A Neoteric Web Recommender System based on Approach of Mining Frequent Sequential Pattern from Customized Web Log Preprocessing

Manisha Valera
GTU,
C.U. Shah College Of Engineering and Technology,

Kirit Rathod
GTU,
C.U. Shah College Of Engineering and Technology,

Uttam Chauhan
GTU,
Vishwakarma Government Engineering College,

ABSTRACT

A real world challenging task of the web master of an organization is to match the needs of user and keep their attention in their web site. So, only option is to capture the intuition of the user and provide them with the recommendation list. Web usage mining is a kind of data mining method that provide intelligent personalized online services such as web recommendations, it is usually necessary to model users' web access behavior. Web usage mining includes three process, namely, preprocessing, pattern discovery and pattern analysis. After the completion of these three phases the user can find the required usage patterns and use this information for the specific needs. The data abstraction is achieved through data preprocessing. The aim of discovering frequent sequential access patterns in Web log data is to obtain information about the navigational behavior of the users. In the proposed system, an efficient sequential pattern mining algorithm is used to identify frequent sequential web access patterns. The access patterns are retrieved from a Graph, which is then used for matching and generating web links for recommendations.

General Terms

Web Mining, Sequential Pattern Mining, Recommendation.

Keywords

Web Usage Mining (WUM), Preprocessing, Pattern Discovery, Pattern Analysis, Weblog, Sequential Patterns.

1. INTRODUCTION

In this world of Information Technology, The Internet has impacted almost every aspect of our society. Since the number of web sites and web pages has grown rapidly, discovering and understanding web users' surfing behaviour are essential for the development of successful web monitoring and recommendation systems. Meanwhile, the substantial increase in the number of websites presents a challenging task for web masters to organize the contents of websites to cater to the need of user's [1].

Web mining can be categorized into three areas of interest based on which part of the web to mine [3]:

1. Web Content Mining
2. Web Structure Mining
3. Web Usage Mining

1.1 Web Content Mining

It deals with discovering important and useful knowledge from web page contents. It contains unstructured information like text, image, audio, and video.

1.2 Web Structure Mining

It deals with discovering and modeling the link structure of the web. Web structure mining aims to generate structural summary about web sites and web pages.

1.3 Web Usage Mining

It is the application of data mining techniques to discover interesting usage patterns from Web data, in order to understand and better serve the needs of Web-based applications.

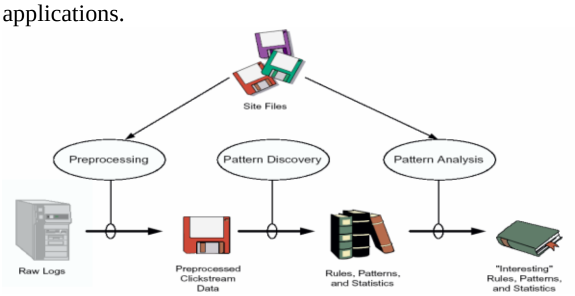


Fig 1: Process Of Web Usage Mining

2. PROPOSED ALGORITHM

Sequential Web page Access pattern mining has been a focused theme in data mining research for over a decade with wide range of applications. The aim of discovering frequent sequential access (usage) patterns in Web log data is to obtain information about the navigational behavior of the users.

This can be used for advertising purposes, for creating dynamic user profiles etc

Table 1. Steps of Old and Modified Algorithm

Old algorithm[11]	Modified algorithm
Create graph from Sessions Pruning (Remove infrequent nodes and edges.) Generate frequent sequential pattern using DFS order by their frequency.	Remove infrequent WebPages. Creating graph from the web access sequences. Pruning (removing infrequent edges.) Generate frequent sequential pattern using DFS order by their frequency. Generate recommendation rule from given frequent sequential patterns order by their support.

The proposed algorithm constructs a Graph to capture the user's web access behavior of a website and then uses the data mining steps in order to find out the Frequent Sequential Web Access Patterns. Then Web recommendation system is designed based on the Frequent Sequential Web Access Patterns. In this approach, firstly we are applying the pre-processing step on Input Web server logs to get the list of accessed web page sequence within different sessions, and then will construct a Web Usage Graph using accessed web page sequence in all sessions. Finally applying mining steps to mine the useful sequential user access patterns. From sequential user access patterns we generate recommendation of web page.

