Predicting User’s Web Navigation behaviour using AMD and HMM Approaches

Rita Roy¹, G Appa Rao²
Research Scholar¹, Professor²
Department of CSE, GITAM Deemed to be university
Ritaroy89@gmail.com¹, apparao.giduturi@gitam.edu²

Abstract. The academics were introducing various techniques in web usage mining to reduce user latency time to improve their web performance. The server-side web log data helps to identify the most appropriate pages, based on the user’s request. Analyzing web log data creates difficulties since it comprises exhaustive web page data. This paper proposes a new technique in which the web log data can preprocess in order to extract sequence and navigation patterns useful to predict. In this paper, we cluster users into communities labelled by the websites most frequently visited, to discover their preferences and to describe website reorganisation. The URL sequence navigated by a user during a session. Session denotes a user’s browsing pattern. Multiple users’ sessions are clustered using the hierarchical clustering technique to analyse the sequence in their navigation patterns. The two model predictions are Adaptive Mahanabolis Distance (AMD) and Hidden Markov model (HMM), providing a list of web pages of interest to the user. We validate the proposed framework on NASA, Clarknet and serc datasets web log files.

Keywords: Prediction model, clustering, User Sessions, Adaptive Mahanabolis Distance

1. Introduction

As a practice, Internet surfing logs on internet services have been a crucial source of information to evaluate hidden preferences of users and to conclude their authentic lives. With a better knowledge of the mobile usage pattern, network service companies can offer more customised offerings and enhance the efficiency of the service. The browsing preferences of users in fields like community design, smartphone ads, transportation, schooling etc. are often beneficial [1–3]. Web use mining relates to the identification of interesting web access trends from the background of web data. Typically, an organization’s Database servers monitor its users’ web behaviour in a file named log file. Three forms of log files are available, which can be used for web mining. On the web, network and proxy clients, log files are processed. It allows mining quite complicated by providing over one place to store information concerning navigation patterns of users. Only when one requires data from all three, log they will achieve file forms. This is because the server-side may not include archives of the webpage accesses stored on the proxy site or on the client’s side. Besides the application log sheet, which includes detailed information on the proxy server. Therefore, the client-side page queries are absent. However, it is challenging to gather all customer information. Most algorithms therefore only work on the server side.

While the user accessing the website, the user transactions are monitored and stored as log. The log contains unstructured format, the conversion of unstructured to structured format by preprocessing technique. The weblog comprises various entries like IP address of user, date/time, categories of product and status code. The preprocessing is a three-step process. The data cleaning cleans the irrelevant data in a log. The user identification is identified by IP address of the specified system. It classifies the session identification based on session management. Identifying user needs and user interest is a laborious task. With the help of log, the user needs and interest can identify by user navigation pattern. They can analyse the user navigation pattern based on the previous user navigation that was stored on the weblog. Using pattern discovery, the processed log is to convert it into sequence and sub sequence of similar pattern. By using forward and backward technique, it can generate subsequence. They group the sub sequence pattern to form a cluster that will helpful to identify the user needs. Some commonly used data mining algorithms for Web usage mining are association rule mining, sequence mining and clustering.
Clustering studies how domain expertise can enhance the efficiency of clusters. It presents domain information as a collection of constraints to apply to the clusters. Orthodox methods to semi-supervised (or restricted) clustering use one of three restrictions. Next, it may update an established clustering algorithm to take into consideration. This method has been adopted by COPKMeans [4], among the first clustering algorithms to solve pairs of restrictions. Second, a distance metric focused on the limitations [5] may be taught, during which it uses the metric in a conventional cluster analysis. Third, the above two techniques may be merged and the so-called hybrid methods developed [6].

