An Approach To Find Frequent Pattern From Logs Using Modified Apriori Algorithm

Priyanka Makkar[#1]
[#]*Assistant Professor & Computer Engineering Department & Pune University, Pune, India*

Abstract *- Web Usage Mining is an application of Data Mining to generates pattern from the logs which is created when user interacts with websites. Web usage mining is about processing or analysing clickstream data. We can analyse these logs and can find out the user interest and can recommend pages based on user interest. In this paper we have applied Modified Apriori Algorithm on web logs to find patterns. Logs are first pre-processed and then modified Apriori is applied to find interesting pattern which can be used to predict the next page visit of user. The Modified Apriori Algorithm is fast as it requires less scan of database than the basic Apriori algorithm. We have used the dummy dataset and find the frequent patterns using Modified Apriori Algorithm. Modified Apriori is faster as compared to basic Apriori Algorithm as it requires less database scan and therefore suitable for many real time applications.*

Keywords - *Pre-processing weblogs, web usage Mining, Modified Apriori algorithm, faster Apriori.*

I. INTRODUCTION

In this era of information, we have been collecting massive amount of data than we can handle, and therefore there is a need to properly summarize and analyse data and discover useful patterns from it. Data Mining is the process of analysing data from different perspectives and summarizing it into useful information that can be used to increase revenue of the service provider [1]. Web Mining is the application of data mining techniques to discover patterns from the World Wide Web. Through web Mining, we are able to gain a better understanding of both the web and web-user preferences, a knowledge that is crucial for mass customization [2].

Web Mining are of three types:
1) Web structure mining
2)Web content mining
3) Web usage mining

Web structure Mining**:** Web Structure Mining is about extracting knowledge from the hyperlinks. Significant web pages can be identified, also users that have common interests, i.e. using the identical clusters of linked pages can be identified [5].
Web Content Mining: It gives detailed account about the finding of useful information from Web documents. Basically, Web content consists of several types of data like metadata, text, audio, hyperlinks, image as well as videos. Mining from this content is web content Mining [4].
Web Usage Mining: It provides the information that describes the usage patterns of Web pages from the logs, logs consist of details such as IP addresses, page references, date and time of accesses, other information depending on the log format, free texts, HTML Files, XML Files, Dynamic Content, and Multimedia Files [3].

Web log Mining includes three main stages:

Data Pre-Processing: In Data pre-processing, logs are scanned to checked for any irrelevant data, if there are any irrelevant data they are removed or path completion is done which makes log more relevant. After this user identification is done. In this we identify different users and their access patterns are stored. After that session identification is done and a transaction table is created for each user session [6][7].

Pattern Discovery: Several patterns are discovered using various algorithms like Association rules, Sequential Pattern [8].

Pattern Analysis: After pattern discovery various pattern are analysed to find the behaviour of user so that we can predict user next click which can improve performance.

There are various limitation of web Usage Mining:

1)As the data is huge it becomes challenge to mine web.
2)Difficult to handle irregular, unstructured data.
3)Managing hardware and software for large processing is difficult [9].

In the paper we are showing using a simulated example how pattern in the logs can be found efficiently using modified Apriori algorithm. Our example shows that less scanning to database is required to predict user next page access than the basic Apriori algorithm.

Layout of the paper is: Section II discusses Related Works, Section III is having proposed Architecture with its description, Section IV Explain the modified Apriori algorithm with Simulated Example, Section V Conclusion, followed by References.

.

II. RELATED WORK

The analyses of the information from the weblogs was introduced by [14].

Web usage mining (WUM) is the process of the retrieving useful information/knowledge from the server logs. Server logs contain irrelevant data which does not contribute towards extracting useful information, so these log files require pre-processing. Then from the pre-processed files different patterns are required to be discovered in order to comprehend the behaviour of the users. The found patterns required to be analysed to form useful knowledge. The knowledge obtained from web usage mining can be used to enhance web design, introduce personalization service and facilitate more effective browsing. The various applications of web usage mining are: robots' detection and removal, extracting user profiles So the basic introduction was covered in this paper [12].

Identifying the usage patterns of users is very vital in use the information available in the World Wide Web. This paper works on the future trends of web mining and trying to give a brief idea regarding web mining concerned with its techniques, tools and applications [13].

This paper has attempted to provide an up-to-date survey of the rapidly growing area of Web Usage mining. With the growth of Web-based applications, specifically electronic commerce, there is significant interest in analysing Web us- age data to better understand Web usage, and apply the knowledge to better serve users. This has led to a number of commercial offerings for doing such analysis [11].