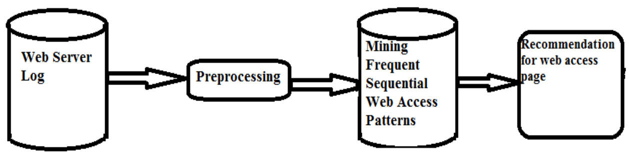


Fig 2: Block diagram of the proposed web recommendation system

The purpose of data preprocessing is to extract useful data from raw web log and then transform these data in to the form necessary for pattern discovery. The input for the proposed web recommendation system is a web server log data and it comprises IP address, access time, HTTP request method used, URL of the referring page and browser name. It is difficult for these web server log data to be directly used to mine the desired sequential pattern mining process [10]. There are three steps in preprocessing of log data: Data cleaning, user identification, session identification. The log entry contains various fields which need to be separate out for the processing. The process of separating field from the single line of the log file is known as field extraction. The server used different characters which work as separators. The most used separator character is or 'space' character.The process of data cleaning is removal of outliers or irrelevant data. The Web Log file is in text format then it is required to convert the file in database format and then clean the file. First, all the fields which are not required are removed and finally we will have the fields like date, time, client ip, URL access, Referrer and Browser used/ Access log files consist of large amounts of HTTP server information. Analyzing, this information is very slow and inefficient without an initial cleaning task. All log entries with file name suffixes such as gif, JPEG, jpeg, GIF, jpg, JPG can be eliminated since they are irrelevant [4].Web robot (WR) (also called spider or bot) is a software tool that periodically a web site to extract its content[6]. To identify web robot requests the data cleaning module removes the records containing "Robots.txt" in the requested resource name (URL). The HTTP status code is then considered in the next process of cleaning by examining the status field of every record in the web access log, the records with status code except 200 are removed because the records with status code 200, gives successful response[7].

Using User Identification, we can identify individual user by using their IP address. If new IP address, there is new user. If IP address is same but browser version or operating system is different then it represents different user. User identification an important issue is how exactly the users have to be distinguished. It depends mainly on the task for the mining process is executed. In certain cases the users are identified only with their IP addresses .

The different sessions belonging to different users should be identified. Web logs can be regarded as a collection of sequences of access events from one user or session in timestamp ascending order. Here we are defining a time interval of 1 hour for each session [11].

Algorithm : Frequent Sequential Pattern Mining

Input: Web_Access_Sequence S – accessed page list of each session.

Min_count=No. Of Sessions * Percentage of support/100

Output: Frequent_Web_Access_Patern – Table of frequent_pattern with their respective frequency, arranged according to length of sequence.

Remove_infrequent _webpages()

```
{
  For (each session Si in web _access_sequence  S)
  {
   For(each webpage Pj in Si)
   {
        If(i=j=1)
        {
              Set page_count(Pj)=1.
        }
        Else
        If (page Pj does not exist in  pages seen so far)
        {
             Set page_count(Pj)=1.
        }
        Else
        {
              page_count(Pj)= Page_count(Pj)+1.
        }
```

```
  }
}
For (each session Si in web_access_sequence  S)
{
        For(each webpage Pj in Si)
        {
                If(page_count(Pj)<min_count)
                {
                        Remove webpage(Pj).
                }
        }
}
}
```

Create_graph (Web_Access_Sequence S)

```
{
For(Each Session Si in Web_Access_Sequence S)
{
For(Each webpage Pj in page sequence of Si)
{
If (i=j=1) {
        create_node(Pj)
 }
Else If (Pj does not exists in all nodes generated   so far)
{
        create_node(Pj)
}
For(Each adjacent pair of webpage Pj in page
sequence)
{
If (i=j=1) {
create_edge(source node(Pj),destination node(Pj+1))
set link_count= 1
}
Else If (an edge with same source, destination does not exists in all edges generated so far) {
create_edge(source node(Pj), destination node(Pj+1))
set link_count= 1
}
Else If (an edge with same source, destination exists in all edges generated so far in Si) {
 link_count = link_count + 1
                        }
                }
        }
}
```

prune_Graph(Web_Usage_graph <N,E>,min_count)

```
{
For (Each edge E in Pruned_Web_Usage_graph G)
{
    If(link_count(E) < min_count)
    remove edge E from Pruned_Web_Usage_graph G
}
}
```

