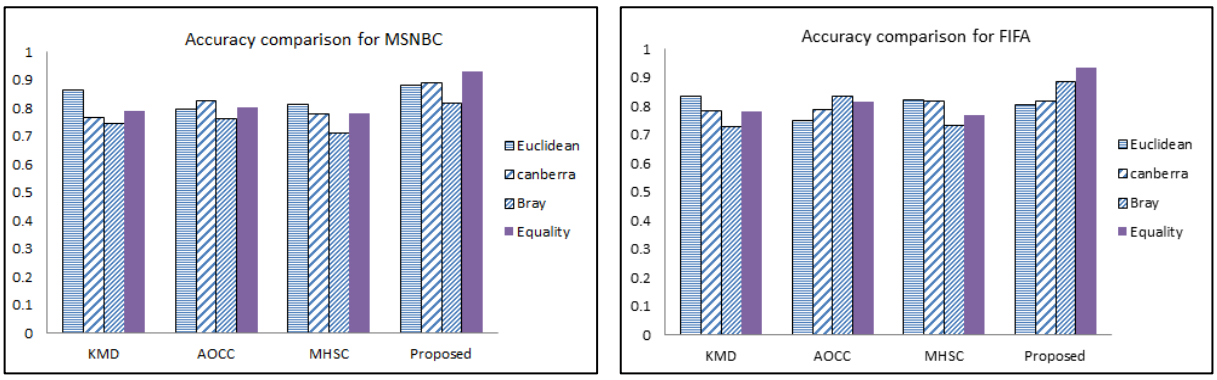

Figure 5: Accuracy comparison

From the table 5 and figure 5 it is quite evident that the proposed hybrid method with equality distance outperformed all the existing algorithms with respect to accuracy. Next to the proposed equality distance measure, the Bray-Curtis distance performed reasonably well and

almost matched the performance of the proposed equality on certain occasions.

FIFA dataset executed with Equality Distance								
Dataset size	KMD		AOCC		MHSC		Proposed	
	Time	Memory	Time	Memory	Time	Memory	Time	Memory
50000	512	311	518	316	456	301	141	138
40000	469	260	479	281	320	251	120	120
35000	430	221	421	238	291	207	102	109
20000	326	201	343	217	267	188	90	91
10000	291	138	268	149	231	166	78	82
MSNBC dataset executed with Equality Distance								
Dataset size	KMD		AOCC		MHSC		Proposed	
	Time	Memory	Time	Memory	Time	Memory	Time	Memory
900000	653	519	612	490	578	389	222	156
800000	619	489	582	471	547	366	189	142
700000	582	461	561	443	513	348	170	128
500000	561	428	530	410	483	310	151	104
200000	501	369	486	391	451	282	133	93