In this paper pre-processing of web logs was explained First step in any web usage mining task is pre-processing. K. R. Suneetha and Dr. R. Krishnamoorthy give an insight upon how important it is to properly pre-process [15].

Apriori algorithm [16], proposed by Rakesh Agrawal and Ramakrishnan Srikant is an influential data mining algorithm. It is an algorithm which can solve the problem of web usage mining. It generates a list of most frequent web pages visited. Being a very slow algorithm is the biggest disadvantage for the service providers. Due to fast changing contents of database one needs an algorithm which is real time.

This paper proposes, the feature-matrices (FM) model, to discover and interpret users' access patterns. With FM, various spatial and temporal features of usage data can be captured with flexible precision so that we can trade off accuracy for scalability based on the specific application requirements. Moreover, complexity of the FM model allows real-time and adaptive access pattern discovery from usage data [10].

Another way to improve Apriori is to use most suitable data structure such as frequent pattern tree. Han et. al. [17], in introduced an algorithm known as FP-Tree algorithm for frequent pattern mining. It is another milestone in the development of association rule mining and avoids the candidate generation process with less passes over the database.

An improved version of original Apriori- All algorithm is developed for sequence mining in [15]. It adds the property of the userID during every step of generation of candidate set and every step of scanning the database to decide about whether an item in the candidate set should be used to produce next candidate set. The algorithm reduces the size of candidate set in order to reduce the number of database scanning.

To overcome this difficulty modified version of Apriori Algorithm is proposed to generate frequent web pages. After first pass, list of all frequent web pages is determined. After second pass a correlation is found between frequent web pages from the database. Now these correlations are put in form of a graph. The graph is mined for finding patterns instead of the database [1].

III. PROPOSED ARCHITECTURE

The propose architecture shown in Fig 1 represent when the user request any page then these requests are stored in web logs. These Web logs are then pre-processed. Processing includes cleaning of logs; all the irrelevant or missing data is removed in cleaning step. Next step is user identification. All the user in logs are identified and then session identification is done. a transaction table is created for each user session Once we get pre-processed logs, we apply modified Apriori algorithm. Modified Apriori algorithm is faster than basic Apriori Algorithm as it requires less scanning to database and therefore more efficient and suitable for real time applications. This Algorithm will provide us with the recommendation rules. It analyses user access pattern and provide recommendation of next click to be done by user. Service providers can recommend user next click which can improve performance. Modified Apriori algorithm scan the dataset and then generates table and graphs. From these tables and graph, Patterns are found. Frequent patterns are the patterns having support value greater than or equal to threshold and then we calculate the confidence and provide rules to the service provider.

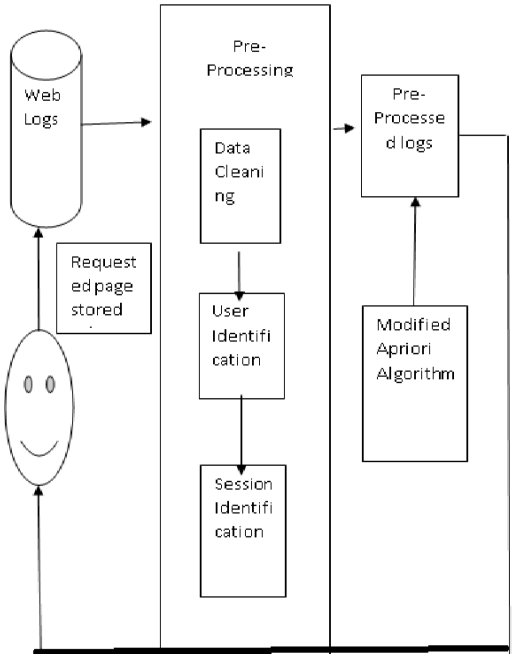

Fig 1: Architecture of Proposed work

IV. SIMULATED EXAMPLE

We have used the following example to find frequent patterns in weblogs using modified Apriori Algorithm. we have set the confidence to 50% and support to 20%, support will be 0.2*9=1.8. The following table will consist of ids and the web pages represented as A, B, C. accessed by the user

ID	Webpages Accessed
1	A, B, D
2	B, E
3	B, C
4	A, B, E
5	A, C
6	B, C
7	A, C
8	A, B, C, D
9	A, B, C

Table 1: Sample Logs

First step is to find all pages which are accessed more than the support value i.e. 1.8 from the sample dummy logs. The pages which are having accessed count more that support I are A, B, C, D, E which are having 6,7,6,2,2 counts respectively.
Next step is to initializes a 2D matrix shown below with zeros. In below shown 2D array, X and Y axis will have pages which are selected in first step.

