An Improved Approach for Mining Frequent Sequential Web Access Patterns from Web Logs

Sheetal Jain¹ Prof. Jasmine Jha²

¹,²L. J institute of Engg & Technology, Ahmedabad, Gujarat, India

Abstract—In the growing world of IT, We need to evaluate users' browsing behavior and provide them recommendations to maintain the users' interest. we propose a naive graph based approach which stores the items into the adjacency matrix by creating the directed graph. The frequent sequential web access patterns are then retrieved from the matrix, which is then used to fetch and generate recommendation links.

Key words: Frequent Pattern Mining, Recommendations

I. INTRODUCTION

Data mining is referred to as an extraction of knowledge or analysis of patterns from the huge amount of data set[s]1. With the substantial growth of information sources on the world wide web, It is important for the users to use automated tools to find the required information and keep track of the data usage and web access patterns. Web usage mining deals with understanding of user behaviour, while interacting with website, by using various log files to extract knowledge from them.

II. RELATED WORK

Graph based algorithms are used to mine the frequent sequential patterns from the web and through that the navigational behavior of a user can be obtained. The basic steps in mining the frequent patterns are constructing a graph, pruning of graph and mining the frequent patterns by different methods by considering client side data and server logs. Patterns can be mined by different approaches that are SWARS, or FSP algorithms, tree based approach or graph based approach.

III. PROBLEM SPECIFICATIONS

Mining the web data is one of the most challenging tasks because the data available on the web are generally huge, heterogeneous and less structured. In the traditional graph based approach, numbers of traversals are more for creating and pruning of web usage graphs. Thus It takes more time to generate web access patterns and has higher computational time. The proposed graph based approach will take less time and memory to generate the web access patterns

IV. PROPOSED WORK

In proposed graph based algorithm, Transaction dataset D is given as an input. here the whole database is compressed by converting in the form of a directed graph which is stored as an adjacency matrix. The vertex represents the item of a dataset and edges represent the links between the item sets. The diagonals of the adjacency matrix shows the frequent-1 item sets and rest of elements shows occurrence of frequent-2 item sets. here we take upper and lower triangular elements in consideration because the items are arranged in lexicographical order. Proposed work flow is as shown in figure.

Fig. 1: Proposed Work Flow of Naive Graph Based Approach

A. Algorithm 1: Frequent Sequential Pattern Mining Algorithm

- Input: Transactional Dataset D and total number of items is n
- Output: Frequent Patterns Lk

Algorithm works in following steps,
1) Scan the database D and create directed graph G in the form of an adjacency matrix I
2) Update the value of each element with its position for Ii,j list and Ii,j count
3) Read each element Ii,j of matrix I , if Ii,j count < min_sup then Ii,j count = 0
4) Construct the reduced adjacency matrix by deleting corresponding row and column of element Ii,j count = 0 only for i=j
5) Find frequent - 1 item sets and 2 item sets direct from matrix elements
6) Find frequent k item sets (k >2) from each row using logical AND operation with its position procedure between elements of the row
7) End

B. Algorithm 2: Recommendation Rule Generation Algorithm

- Input : Frequent Sequential Patterns
- Output: RR - recommendation rule of a set of ordered access events for S.

Algorithm works in following steps,
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 = φ
4) If |S| > MaxLength then remove the first |S| - MaxLength + 1 items
5) If |S| < MinLength then return RR
6) For each item ai from S to th end do
   If current item points to next item
   Then insert the next item into RR order by their Support.

All rights reserved by www.ijsrd.com
Else
Remove the first item from S and repeat from step 5
7) Return RR

1) Example for Proposed System

<table>
<thead>
<tr>
<th>Session ID</th>
<th>Web Access Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>ABDAC</td>
</tr>
<tr>
<td>S2</td>
<td>EAEBAC</td>
</tr>
<tr>
<td>S3</td>
<td>BABFAE</td>
</tr>
<tr>
<td>S4</td>
<td>AFBCAFC</td>
</tr>
</tbody>
</table>

Table 1: Set of Web Access Patterns

Fig. 2: Web Usage Graph

Fig. 3: Adjacency Matrix I for item sets

Each Element of a matrix has two fields,
1) List (contains session ids of corresponding patterns)
2) Count (contains integer value which is equivalent to the number of session ids in list)

In directed graph, nodes D, E, F are removed because their support count is less than minimum support count. Now, the filtered matrix or reduced matrix can be generated from the full adjacency matrix as follows,

Except diagonal elements, from the other elements of an adjacency matrix Frequent - 2 item sets,

- \( I_{AB} \) list = \( S_1, S_2, S_3, S_4 \)
  \( I_{AB} \) Count = 4
- \( I_{BC} \) list = \( S_1, S_2, S_3, S_4 \)
  \( I_{BC} \) Count = 3
- \( I_{CA} \) list = \( S_2 \)
  \( I_{CA} \) Count = 1 < minimum support count
- \( I_{CB} = \phi \)

For extracting frequent k (k=2) item sets, we apply logical AND operation in between each row elements of the adjacency matrix. Some Frequent-3 item sets are calculated as below,

- \( I_{ABC} = I_{AB} \cap I_{AC} \) List \( = (S_1, S_2, S_3, S_4) \cap (S_1, S_2, S_3, S_4) \)
  \( I_{ABC} \) Count = 3
- \( I_{ABC} = I_{AB} \cap I_{BC} \) List \( = (S_1, S_2, S_3, S_4) \cap (S_1, S_2, S_3, S_4) \)
  \( I_{ABC} \) Count = 3
- \( I_{ABC} = I_{AB} \cap I_{AC} \) List \( = (S_1, S_2, S_3, S_4) \cap (S_1, S_2, S_3, S_4) \)
  \( I_{ABC} \) Count = 4
- \( I_{ABC} = I_{AB} \cap I_{AC} \) List \( = (S_2) \cap \phi = \phi \)

V. EXPERIMENTAL RESULTS AND COMPARISONS

To study the performance of the graph based approach, we have implemented this algorithm in C# dot net and microsoft visual studio 10 with 512 mb RAM and pentium IV machine. Comparison of traditional approach to proposed approach in terms of computational memory and execution time parameters. The chart for the comparison of execution time in milliseconds with no. of logs.
VI. CONCLUSIONS AND FUTURE ENHANCEMENTS

For providing better recommendation for a user, it is necessary to model the users’ behaviour. Adjacency matrix is used to find the web access patterns effectively. The proposed approach does only single scan of database and time for pattern generation is reduced.

It is a novel approach that uses graph theory in the best possible way using an adjacency matrix. It is an improvement over traditional pattern mining algorithms but it has enough space for further modifications and refinements.

REFERENCES