Table 6: Memory and time comparison

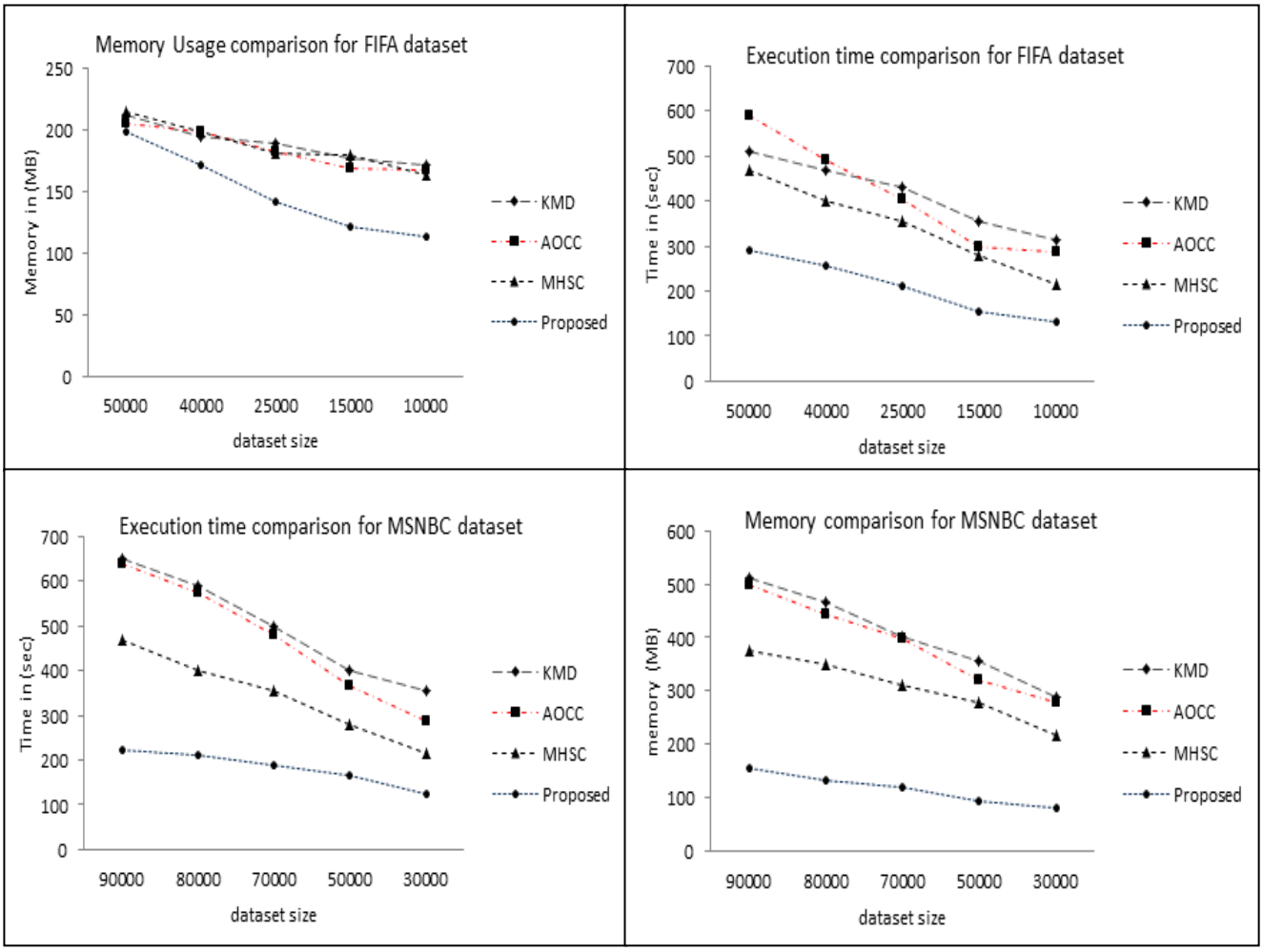

Figure 6: memory and execution time comparison

From the table 6 and figure 6 it is proved that the proposed algorithm performed extremely well by outperforming the other algorithm with respect to time and memory consumption. Since the main purpose of the paper is to find the frequently visited pages in tandem, the candidate generation is necessary and it takes some extra time and extra memory as the pruning is not carried out while the generation.

The proposed hybrid algorithm which combines the features of frequent itemset and clustering to detect the frequently visited web pages is explained very clearly with examples. The proposed algorithm is compared with the state of the art algorithms and the results are plotted in graphs and displayed. Three criteria are used for evaluation namely time, memory and the accuracy and three distance measures are used in the hybrid to test the performance and the proposed algorithm outscored the other three algorithms by a big margin.

10. CONCLUSION

This paper investigates the working and the experimental evaluation of the proposed hybrid algorithm to discover frequently visited pages using strict clustering method and proved to be an efficient approach. Even though the proposed algorithm proves to be efficient in both run time and memory consumption, but there is always room for further research and improvement. Improvements can be made in the pruning strategies to ensure that minimum numbers of candidates are generated with less running time and with less memory consumption.

ACKNOWLEDGEMENT

I would like to thank to the Principal, faculty members of P.G and Research department of computer science and research scholars, Government arts college (Autonomous),Karur,for their encouragement to publish this work.

REFERENCES

[1]. R. Agrawal, and R. Srikant, “Mining sequential patterns”, In ICDE’95, Taipei, Taiwan, Mar. 1995.
[2]. F. Masseglia, F. Cathala, and P. Poncelet, ”The PSP approach for mining sequential patterns”, In PKDD’98, Nantes, France, Sept. 1995.
[3]. J. Han, J. Pei, B. Mortazavi-Asl, Q. Chen, U. Dayal, and M.-C. Hsu, “FreeSpan: Frequent Pattern-Projected Sequential Pattern Mining,” Proc. 2000 ACM SIGKDD International Conference Knowledge Discovery in Databases (KDD ’00), pp. 355-359, Aug. 2000.
[4]. R. Srikant and R. Agrawal, “Mining Sequential Patterns: Generalizations and Performance Improvements”, Research Report RJ 9994, IBM Almaden Research Center, San Jose, California, December 1995.
[5]. Cao,F.Liang, J.Li, D.Bai, “A dissimilarity measure for k-mode clustering algorithm”, knowledge based system,26(1):120-127, 2012
[6]. Lance, G. N. and Williams, W. T., “Computer programs for hierarchical polythetic classification (similarity analyses)”, Computer, 9(1):60–64, 1966.
[7]. Bray, J. R. and Curtis, J. T., “An ordination of the upland forest communities of southern Wisconsin”, Ecological Monographs, 27(4):325–349, 1957.
[8]. Shelokar, P. S., Jayaraman, V. K., and Kulkarni, B. D., “An ant colony approach for clustering”, Analytica Chimica Acta, 509(2):187–195, 2004.
[9]. Kumar .V, Chhabra.J.K, and Kumar.D,” Clustering using modified harmony search algorithm”, International journal of computational intelligence studies,3(2), P.P:113-133, 2014.