2. Related works

In this section, we briefly review some of the methods that are proposed for clustering for webpage prediction techniques with relevant information:

In [7], the authors implemented an intuitive difference test to quantify the issues of site users with expanded web user sessions. It bases the suggested assessment of the disparity of usage between two web user sessions on the page importance, the syntactic page URL structure, and the hierarchical website structure. This suggested they have used logical dissimilarity test for research with the K-Medoids clustering algorithm and contrasts the findings to other independent dissimilarity measurements. Two unchecked cluster validity indexes assessed the significance of the created clusters. The findings from the experiment show that intuitively enhanced session dissimilarity metrics are more practical and superior to the other separate cluster validity index dissimilarity measurements. In [8], authors mentioned how site log data on the server side allows users to find the most suitable pages. Web log data review presents difficulties since it contains enough details about a web page. They developed a novel strategy for pre-processing Web log data to derive a predictable series of events and navigation patterns. They scan every URL in the server log data into web-based tokens. Tokens for URL identification are individually defined. The URL sequence, which a user navigates for 30 minutes, is known as a session. Session reflects a user’s browsing routine. It groups multiple user sessions using the hierarchical clustering method to evaluate the sequence occurrence of the browsing trends.

Each cluster recognises the session as the representative since it includes as many pages as possible in the sequence, whereas the representative session is a subset for other sessions in the cluster. Session model navigation trends are helpful to predict the user request pages most suitable. In [9], the authors found that site user session clustering is highly necessary to recognise their internet browsing behaviour. Users with common browsing activities and study by domain specialists in discovered user classes may yield valuable and constructive feedback. A conglomerate clustering technique is proposed to classify site users’ clusters based on their browsing activity from the Web server access logs. The algorithm provided incorporates interesting concepts from the subtractive and associated c-mean clustering approach. This algorithm operated in the first step and automatically identified the number of clusters dependent on each relational data and its respective centres’ successive subtractive potential density value (centroid). In the second step, it allocates fuzzy membership values to fuzzy clusters from some kind of relational matrix. The proposed method is an improvement in session dissimilarity matrix from the NASA web server log data that is freely available.

In [10], the authors presented an alternative method to creating sessions, the first big phase in network use mining. The suggested approach will get all potential Site users’ full navigation sequences. Finding suggest that when our current method is used, previous methods in web-based mining applications are outperformed, for example next-page prediction.

In [11], Relational Medoid-based fuzzy relational clustering (FRC) algorithms are common to the analysts and work better than oriented FRC. In medoid FRC, however, medoid collection is random and often contributes to conflicting outcomes. This paper suggests a subtractive medoid sorting approach focused on the SMS-FRC relational clustering. Geometry and density of peer-
referential dissimilarity values in the SMS-FRC algorithm are favoured over arbitrary initial values of medoids. The SMS-FRC is used to classify clusters of user sessions based upon the browsing behaviour of server log info. The definition of enhanced sessions is used to extract the intuitive matrix of expanded dissimilarity dependent on the page relevance.

In [12], the authors suggest a method to decrease the user’s search time and meet the user’s predicted intent website (request) through an increased prediction accuracy through the combination of the clustering fuzzy c-means and the model recommendation vector order Markov. At first, methods that allow are performed by applying a web log preprocessing method accompanied by a c-means clustering method to classify patterns of similarity. After that, it renders the recommendation of the web page utilizing a model of variable orders to predict access to the next web page by reducing search time and improved predictive accuracy. In [13] the authors recommend a contribution to solving prediction space and to increase precision by integrating CPT+ and PageRank algorithms. In [14] the authors present a global eye monitoring index descriptive named heat map entropy and distinguish its webpage aesthetic predictive benefit. All the above works are inadequate for non-convex data, comparatively vulnerable to surfaces, quickly drawn to optimum local data, the sum of clusters required to be preset and the product of a cluster sensitive to the number of clusters. Our framework that can cope with them and results in optimum efficiency, thus is implemented by implementing the Adaptive Mahanabolis Distance algorithm and the K-means clustering algorithm that can be addressed later.

3. Material and Methods

A system has been proposed, in which session identification is utilized and implemented based on the user who spent time on each page, and also, we analyze the number of webpages accessed by an individual user in a particular session. Potential and non-potential web users were identified from the web server log file. Adaptive Mahanabolis Distance clustering algorithm and K-means clustering algorithm (for comparison purpose) were used to identify the frequently accessed pages. We focus on these models on weight variables, such as time spent on the webpage and the maximum amount of page accesses. The analysis also reveals that performing the AMD cluster algorithm is better than the clustering algorithm of K-means for web log results. The analysis proves the AMD has a substantial advantage in terms of the execution period over the K-means clustering algorithm. The aim of this paper is to effectively predict and prefect a website in order to minimize latency. The overview of the pre-fetch and prediction unit is shown in Figure 1 with necessary details and records for the phase after pre-processing. We have established unique users based on the unique IP address. After we have established the particular user, web users have been classified by web session. A web session is a period the web user takes on a single website. The web session begins from the user’s login period to the site’s leaving time. We may develop session Ids to store those variables as web visitors are going through the website sites. Before presenting the log file analysis technique, this portion of work is a minor part in considered objectives, which minimize the log file size and cluster the frequently accessed webpages by considering time spent on the page as a weight factor.

