To Enhance Website Performance for Caching with SP-SOM Technique using Progressive Database

Richa Mishra¹, Prof. Yogesh Rai²

Shree Institute of Science and Technology, Bhopal

Abstract: Every Organization need to understand their customer’s behavior, preferences and future needs, which depend on past behavior. Web Usage Mining is an active research topic in which user session clustering is done to understand user’s activities. In this paper, we use Neural based approach Self Organizing Map for clustering of session as a trend analysis with some parameters. It depends on the performance of the clustering of the number of requests. Here we are using SOM algorithm in Most Frequent Sequential Traversal Pattern Mining called SP-SOM and generated cluster of web data. In this research we establish good prediction with quantity of data and the quality of the results.

Keywords – Web Usage Mining; Frequent Sequential Patterns; Sequence Tree; Web Log Data; Web Services; Neural Network; Clustering.

I. INTRODUCTION

The WWW [2] is an immense source of data that can come either from the Web content, represented by the billions of pages openly available, or from the Web usage, represented by the register information daily collected by all the servers around the world. Web Mining is that part of Data Mining which deals with the extraction of interesting knowledge from the World Wide Web. Web usage mining [4] has many applications, e.g., personalization of web substance, support to the design, recommendation systems, pre-fetching and caching [23]. Kohonen Self-Organizing Maps (SOM) [8, 10] developed by Tuevo Kohonen, a professor emeritus of the Academy of Finland. SOMs learn unsupervised competitive learning. “Maps” is because they attempt to map their weights to conform to the given input data. The nodes in a SOM network attempt to become like the inputs presented to them. The topological relationships between input data are preserved when mapped to a SOM network.

II. RELATED WORK

Prefix Span [1], a more efficient pattern growth algorithm was proposed which improves the mining process. The main idea of Prefix Span is to examine only the prefix subsequences and project only their corresponding suffix subsequences into projected databases. The database projection growth based approach, Free Span [1], was developed. Although Free Span outperforms the Apriority based GSP algorithm, Free Span may generate any substring combination in a sequence. The projection in Free Span must keep all
sequences in the original sequence database without length reduction.

In SPADE [11], a vertical id-list data format was presented and the frequent sequence enumeration was performed by a simple join on id lists. SPADE can be considered as an extension of vertical format based frequent pattern mining. The discovery of the user’s navigational patterns using SOM [8, 10] is proposed by Etminani. Huge amount of information are collected repeatedly by web servers and gathered in access log files. Analysis of server access data can offer important and helpful data. The author used the Coonan’s SOM (Self Organizing Map) to preprocessed web logs for extracting the common patterns.

In WUM [14] we find the behavior of user either it is registered or not. If a website requires users to sign in before they can start browsing, it will be very easy not only to differentiate between users but also to identify each single user. The problem arises when a website allows visitors to anonymously browse its content, which is common place. In this paper differentiate between visitors activity as a challenging task in the log using common log format [6].

III. PROPOSED WORK

In this paper, we have used Self Organizing Map (SOM) with frequent sequential pattern. SOM is a type of neural network. In the process of Web Usage Mining [14] to detect user’s patterns it is usage as a trend analysis. It depends on the performance of the clustering of the quantity of requests. Here we are using SOM algorithm with SP-SOM (Frequent Sequential Traversal Pattern Mining with SOM) algorithm.

![Diagram of Proposed SP-SOM Approach](image)

Figure-1: Proposed SP-SOM Approach

The procedure details the transformations essential to modify the data storage with clustered in the Web Servers Log files [6] to an input of SOM. By proceeding this way, first we use SOM algorithm and getting some cluster of web-data. Here we load the web-data cluster, which is almost related to frequent pattern. After that we are applying min-max weight of Page in Sequential Traversal Pattern. Finally we establish good prediction with quantity of results. The figure-1 shows the process of proposed work where it collect the sessional web
data and applying SOM after preprocessing [7]. Here it mine density based clustering and then find the closed frequent item from the sessions web data for getting useful information.

