Intelligent Analysis of WebLog Mining

Sheetal Malpathak[1], Sarika Zaware[2], Neha Harpale[3], Suffiyana Shiledar[4], Alsaba Shaikh[5]
[1, 2, 3, 4, 5] AISSMSIOIT, Pune, India

Abstract- *In this paper, we are going to investigate a new way to search the most evident co-clusters of users and the corresponding web pages in the web log dataset using frequent super-sequence mining technique. In our project we are going to do mining on the Weblog and then co-cluster the user sessions so as to get a good recommendation and report generation for business analysis. Through experiments it is important to mine the weblog and web content mining. In our research, we are going to investigate web pages structure and find the most evident groups of users and web pages. Nowadays, big data is everywhere. Facing huge amount of web logs, it is not always necessary to group all the users in a web log dataset into different clusters, sometimes, finding out the major dominant user groups and the corresponding web pages is more important. We find interesting result which give helpful information. There are three kinds of mining on weblog data which are web usage mining, web structure mining .We are using structure mining for finding real-time clusters.*

Keywords*:* Super-sequence mining, Pattern mining, Weblog dataset, Heavy paths, Co-Clustering

I. INTRODUCTION

Clustering analysis on web log datasets is important and useful .For example we can find user interest by collaborating them in groups with using web log data and transaction. Clustering analysis is to collect data from dataset and gather into understandable and useful form. The dataset contains variety of instances in relation to features. Thus clusters can be form either on feature or instances .For instances in dataset of papers, the papers can be grouped according to the meaningful words or else, the words can be grouped according to the papers to which they belong. Subsequently, we may also want to combine similar papers and to discover their related words as clusters. The discovery of doing clustering on both instances and features is called co-clustering. The grouping of web users and pages can be done concurrently using sequential data, which can be used to relate user with their user sessions. Many times we do not need to combine all the user sessions into various clusters. Instead, we only need to discover the prominent groups, which is partial clustering.

In this paper, we are doing investigation of finding the necessary partial co-clustering which is most evident co-clusters on datasets. The reason of using partial co-cluster, the web users and web pages is we want detect evident co-cluster in it. Empirical work focused on how to cluster all elements in a domain or to acquire each element that could be categorized to belonging group. In the big data world, to analyze each and every detail in the dataset is impractical, while finding the most useful and evident information is more complicated.

II.RELATED WORK

Clustering is important process to extract and analyze useful information. Previously clustering methods used such as K-means and Pattern-Oriented Partial Clustering was grouping the sequence of web pages 123 and 321 in same cluster. But there might be the case that these sequences actually belong to different clusters. Also traditional clusters used to group the users but not webpages related to the users simultaneously. Suppose one sequence of web pages is 1234 and other sequence is 678.Suppose the user who visit 4 is also visiting 5 but others users 123 are not visiting. In the previous situation, it is possible that web page 5 is not reachable to users of pages 123,so they are directly visiting next web page. We can merge the sequence 1234 and 678 by advertising the web page 5 to users who visit web page 4.

.

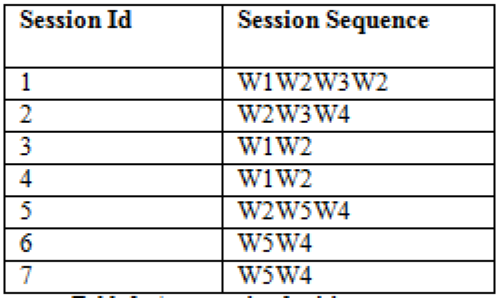
III.EXAMPLE

Table 1 shows a simple example of a web log history, the first column is the user's session ID, we can think of them each as a different user. The second column shows the sequences of web pages they visited. If we would like to cluster these users and web pages into groups, we can see that user 1, 2 and 3 can be grouped together with web page W1, W2, W3 and W4 since 1 and 2 both visited web page W2 and W3 while 1 and 3 both visited W1 and W2. We can also group user 4 to them. We can see that user 4 is more close to 5 since they have W5W4W6 as common pages. At last we have user 6 is not belonging to any of the previous two groups. The two dominant co- clusters are [(1,2,3),(W1,W2,W3,W4)] and [(4,5),(W2,W5,W4,W6)]. If we look closer, we found that D does not contribute as much as the other web pages in the first group, so did B in the second. So we may update the two groups as [(1,2,3),(W1,W2,W3)] and [(4,5),(W5,W4,W6)].