Algorithm 1: Data Cleaning

**Input**: Raw Web log Data

**Output**: Pre-processed Web log Data

1. **begin**
2. Read Web log data from log file
3. Parse web log data into tokens
4. **If** (web log data! =null) **then**
5. **Assign** group based on the following Categories
6. **If** (Web log data.url= “*.jpg, *.gif, *.png”) **then**
7. Assign as Unwanted Images Files
8. **else if** (Web log data.url= “*.flv, *.avi, *.mpg, *.mpeg”) **then**
9. Assign as Unwanted Video Files
Algorithm 2 explains user identification algorithm. We can identify Web users from the web server log file constructed on their IP address. If the same IP address exists, we categorize them as same users. If the IP address is different, then the user is different and is identified from the user identification algorithm.

**Algorithm 2: User Identification**

**Input:** Pre-Processed Web log data  
**Output:** Frequently accessed webpages  
1. **begin**  
2. Read the Pre-Processed Web log data  
3. Compare the IP address of two consecutive records  
4. **If** (IP address = same) **then**  
5. Assign as the Same User  
6. **else**  
7. Distinct User  
8. **end**  
9. **end**  
10. Save Records  
11. **end**  
12. Repeat until the last record  
13. **end**
8. endif
9. end

Each clustering algorithm seems to have an objective function that directs the search for potential solutions. This function assesses candidate solutions and shows how good (bad) each would be. The method then seeks to optimize (reduce) the function value. If we provide the algorithm with a certain number of constraints, we request that the algorithm only takes into account solutions which completely fulfill the constraints. Clustering is one of the data processing methods used for grouping related data. The algorithm Adaptive Mahalanobis Distance (AMD) depends on iteratively recalculating the distance between Mahalanobis and a neighborhood center.

Algorithm 3: Clustering using Adaptive Mahalanobis Distance (AMD)

Input: Pre-Processed Web log data
Output: Frequently accessed webpages

1. begin
2. Assume an isotropic Mahalanobis distance (Σ = I)
3. Find the closest $K$ neighbors by computing the Mahalanobis distance
4. Using $K$ neighbors, compute the covariance
5. If covariance = Mahalanobis distance go to step 6
6. else go to step 2 and re-compute the Mahalanobis distance
7. Record the points as neighbors of dataset
8. end

Algorithm 4 depicts the proposed algorithm for session identification. It identifies the potential and non-potential users based on time stamp and number of pages accessed by the user. Webpages in which the user spent time more than the threshold value (i.e. 60 seconds) is considered. The web users who accessed more the three pages were considered as potential users and the users who access less than three pages were considered as non-potential users.

Algorithm 4: Session Identification

Input: User Identified Web log data
Output: Maximum time spent on the webpage and the potential web user

1. begin
2. Let $P$ ← Potential User;
3. Let $NP$ ← Non-Potential User;
4. Let $T$ ← Threshold Value;
5. Let $Pi$← first entry of log file in a particular session;
6. Let $Pj$← second entry of log file in the same session;
7. Compare two consecutive entries within the session;
8. if timestamp ($Pi$) is < timestamp ($Pj$) then
9. The difference between the timestamp $Pi$ & $Pj$ should be greater than $T$;
10. endif
11. if each user accessed more than three pages within the session then
12. Assign $P$;
13. else
14. Assign $NP$;
15. endif
16. end

After identifying the potential users from web server log file using Algorithm 4 and then applied clustering algorithms to find frequently accessed webpages.
3.1 Framework of Prediction and Prefetching:

As Figure 2 shows, the suggested model is applied relying on the log file of the web server. We have collected it on the NASA website [15]. The raw web log data gathered includes irrelevant data. It must then be pre-processed and grouped we know such that commonly visited webpages. We tracked each web user interaction and its actions on a specific website. Once their actions and interest have been established, the process of prediction begins by examining the previous web user behavior. Table 1 displays the user navigation sample list. The accessed user 1 series was recorded from the table as P1→P3→P5→P7→P9 etc.