3.1 SP-SOM Algorithm
The proposed algorithm is used for finding most frequent sequential traversal patterns with clustered index. To handle the ordered problem, the SP-SOM first filtered frequent sequential pattern by using support with min-max or average weight parameter [12] of item. After that it uses neural network algorithm for clustering of index with similarity of object. At last it create most frequent sequential pattern tree [21]. This tree is create less candidate set and also uses to predict next item in caching [23].

3.2 Procedure of Sequential Pattern with SOM Technique
The procedure for constructing the pattern tree in the proposed system is as follows:
Step-1: Collect the web logs of website.
Step-3: Supply input as number of support by the user and checks Min-Max weight or Average weight of page and generate frequent sequential pattern.
Step-4: Apply SOM algorithm in Sessional Frequent Sequential Pattern Item. So here each and every item belongs to at-least one cluster according to similarity.
Step-5: Convert frequent sequential pattern into Frequent Sequential Pattern [18] and generate the Pattern-Tree for next item prediction.
Step-6: Finally establish good cluster items for prediction into the caching to improve the quality of the results and response time.

IV. EXPERIMENTAL RESULT
This paper showing the result of clustered web data using SOM and also used frequent sequential pattern for pre-fetch the next item in cache as on the behavior of similar pattern access of user. The Table-1 showing page details with Support and Min-Max weight range.

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Page ID</th>
<th>Page Name</th>
<th>Support</th>
<th>Min. Weight</th>
<th>Max. Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>P1</td>
<td>Books</td>
<td>9</td>
<td>2</td>
<td>31</td>
</tr>
<tr>
<td>2</td>
<td>P2</td>
<td>Electronics</td>
<td>7</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>P3</td>
<td>Cloths</td>
<td>7</td>
<td>4</td>
<td>22</td>
</tr>
<tr>
<td>4</td>
<td>P4</td>
<td>Jeweler</td>
<td>6</td>
<td>5</td>
<td>9</td>
</tr>
<tr>
<td>5</td>
<td>P5</td>
<td>Furniture</td>
<td>6</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>P6</td>
<td>Toys</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>P7</td>
<td>Root</td>
<td>2</td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>

Table-1: The example of page with weight range

The Table-2 showing the Items details of every page which belongs to page.
The Table-3 shows the Running time (in ms) when we having different database record size with different supports.

<table>
<thead>
<tr>
<th>Support / Size</th>
<th>1%</th>
<th>2%</th>
<th>3%</th>
<th>4%</th>
<th>5%</th>
<th>6%</th>
</tr>
</thead>
<tbody>
<tr>
<td>1033</td>
<td>6817</td>
<td>6708</td>
<td>6723</td>
<td>6505</td>
<td>6474</td>
<td>6708</td>
</tr>
<tr>
<td>2040</td>
<td>18408</td>
<td>18111</td>
<td>18252</td>
<td>18158</td>
<td>18376</td>
<td>18470</td>
</tr>
<tr>
<td>3050</td>
<td>15865</td>
<td>17953</td>
<td>18565</td>
<td>18764</td>
<td>18451</td>
<td>18487</td>
</tr>
<tr>
<td>4010</td>
<td>24310</td>
<td>24688</td>
<td>24927</td>
<td>25205</td>
<td>26365</td>
<td>24949</td>
</tr>
<tr>
<td>5030</td>
<td>83413</td>
<td>38079</td>
<td>84630</td>
<td>54116</td>
<td>40731</td>
<td>38391</td>
</tr>
</tbody>
</table>

The Table-4 shows the probability of occurrence of each item with different support.

<table>
<thead>
<tr>
<th>Support 3%</th>
<th>Item-1</th>
<th>Item-2</th>
<th>Item-3</th>
<th>Item-4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.44</td>
<td>0.19</td>
<td>0.15</td>
<td>0.22</td>
<td></td>
</tr>
<tr>
<td>Support 6%</td>
<td>0.50</td>
<td>0.17</td>
<td>0.08</td>
<td>0.25</td>
</tr>
<tr>
<td>Support 10%</td>
<td>0.43</td>
<td>0.20</td>
<td>0.13</td>
<td>0.23</td>
</tr>
<tr>
<td>Support 20%</td>
<td>0.33</td>
<td>0.00</td>
<td>0.33</td>
<td>0.33</td>
</tr>
</tbody>
</table>

The figure-2 shows the Running time (in ms) when we having different record size with different support.