	A	B	C	D	E
A	0	0	0	0	0
B	0	0	0	0	0
C	0	0	0	0	0
D	0	0	0	0	0
E	0	0	0	0	0

Table 2: Matrix initialized with 0

Next is to create combinations of two item combinations for every transaction. For each combination present in Matrix increment the particular value by one.

	A	B	C	D	E
A	0	4	4	2	1
B	0	0	4	2	2
C	0	0	0	1	0
D	0	0	0	0	0
E	0	0	0	0	0

Table 2: Matrix after applying above steps

The above matrix shows that the access pattern A, B occurs 4 time in dummy dataset. The pattern B, E occurs 2 time in dataset and so on. The values are known as edge weights. These matrices are created using two scans of dataset. Now these matrices will be scanned to generate patterns.
Further is next step two list variable are declared, one is Global 2D list variable "Answer "to store all frequent patterns and the other is the "result" list variable. The algorithm will focus on finding frequent patterns from the graph. In Every iteration one vertex is added from [A, B, C, D, E] and appended in "Result" list, provided Following three conditions are satisfied.
1) The new node found must not be present in the list result list.
2) The Edge weight from the last element in result list is to the new node found must be >= Support Value.
3) At least 1 new vertex should be found satisfying the above two criteria.

Initially result list is having null entries representing it is empty.

NULL	NULL	NULL	NULL

Table 3: Result list

Next is to append "A" to it

A	NULL	NULL	NULL

Table 4: Result list

We have defined a variable count. This is used to keep a track whether any new vertex has been added during the present iteration. If not, even a single new vertex fulfilling the three conditions previously mentioned are encountered count will have value as zero. Next vertex 2 holds each one of the elements from Y coordinates of the graph with the X coordinate as "A". Let in first instance vertex2 (from Y coordinate) hold B. i.e. WebMatrix[A][B]. Edge weight from A → B is four. This is greater than the support value 2. Also, result list contains [A] and there is no B in it. Hence B is accepted and stored in result list.

We will be recursively adding new node to result list. we have declared count variable which will be keeping track whether any new vertex satisfying three conditions is found or not. If not found count will have value 0.Now adding vertex to result list 1st node will be A represent v1,we need to find v2 from matrix satisfying three conditions.so the node is B .i.e. web matrix[A][B] having edge weight 4 which is greater than support value and result list is having [A] and there is no B in it.so B is stored in result list

A	B	NULL	NULL

Table 5: Result list

Now again node 1 will be B now. Also, count is incremented by one so avoid redundancy. Since we increased the value of count by one, we have to now check A->B->other vertex satisfying conditions which has value greater than support->C is found and also C is not in the result list.so pattern till now is A->B->C.

A	B	C	NULL

Table 6: Result list

We need to find new vertex from C that can satisfy conditions. Since no node is found the count for this will remain zero so new node will not be added.so the pattern found is A->B->C. This pattern supports
is greater than equal to 2 when checked in dataset. So, this is accepted frequent pattern.
To find next pattern delete C and scan again from B.

We now find B->D as the count is 2 for this. The next frequent pattern (A->B->D), this combination appears 2 in dataset so it is also accepted.

A	B	D	NULL

Table 7: Result list

Next, we found was B->E which is again having edge Weight 2. The above combination will be
A->B->E which appears only once in dataset so it is rejected as frequent pattern.
Now we have found two high frequency patterns [A->B->C], [A->B->D].

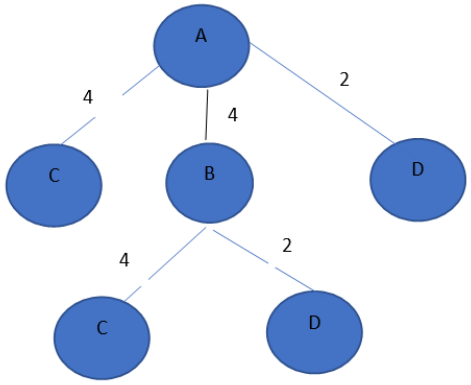

Fig2: Tree structure formed by Modified Apriori Algorithm

This algorithm takes care of the fact that all the pattern which exist in or in subject of found patterns are discarded as they are already present and thus the frequent patterns will be free from redundancy.