Mining_graph(Pruned_Web_Usage_graph G')

```
{
For (All nodes & edges in Pruned_Web_Usage_graph G')
{
        Traverse the longest path
        set length=number of nodes in longest sequence
while (length>0)
{
Traverse all paths with number of nodes=length,in Pruned_Web_Usage_graph G'
add the visited nodes in the path as frequent_pattern
add frequent_pattern to frequent_pages_list
if(length==1)
{
        set frequency of frequent_pattern=node_count
}
Else
{
set frequency of frequent_pattern=minimum of link_count of visited edges in the frequent_pattern
length=length-1
}
Return (frequent_pattern, frequency, length)
}
}
}
```

Algorithm:Web Page Recommendation Rules Generation

Input: Frequent Sequential Patterns

Output: RR – recommendation rule of a set of ordered access events for S.

1. frequent sequence with minimum support count.

   S = a1a2… an - current access sequence of a user,

2. *MinLength* - minimum length of access sequence,

   MaxLength - maximum length of access sequence

3. Initialize *RR* = ∅.

4. If |*S*| > *MaxLength* then remove the first |S|-*MaxLength*+1 items

   from *S*.

5. If |S| < *MinLength* then return *RR*

6. For each item ai from S to the end do

   If Current item points to next item

   Then insert the next item into RR order by their Support.

   Else

   Remove the First item from S and repeat from Step5.

   Return RR.

In general, each line of web logs (one access record) includes the following key information: date-timestamp, client IP address, user ID, requested URL, and HTTP status code. Given a sequence database where each sequence is a list of transactions ordered by transaction time with each transaction comprising a set of items, find all sequential patterns with a user-specified minimum support, which is defined as the number of data sequences containing the pattern. Let E be a set of unique access events, which represents web resources accessed by users, i.e. web pages, URLs, topics or categories. A web access sequence S = e1e2… en (ei ∈ E) for 1 ≤i ≤n is an ordered collection sequence) of access events, and |S| = n is called the length of the web access sequence. A web access sequence S is called a sequential web access pattern, if sup(S) ≥ MinSup, where MinSup is a given support threshold. An access event ei ∈ E is called a frequent event, if sup(ei) ≥MinSup. Otherwise, it is called an infrequent event. A Pattern-Data Structure model is proposed for storing sequential web access patterns compactly, so that it can be used for matching with a user's current access sequence and generating recommendation rules more efficiently in the Recommendation Rules Generation component. Based on list of accessed web page sequence within different sessions, a Directed Graph can be constructed called web usage graph. Graph consists of vertices (nodes) and edges (links) in which, nodes are for web pages, and edges represent sequential access between the pages. The number of nodes in the graph is equal to number of distinct web pages accessed during all sessions But before creating a graph first we have to remove the infrequent element from the given web access sequence, which can be decided by checking that the occurrence of web page in each sequence as a page count is less than the given min_count then it is called infrequent. So we have to just remove the webpage from the given web access sequence. Then we have to create a graph from the given sequence . Each edge in graph contains weight as link count, that represents the frequency of edge, edge id & list of sessions involved in that path or edge. In the prune_Graph(Web_Usage_graph G<N,E>, min_count),N is for nodes and E is for Edges. If the link count is less than the min_count then we have to remove the link. In the Process of Mining, By traversing nodes of Pruned Web Usage graph starting through each node, we get the frequently occurred sequence of nodes that represents frequent web access patterns. The frequency of the sequence will be the minimum link count of all the edges involved in that sequence. In such manner, traverse all the existing path in the Pruned Web Usage graph and enlist all frequent patterns along with their frequencies and arrange them by order of length. Frequent sequence with length of 1 are all the nodes in Pruned Web Usage graph itself, node count will represent their frequency. Following Algorithm is for mining the pruned web usage graph to get set of frequent sequential web access Patterns.

The recommendations are retrieved for a given user's web access sequence S , length of the user web access sequence S must satisfy the thresholds (minlen and maxlen). If its length is greater than maxlength then we have to remove first (S-maxlength+1) element. If it contains next item then return the recommendation rule order by the support [12].