Figure-2: Running Time (in ms) with different size and different support

The Figure-3 shows the probability of occurrence of each item with different support.

Figure-3: Probability of items with different support
The Table-5 shows Info-Gain of Item with different support.

<table>
<thead>
<tr>
<th></th>
<th>Item-1</th>
<th>Item-2</th>
<th>Item-3</th>
<th>Item-4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support-3%</td>
<td>07.02</td>
<td>06.08</td>
<td>05.51</td>
<td>06.51</td>
</tr>
<tr>
<td>Support-6%</td>
<td>12.00</td>
<td>10.34</td>
<td>07.17</td>
<td>12.00</td>
</tr>
<tr>
<td>Support-10%</td>
<td>07.84</td>
<td>06.97</td>
<td>05.81</td>
<td>07.35</td>
</tr>
<tr>
<td>Support-20%</td>
<td>01.00</td>
<td>01.00</td>
<td>00.00</td>
<td>01.00</td>
</tr>
</tbody>
</table>

**Table-5: Info-Gain of items with different support**

The Figure-4 shows Info-Gain of Item with different support.

![Figure-4: Info-Gain of items with different support](image)

The Figure-5 shows comparison between WSpan and SP-SOM algorithm.

![Figure-5: Comparison of WSpan and SP-SOM Algorithm with different support](image)

The table-6 shows the comparison between WSpan and SP-SOM Algorithm with different support. Here record size 5030 is taken in the database.

<table>
<thead>
<tr>
<th></th>
<th>1%</th>
<th>2%</th>
<th>3%</th>
<th>4%</th>
<th>5%</th>
<th>6%</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>WSpan</strong></td>
<td>83413</td>
<td>83210</td>
<td>84630</td>
<td>8339</td>
<td>84645</td>
<td>8399</td>
</tr>
<tr>
<td><strong>SP-SOM</strong></td>
<td>82009</td>
<td>83100</td>
<td>80745</td>
<td>5411</td>
<td>40731</td>
<td>3839</td>
</tr>
</tbody>
</table>

**Table-6: Comparison of WSpan and SP-SOM Algorithm with different support (By using Record-Size 5030)**

**V. ANALYSIS AND PERFORMANCE EVALUATION**

In this section, we present performance study over various datasets (e.g. 1000, 2000, 3000, 4000 and 5000 sessions) and also with different support (e.g. 3, 6, 10 and 20). The experimental results explored for the performance of SP-SOM with a recently developed algorithm, WSpan [1], which is the fastest algorithm for mining sequential patterns. The main purpose of this experiment is to demonstrate how effectively the sequential traversal patterns with min-max weight constraint can be generated by incorporating a support and weight page with clustering. First, we shows how...
The number of sequential traversal patterns can be adjusted through user allocate weights, the efficiency in terms of runtime of the SP-SOM algorithm, and the quality of sequential traversal patterns. Second, we show that SP-SOM has put related items in cache. Third we are using web services which provide automatically update min-max weight of every page in every fifteen days. It is also decrease back and forth time while finding next page from cache because it also store related page prior in cache [23].

VI. FURTHER EXTENSION
SP-SOM algorithm basically focuses on sequential pattern mining with average weight constraint uses a weight range to adjust the number of sequential traversal patterns with the clustering of session. SP-PM can be extended by considering levels of support and/or weight of sequential traversal patterns with number of clustering. We can also extended by using Distributed Weblog. There are many areas just like parallel sequential pattern, grouping of similar type of users in distributed servers.

VII. CONCLUSION
This proposed research just begins to touch on the possibilities of SOMs with the frequent sequential pattern mining. One of the main limitations of the traditional approach for mining sequential traversal patterns is that weight of every page is updated manually but here we updated automatically using web services. Second fully database scan is done while find the next item in traditional approach. Here we clustered the items so that clustered items are only scan not whole database. Third we use min-max weight and support of every page so that every page having different importance. So it is enough to perform extremely computationally expensive operations in a relatively short amount of time for finding next page prediction.

REFERENCES
using A Bitmap Representation”, SIGKDD’02, 2002.