Let us consider a website comprising \( w \) number of webpages, namely \( up_1, up_2, \ldots, up_n \) and number of clusters \( C_1, C_2, \ldots, C_n \). Let Web Server Log File be referred to as \( WSLF \). The pre-processed log file contains the user id, name of the page and time spent. The pre-processed log file is \( WSLF_1, WSLF_2, \ldots, WSLF_n \). Calculate the overall access rate for each cluster, determined by taking into consideration the amount of times the page requires for a cluster. We arrange the web pages according to the method suggested. After analyzing the use of web pages for the time needed (every day, weekly), we generate the clusters based on web pages regularly visited. Calculate the average web page access level for each cluster. We propose an algorithmic architecture for the reorganization of the built environment. We propose an algorithm for reorganizing web pages. After the webpages are frequently accessed, threshold frequency pages are expected, and the number of users viewed by the same Hidden Markov Model (HMM) page is stated to be ideal for discovering several sites to be loaded and it can be submitted to a browser as a hint for pre-fetching.

\[
Thr_f = \text{Frequency count (FC)} > \text{(threshold)} \lambda \tag{1}
\]
Web pages frequently visited were detected by examining a user’s previous behavior. Prediction is not dependent on a previous user’s regularly visited websites. Predictions focused solely on frequent count contribute to mis-prediction, and there would be no satisfactory prediction outcomes from this method.

3.2 Hidden Markov Model to Predict User’s Browsing Behavior:

After processing the data unit, we need to build an effective model that contains the parameter structure of the model. The consistency of the model depends on the state transition diagrams and the related parameters. At first, we assign a value between 0 and 1 to arbitrarily assign model parameters (transition likelihood $A$), satisfying $a_{11} + a_{12} = 1$ and $a_{21} + a_{22} = 1$ (see Figure 3). And the original distribution of odds, $\pi 1$ and $\pi 2$ ($\pi 1 + \pi 2 = 1$). We use the Baum-Welch algorithm to reach an acceptable parameter. We use parameters $\lambda = (A, B, \pi)$ to simultaneously reveal secret states when browsing the website through the Viterbi algorithm.

![Figure 3: The training phase of HMM](image)

Here $N$ separate web pages and two hidden states are present in our method, called $S_1$ and $S_2$. The $x_t(P_t)$ notation shows the possibility that page $P_1$ is $S_1$. The Baum-Welch algorithm describes the formula as:

$$\xi_t(S_1, S_2) = \frac{a_{11}(S_1)a_{12}x_2(o_t + 1)b_{t + 1}(S_2)}{P(O|A)}$$  \hspace{1cm} (2)

Furthermore $\sum_{t=1}^{T-1} \xi_t(S_1, S_2)$ is the expectation probability of all state transit from $S_1$ to $S_2$. Thus, the probability of state $S_1$ at time $t$, $\gamma_t(S_1)$, can be represented as: $\sum_{j=S_1}^{S_2} \xi_t(S_1, j)$. We can estimate the new model parameter by the following equations:

$$\pi_1^* = \gamma_1(S_1) = \sum_{j=S_1}^{S_2} \xi_t(S_1, j) = \xi_1(S_1, S_1) + \xi_1(S_1, S_2)$$  \hspace{1cm} (3)
\[ \pi_2 = \gamma(S_2) = \sum_{j=1}^{S_2} \frac{\xi_j(S_2 \cdot j)}{\xi_j(S_2 \cdot S_1)} = \xi_1(S_2 \cdot S_1) + \xi_1(S_2 \cdot S_2) \] (4)

The estimated transition probability \( a^* \) is the ratio of expected numbers of transition from \( S_1 \) to \( S_2 \) expected numbers of transition from \( S_1 \) to \( S_1 \).

\[
\frac{T-1}{\sum_{t=1}^{T-1} \gamma_t(S_1)} = \frac{1}{\sum_{t=1}^{T-1} \gamma_t(S_1)} = \frac{\text{number of times in states with page}_N^*}{\text{expected number of times in states } S_1}.
\]