3. IMPLEMENTATION AND RESULT ANALYSIS

Software Requirements

- Microsoft SQL server
- .NET Framework 3.5
- Jitbit Log2SQL Software

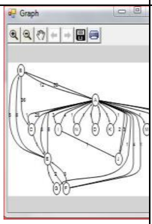
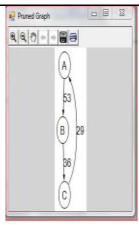
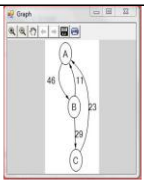
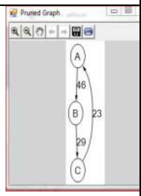
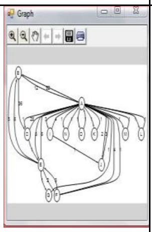
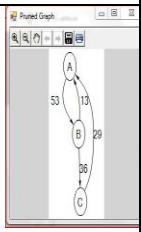
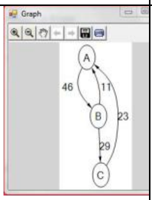
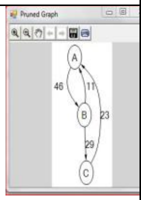
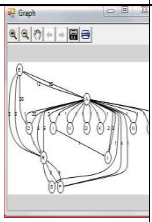
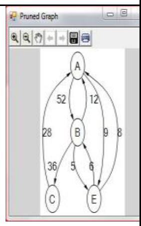
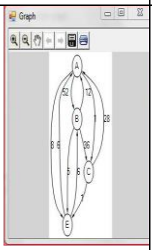
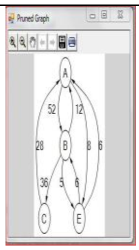
The log Files are Stored at IIS Server. For pattern discovery first we have to convert the Log File into SQL table .for that we have used JITBIT LOG2SQL software, which converts the log file into SQL table.We have used the Visual Studio .net framework 3.5 and SQL Server to preprocess the Web Log Data. We have taken the Web Log data of one site. The total number of records is 42944. After that we have Performed Cleaning on the Data. So the Data is reduced. The Records' URL Stem which contains .jpg, .css, .js, .png, .gif, robot.txt, .bmp etc is removed. Then we have done the User Identification. So, total distinct 23 users are found. Then we have Performed Session Identification. So individual User Contains Session of one Hour. If his total hit time occurs more than 1 hour then new Session occurs. After the session identification, we store the session's web access sequence in a table and giving each page one alias name. In old algorithm for creating a web usage graph we are giving text file of web access sequence as an input. Then we create a graph. and after pruning it's structure will be reduced. so, complexity decreases. The pruned graph is applied as an input for the depth first search algorithm. From that we mine the frequent sequential web access patterns order by their length. Where as in modified algorithm first we remove infrequent elements then we create a graph. After creation of the graph we apply depth first search algorithm to create a frequent sequential web access patterns.

From the frequent sequence patterns we get the recommendation rule order by their support.

Example: Let's consider the current access sequence of a user is S = cab (infrequent events have been removed) and the length thresholds of web access sequence are MinLength = 2 and MaxLength = 4 .The recommendation rules for S are generated as follows. The sequence matching process starts from the first sequence item c. We found a node (c:3) , then the second sequence item a is scanned. But the node (c:3) has no next node labelled a. The first item of S is then removed, and then S = ab. Because |S| =2 = MinLength, the sequence matching process is repeated. Now, the matching path (a:4) to (b:4) can be found . The node (b:4) has two nodes, which are (a:4) and (c:3). Finally, the recommendation rules for the current access sequence S = cab are generated as {a, c}, which are ordered by their support. Based on the recommendation rules, the corresponding web pages can be determined and recommended. From comparison between the time taken for generating the graph using old algorithm and modified algorithm shows that the total time for generating a graph and pruned graph in old algorithm takes much time than the modified algorithm.

And in old algorithm first graph generation occurs so more space is required for Memory in compare of modified algorithm because in modified algorithm we first remove the infrequent elements then we create a graph.