To generate rules from the frequent pattern the below formula is used:

$$Confidence\ (X \rightarrow Y) = support\ (X\ U\ Y)\ /\ support(X)$$

This can be understood as the Probability of occurrence of A, given B has already occurred.

$$P\ (A \mid B) = P\ (A \cap B)\ /\ P\ (B)$$

These rules are very useful in Web Usage Mining. Web Servers can predict what the next click of the user can be and restructure themselves accordingly
We have found two frequent sets [A, B, D], [A, B, C]]. We will generate all the subsets of the same. Confidence is calculated as per first equation for calculating confidence for (A, D) → B. A, B and D occur together in 2 transaction Id's. A and D together occur in 2 transactions Id's. Therefore, confidence =

2/2 = 1 or 100% This means that, if a User access page A and D then 100% they will access page B.
List of all the rules found for (A, B, D) along with their confidence are:
1) (A, D) → B (100%)

2) (B, D) → A (100%)
3) (D) → (B, A) (100%)
And similarly, for A, B, C in this no combination of patterns are having confidence 100%.

V. CONCLUSION

In this paper we have applied modified Apriori Algorithm on web logs .The modified algorithm generates the frequents pattern by scanning the database comparatively less no of times than basic Apriori Algorithm which scans the database many number of times .Due to many times scanning the database Basic Apriori Algorithm has more time complexity than modified Apriori Algorithm and therefore making it unrealistic for real time application .The modified algorithm when applied on dataset generates efficient pattern in less space and time and therefore suitable for realistic applications

REFERENCES

[1] Pritish Yuvraj ,Suneetha K. R. ," Modified Apriori Graph Algorithm for Frequent Pattern Mining", 2016 International Conference on Innovations in information Embedded and Communication Systems (ICIIECS'16).

[2] Mobasher B., R. Cooley, J. Srivastava." Automatic personalization based on web usage mining.",Communications of ACM 43 142-151.

[3] Simmi Bagga, " Ethos of Web Usage Mining - A Survey", IJA-ERA)ISSN: 2454-2377 Volume – 2, Issue – 1, May – 2016.

[4] K.Harish Kumar , "A Study on Web Mining Types and Applications ", International Journal of Trend in Research and Development, Volume 3(5), ISSN: 2394-9333 www.ijtrd.com

[5] IOANA MOISIL, "Advanced AI Techniques for Web Mining", Mathematical Methods, Computational Techniques, Non-LinearSystems, Intelligent Systems.

[6] Aggarwal, B. B. D. S., and Shivangi Dhall, "Web mining: Information and pattern discovery on the world wide web.", International Journal of Science, Technology & Management .

[7] Praveen Kumari, "Web Mining - Concept, Classification and Major Research Issues: A Review", Asian J. Adv. Basic Sci.: 2016, 4(2), 41-44 ISSN (Print): 2454 – 7492 ISSN (Online): 2347 – 4114.

[8] Neha Sharma, "A Hand to Hand Taxonomical Survey on Web Mining", International Journal of Computer Applications (0975 – 8887), Volume 60– No.3.

[9] J. Srivastava , P. Desikan , V. Kumar, ―"Web Mining – Concepts, Applications and Research Directions‖", Studies in Fuzziness and Soft Computing, Volume 180, pp. 275–307.

[10] Cyrus Shahabi,Farnoush Banaei-Kashani," Efficient and Anonymous Web-Usage Mining for Web Personalization", https://infolab.usc.edu/DocsDemos/informs.pdf.

[11] Cooley:l: Jaideep Srivastava t, Robert , Mukund Deshpande, Pang-Ning Tan," Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data ",SIGKDD Explorations, Volume 1, Issue 2

[12] Rajinder Singh Rao, Jyoti Arora," A Survey on Methods used in Web Usage Mining", International Research Journal of Engineering and Technology (IRJET)

[13] Sahaj Chavda, Saurabh Jain, Nikunj Panchal, Manisha Valera," Recent Trends and Novel Approaches in Web Usage Mining ",International Research Journal of Engineering and Technology (IRJET)

[14] Agrawal, R., Imielinski, T., and Swami, A. N,"Mining association rules between sets of items in large databases", In Proceedings of the ACM SIGMOD International Conference on Management of Data, 207-216.

[15] K. R. Suneetha and Dr. R. Krishnamoorthi, "Identifying User Behaviour by Analyzing Web Server Access Log File", IJCSNS International Journal of Computer Science and Network Security, VOL..9 No .4, April 2009.

[16] Rakesh Agrawal and Ramakrishnan Srikant, "Fast Algorithms for Mining Association Rules", Proceedings of the 20th VLDB Conference Santiago, Chile, 1994.

[17] Han, J., Jian, Pei., Yiwen, Yin, and Runying, Mao. "Mining Frequent Pattern without Candidate Generation: A FrequentPattern Tree Approach". Journal of Data Mining and Knowledge Discovery, 8, pp.53-87, 2004