When a user joins the web (log server records o1). Our approach can predict the user’s purpose automatically. Similarly, as he selects on the first tab, our process simultaneously guesses the goal and so on. We generate tests to validate our approach in this article. We are using NASA details as a benchmark (https://www.kaggle.com/souhagaa/nasa-access-log-dataset-1995). This information is a log file for the Kennedy Space Center Logs that is recorded in first service access (1 July to 31 July 1995) and second server log (August 1st to August 31st 1995). This is because it is the forecast of a platform offering links to aerospace quotations, full-text online files, and pictures and videos. Conference articles, journals, patents, study findings, photographs, films, and technical videos, STIs produced or sponsored by NASA to display user purpose, we understand it is difficult to carry every intention behind users while searching the Web. We are therefore concerned with the final discovery emblem (page) deciding if our system might discover its secret condition. The pages beyond the threshold frequency shall signify the most commonly visited page and the cumulative number of users visiting the same page.

4. Results and Discussion

Weblog file provides details on user behaviour of the website. Any of the various log types found [16,17] are SEC.gov, NASA and ClarkNet.
A web server log file is a text file published by the web server as operation. Log files gather a broad range of details about the site server knowledge queries. Examples of captured and saved data include date, time, client IP address, referrer, user agent, name of operation, server name, Network key, etc. Figure 4 displays the NASA log file example for a site server. Table 2 describes the data as before by deleting both suffixes, such as *.jpg, *.css, *.gif, and so forth. The file does not include these suffixes. After cleaning the files, it reduces the file size. It often omits incomplete URLs from first picture archives, multimedia files.

| S. No | Records (NASA 1995) | Count |
|-------|---------------------|-------|
| 1     | Records in Original log file | 3,461,613 |
| 2     | Records in Cleaned log file | 4,778,300 |
| 3     | Attributes in Original log file | 15 |
| 4     | Attributes in a Cleaned log file | 8 |

Running K-means clustering algorithm and AMD clustering algorithm for the same dataset reveal that AMD managed to outperform the k-means in any data size of the transaction. Furthermore, AMD clustering algorithm shows an incredible difference in time performance when the size of the dataset grows larger. Table 3 shows the enhanced performance of the AMD clustering algorithm when compared with K-means.

| Datasets               | Time Taken to Cluster |
|------------------------|-----------------------|
|                        | K-means Clustering Algorithm (Seconds) | AMD Clustering Algorithm (Seconds) |
| Dataset 1 (NASA)       | 0.23                   | 0.06 |
| Dataset 2 (SEC.gov)    | 0.41                   | 0.18 |
| Dataset 3 (ClarkNet)   | 0.66                   | 0.21 |
Figure 6 shows the results revealed the enhanced performance. The enhancement is in the sense of execution time taken to complete the entire process from reading the content of the log file to the finding of the most frequently accessed webpages. Here, “apollo.html” has the full AMD value that is prefixed. Figure 7 displays the NASA Kennedy Space Centre AMD sample search data.

Table 4: NASA dataset webpages and its AMD value

| Pages                | Frequency Count | Sequential Hit | AMD    |
|----------------------|-----------------|----------------|--------|
| missions.html        | 1881            | 112            | 17.27  |
| mission-sts-69.html  | 1313            | 124            | 11.07  |
| apollo.html          | 1110            | 52             | 21.53  |
| history.html         | 889             | 52             | 18.91  |
| images.html          | 845             | 87             | 11.19  |
| mission-sts-70.html  | 828             | 46             | 19.75  |
| apollo-13.html       | 783             | 74             | 11.32  |
| movies.html          | 698             | 73             | 9.36   |
| winvn.html           | 584             | 76             | 7.22   |
| stsref-toc.html      | 577             | 42             | 13.58  |
Table 5: Different datasets accuracy values

| Dataset | Number of Records | Number of Records (After Pre-process) | Accuracy |
|---------|-------------------|--------------------------------------|----------|
| DS1 – NASA | 34,61,612 | 9,33,212 | 81% |
| DS2 – SEC.gov | 13,56,862 | 3,52,893 | 79% |
| DS3 – ClarkNet | 33,28,587 | 7,63,234 | 70% |

Figure 8 shows the HMM Prediction Algorithm’s graphical representation of precision. The HMM prediction Algorithm is implemented and tested in different datasets. The experimental findings are more reliable. It offers the site users optimally predicted pages that reduce user latency.