Table 2. Time for generating a graph and pruned graph from old and modified algorithm

				Time to generate
Min_Count=15	Old algorithm			Graph:5.2499ms Pruned Graph:12.4318ms
	Modified Algorithm			Graph:6.7341ms Pruned Graph:5.3923ms
Min_ Count =10	Old algorithm			Graph:5.04074ms Pruned Graph:11.2491ms
	Modified Algorithm			Graph:6.40646ms Pruned Graph:4.69899ms
Min_ Count =5	Old Algorithm			Graph:5.26273ms Pruned Graph:10.9745ms
	Modified Algorithm			Graph:6.50142ms Pruned Graph:5.73109ms

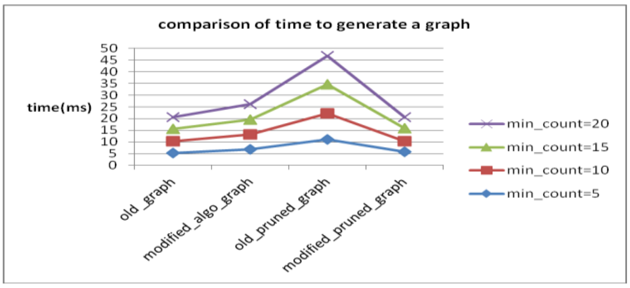


Fig 3: comparison of time between old and modified algorithm for generating graph and pruned graph

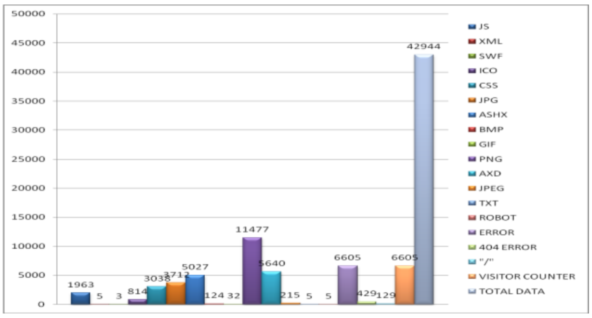


Fig 4: Preprocessing of Web Log Data

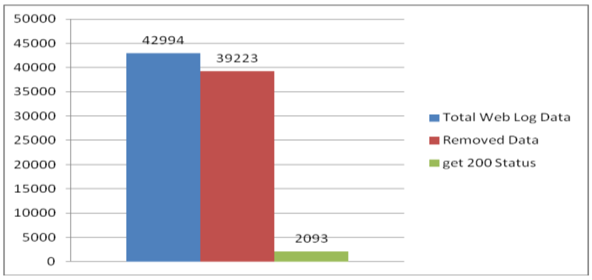


Fig 5: after Preprocessing getting Customized Data for Use

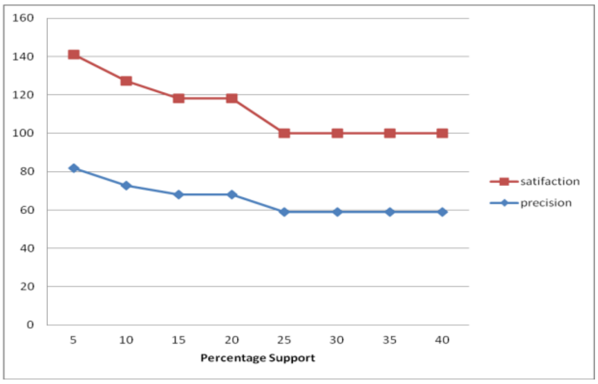


Fig 6: Percentage support Vs precision and Satisfaction

3.1 Evaluation Measures

Let $S = a1a2\ldots akak{+}1\ldots\ldots an$ be a web access sequence.For the prefix sequence $Sprefix = a1a2\ldots ak$ ($k \geq MinLength$), we generate a recommendation rule $RR = \{e1, e2, \dots, em\}$ using the Graph , where all events are ordered by their support [12].

- If $ak+1 \in RR$, we label the recommendation rule as *correct*, and otherwise *incorrect*.
- If there exists $ai \in RR$ ($k+1 \leq i \leq k+1+m$, $m > 0$), we deem the recommendation rule as *m-step satisfactory*. Otherwise, we label it as *m-step unsatisfactory*.

Let $R = \{RR1, RR2, \dots, RRn\}$ be a set of recommendation rules, where RRi ($1 \leq i \leq n$) is a recommendation rule. $|R| = n$ is the total number of recommendation rules in R. We define the following measures.