IV. PPROPOSED SYSTEM

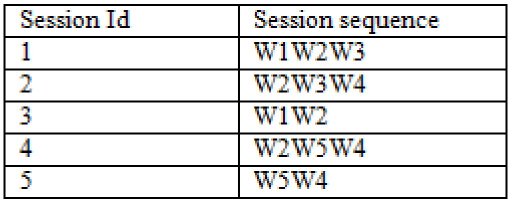
A sequential dataset is a collection of sequence(s) of ordered elements or events.[6] As an example, let us consider web log sessions in Table 1. Each web session is an ordered sequence of web clicks, i.e., two consecutive pages W1W2 means that a user first visited page W1 and then page W2. In this example, traditional FPM algorithms (e.g., Apriori, downward closure) would identify W1W2 and W5W4 as the most frequent sub-sequence patterns as they appear three times in the dataset. Finding such frequent sub-sequences can be helpful for analyzing the most common structures and improving the performance. For example, by analyzing the traversal patterns in a web server's log, one can gather important information such as the most popular pages which are likely to be visited together.

Table 1: An example of web log sequences.



Session Id	Session Sequence
1	W1W2W3W2
2	W2W3W4
3	W1W2
4	W1W2
5	W2W5W4
6	W5W4
7	W5W4

Table 2: An example of sequential dataset consisting of web log sessions.



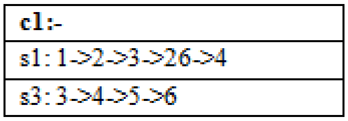
Session Id	Session sequence
1	W1W2W3
2	W2W3W4
3	W1W2
4	W2W5W4
5	W5W4

Web log dataset denoted by S={S1, S2, ..., SN} where Si = {v1, v2, ...,vli} is a sequence of ordered web pages the users visited in each session. W = {v1, v2, ...,vn} be the complete set of all the web pages

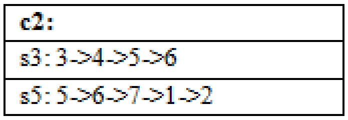
Behavioral segmentation example-

s1: 1->2->3->26->4
s2: 16->17->18
s3: 3->4->5->6
s4: 12->13->14->15
s5: 5->6->7->1->2
s6: 17->24->18 ->19

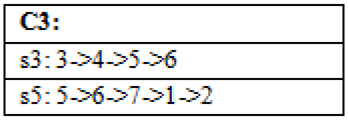
Step 1. First, we create the elementary clusters c1, c2, c3, c4 that contain overlapping sequences.



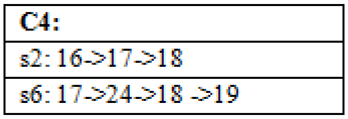
c1:-
s1: 1->2->3->26->4
s3: 3->4->5->6



c2:
s3: 3->4->5->6
s5: 5->6->7->1->2

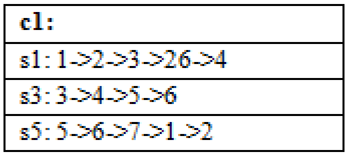


C3:
s3: 3->4->5->6
s5: 5->6->7->1->2

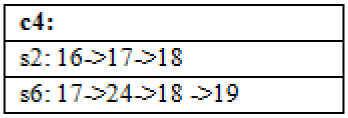


C4:
s2: 16->17->18
s6: 17->24->18 ->19

Step 2. We can merge the clusters c1, c2, and c3 since the sequences they contain overlap between the clusters



c1:
s1: 1->2->3->26->4
s3: 3->4->5->6
s5: 5->6->7->1->2



c4:
s2: 16->17->18
s6: 17->24->18 ->19

V. EXPERIMENTS AND RESULTS

We use an real-time web log dataset consisting of multiple sequences (sessions) and analyze the partial co-clusters in it. The dataset is one month's worth of all HTTP requests to our Online Shopping Portal server. We processed the original file into a file in the sequences format, where each sequence is a session of user's html page requests with user name and their shopping details. We eliminated the irrelevant items by checking the suffix of the products. For instance, all product entries with suffixes such as, product _name(ing),(ed),date of shopping are removed.