5. Conclusion

In this article, we apply several strategies for cluster sessions depend on the navigation patterns which generate clusters containing session participants and their sub-set patterns of navigation. The research path began with the initial extraction stage for the pattern analysis of the web log file. The derived log file was pre-processed user control by utilising the number of sites the user visited and time spent on the page attributes. The efficient development of the clustering algorithm AMD shows that the
clustering algorithm K-means explodes. For pattern study, the clustered data is used to address the first considered problem of the reorganisation of the website. Through introducing the AMD clustering algorithm, we established commonly visited web pages and recent access to web pages. The study provides a new direction for recognising web pages often accessed from web server log file, suggesting another new strategy, namely HMM algorithms, for the identification of commonly accessed Web log pages with an improved time complexity. We hope to find a successful way in the future selection of parameters. We expect to find evidence of the convergence of the algorithm introduced.

References

[1] N. Lin, G. Chen, K. Zheng, and Y. Tang, “Clusoaf: A cluster-based semantic oriented analyzing framework for user behaviors in mobile learning environment,” in Human Centered Computing. Springer, 2015, pp. 340–351.

[2] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi, “Understanding individual human mobility patterns,” Nature, vol. 453, no. 7196, pp. 779–782, 2008.

[3] N. Bicocchi and M. Mamei, “Investigating ride sharing opportunities through mobility data analysis,” Pervasive and Mobile Computing, vol. 14, pp. 83–94, 2014.

[4] Wagstaff, K., Cardie, C.: Clustering with instance-level constraints. In: 17th Int’l Conf. on Machine Learning. (2000) 1103–1110.

[5] Kiri Wagstaff, Claire Cardie, Seth Rogers, and Stefan Schroedl. Constrained K-means Clustering with Background Knowledge. In Proc. Of the Eighteenth International Conference on Machine Learning, pages 577–584, 2001

[6] Eric P. Xing, Andrew Y. Ng, Michael I. Jordan, and Stuart Russell. Distance metric learning, with application to clustering with side information. In Advances in Neural Information Processing Systems 15, pages 505–512, 2003.

[7] Sisodia, Dilip & Verma, Shrish & Vyas, O. (2016). Augmented intuitive dissimilarity metric for clustering of Web user sessions. Journal of Information Science. 43. 10.1177/0165551516648259.

[8] Anupama, D.S. & Gowda, Sahana. (2015). Clustering of Web User Sessions to Maintain Occurrence of Sequence in Navigation Pattern. Procedia Computer Science. 58. 558-564. 10.1016/j.procs.2015.08.073.

[9] Sisodia, Dilip & Verma, Shrish & Vyas, O. (2016). A conglomerate relational fuzzy approach for discovering web user session clusters from web server logs. 8. 1433-1443.

[10] Bayir, Murat & Toroslu, Ismail. (2016). Finding All Maximal Paths In Web User Sessions. 15-16. 10.1145/2872518.2889351.

[11] Sisodia, Dilip. (2017). A subtractive medoids selection based fuzzy relational clustering of augmented web user sessions. Journal of Intelligent & Fuzzy Systems. 33. 2259-2268. 10.3233/JIFS-17122.

[12] Thirumoorthy, Gopalakrishnan & Sengottuvelan, P. & Bharathi, A. & Ramasamy, Lokeshkumar. (2018). An approach to webpage prediction method using variable order markov model in recommendation systems. Journal of Internet Technology.

[13] Nguyen, Da & Tan, Hanh & Pham, Duy. (2019). Improving Webpage Access Predictions Based on Sequence Prediction and PageRank Algorithm. Interdisciplinary Journal of Information, Knowledge, and Management. 14. 027-044. 10.28945/4176.

[14] Gu, Zhenyu & Jin, Chenhao & Chang, Danny & Zhang, Liqun. (2020). Predicting webpage aesthetics with heatmap entropy. Behaviour & Information Technology. 1-15. 10.1080/0144929X.2020.1717626.

[15] https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/

[16] https://www.sec.gov/dera/data/edgar-log-file-data-set.html

[17] https://www.Clarknet/data/edgar-log-file-data-set.html