Definition 1: Let R_c be the subset of R that consists of all correct recommendation rules. The overall web Recommendation *precision* is defined as

Precision= $|R_c| / |R|$

This precision measures how probable a user will access one of the recommended pages.

Definition 2: Let $R_s(m)$ be the subset of R that consists of all *m*-step satisfactory recommendation rules. The *m-step Satisfaction* for web recommendation is defined as

Satisfaction $= |R_s(m)| / |R|$

In Preprocessing we are cleaning the data. So, the data is reduced, it takes less space of memory storage.

The experimental results given in Fig.10 show that as the Percentage Support threshold is increased, the *precision* and *satisfaction* remain relatively stable. From both experiments, we can conclude that better recommendations can be obtained with smaller support thresholds, at the expense of increased computational complexity for sequential web access pattern mining and maintaining a larger.

4. CONCLUSION

Personalization for a user can be achieved through web usage mining. Common access behaviours of the users can be used to improve the actual design of web pages and for making other modifications to a Web site. using the modified algorithm we have reduced the time to generate a graph in compare of old algorithm. And less memory storage is required due to preprocessing. The contribution of the paper is to introduce a new way of web usage mining, and to show how frequent pattern discovery tasks can be accomplished by capturing complex user's browsing behaviour into a graph in order to obtain hidden useful user's access patterns information.

5. REFERENCES

[1] S.K. Pani, L.Panigrahy, V.H.Sankar, Bikram Keshari Ratha, A.K.Mandal, S.K.Padhi, "Web Usage Mining: A Survey on Pattern Extraction from Web Logs", International Journal of Instrumentation, Control & Automation (IJICA), Volume 1, Issue 1, 2011

[2] Yogish H K, Dr. G T Raju, Manjunath T N, "The Descriptive Study of Knowledge Discovery from Web Usage Mining", IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 5, No 1, September 2011

[3] Udayasri.B, Sushmitha.N, Padmavathi.S, "A LimeLight on the Emerging Trends of Web Mining" , Special Issue of International Journal of Computer Science & Informatics (IJCSI), ISSN (PRINT):2231–5292,Vol.-II,Issue-1,2

[4] Navin Kumar Tyagi, A.K. Solanki & Sanjay Tyagi. "An Algorithmic approach to data preprocessing in Web usage mining", International Journal of Information Technology and Knowledge Management July-December 2010, Volume 2, No. 2, pp. 279-283

[5] Surbhi Anand , Rinkle Rani Aggarwal, "An Efficient Algorithm for Data Cleaning of Log File using File Extensions", International Journal of Computer Applications (0975 – 888),Volume 48– No.8, June 2012

[6] J. Vellingiri and S. Chenthur Pandian, "A Novel Technique for Web Log mining with Better Data Cleaning and Transaction Identification", Journal of Computer Science7 (5): 683-689,2011 ISSN 1549-3636 © 2011 Science Publications

[7] Priyanka Patil and Ujwala Patil, " Preprocessing of web server log file for web mining", National Conference on Emerging Trends in Computer Technology (NCETCT-2012)", April 21, 2012

[8] Vijayashri Losarwar, Dr. Madhuri Joshi, "Data Preprocessing in Web Usage Mining", International Conference on Artificial Intelligence and Embedded Systems (ICAIES'2012) July 15-16, 2012 Singapore

[9] Sachin yele, Beerendra Kumar, Nitin Namdev, Devilal Birla, Kamlesh Patidar.,"Web Usage Mining for Pattern Discovery", International Journal of Advanced Engineering & Applications, January 2011.

[10] Chetna Chand, Amit Thakkar, Amit Ganatra, "Sequential Pattern Mining: Survey and Current Research Challenges", International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-1, March 2012

[11] Dheeraj Kumar Singh, Varsha Sharma, Sanjeev Sharma "Graph based Approach for Mining Frequent Sequential Access Patterns of Web pages", International Journal of Computer Applications (0975 – 8887) Volume 40– No.10, February 2012.

[12] Zhou.B, Hui. S.C and Chang. K, "An intelligent recommender system using sequential Web access patterns", IEEE Conference on Cybernetics and Intelligent Systems, vol. 1, pp. 393 - 398, December 2004.