There are 1022 sessions and more than 60 products sale in the processed dataset. All of the experiments are run on Intel Pentium CPU with 4Gb memory. We ran the super-sequence searching algorithm in to search super-sequences. We first formed the clusters of users based on product similarity and then we co-clustered by considering super-sequences not only based on product but also on users. The plus point of this algorithm is ,it automatically makes the decision of how many co-clusters should be formed based on real-time dataset.

VI. CONCLUSION

We have solved the problem of clustering web access sequences with more efficiency. Due to the limitations of the existing clustering methods mostly large amount of time for clustering and limited grouping probabilities, we are providing a modified algorithm, which uses frequent patterns to generate both clustering model and cluster contents. The algorithm iteratively merges smaller, similar clusters until the requested number of clusters is reached or you can get more general clusters. An important feature of the algorithm is that it does not only divide the web users into checking if it contains patterns from any of the clusters descriptions. If the new user access path contains patterns from different clusters, then it belongs to many clusters with different membership probabilities .Thus we are not only clustering members on similarity but also on their access pattern. Clusters also delivers a classification model that can be used to classify future web users. Since the model is formed by a set of frequent patterns to be contained, the classification of a new web user access path simply consists in checking to which cluster they belong.

ACKNOWLEDGMENT

We are pleased to acknowledge Prof.Sarika Zaware and Prof.Suresh Limkar for their invaluable guidance during the course of this project work.

We are also thankful to other staff members who co-operated and helped with some issues.

REFERENCES

[1] R. Agrawal and R. Srikant. Mining sequential patterns. In Data Engineering, 1995. Proceedings of the Eleventh Inter- national Conference on, pages 3–14. IEEE, 1995.

[2] Kale Sarika Prakash,P.M.J Prathap,"A Survey on Iceberg Query Evaluation Stratigies",International Journal of Modern Trends in Engineering and Research ,e-ISSN NO.2349-9745,July 2015.

[3] A. Banerjee, S. Basu, and S. Merugu. Multi-way clustering on relation graphs. In SDM, volume 7, pages 225–334, 2007.

[4] A. Banerjee, I. Dhillon, J. Ghosh, S. Merugu, and D. S. Modha. A generalized maximum entropy approach to breg- man co-clustering and matrix approximation. In Proceed- ings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 509–514. ACM, 2004.

[5] R. Bekkerman, R. El-Yaniv, and A. McCallum. Multi-way distributional clustering via pairwise interactions. In Pro- ceedings of the 22nd international conference on Machine learning, pages 41–48. ACM, 2005.

[6] C. Berge. Graphs and hypergraphs, volume 6. Elsevier, 1976.

[7] Y. Cheng and G. M. Church. Biclustering of expression data. In Ismb, volume 8, pages 93–103, 2000.

[8] I. S. Dhillon, S. Mallela, and D. S. Modha. Information-theoretic co-clustering. In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 89–98. ACM, 2003.

[9] C. H. Ding, X. He, and H. D. Simon. On the equivalence of nonnegative matrix factorization and spectral clustering. In SDM, volume 5, pages 606–610, 2005.

[10] J. Han, J. Pei, B. Mortazavi-Asl, Q. Chen, U. Dayal, and M.-C. Hsu. han2000freespan: frequent pattern-projected se- quential pattern mining. In Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 355–359. ACM, 2000.

[11] N. Slonim and N. Tishby. Document clustering using word clusters via the information bottleneck method. In Proceed- ings of the 23rd annual international ACM SIGIR confer- ence on Research and development in information retrieval, pages 208–215. ACM, 2000.

[12] J. A. Hartigan. Direct clustering of a data matrix. Journal of the american statistical association, 67(337):123–129, 1972.

[13] T. Morzy, M. Wojciechowski, and M. Zakrzewicz. Web users clustering. Citeseer.

[14] J. Han, J. Pei, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M. Hsu. Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In Pro- ceedings of the 17th International Conference on Data En- gineering, pages 215–224, 2001.