IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Ingénierie des Systèmes d'Information
Vol. 27, No. 3, June, 2022, pp. 425-431
Journal homepage: http://iieta.org/journals/isi

A Framework for an Efficient Recommendation System Using Time and Fairness Constraint Based Web Usage Mining Technique

Rita Roy*, Giduturi Appa Rao

CSE Department, Gitam Deemed to be University, Andhra Pradesh 530045, India

Corresponding Author Email: Ritaroy89@gmail.com

https://doi.org/10.18280/isi.270308

Received: 14 May 2022
Accepted: 10 June 2022

Keywords:
web usage mining, clustering, constraints, fairness, cost, recommendation system

ABSTRACT

Users prefer to use various websites like Facebook, Gmail, and YouTube. We can make the system predict what pages we expect in the future and give the users what they have requested. Based on the data gathered and analyzed, we can predict the user's future navigation patterns in response to the user's requests. In order to track down users' navigational sessions, the web access logs created at a specific website are processed. Grouping the user session data is then done into clusters, where inter-cluster similarities are minimized, although the intra-cluster similarities are maximised. Recent clustering and fairness analysis research has focused on centric-based methods such as k-median and k-means clustering. We propose improved constrained based clustering (ICBC) based on fair algorithms for managing Hierarchical Agglomerative Clustering (HAC) that apply fairness constraints regardless of distance linking parameters, simplifying clustering fairness trials for HAC and intended for various protected groups compared to vanilla HAC techniques. Also, this ICBC is used to select an algorithm whose inherent bias matches a specific problem, and then to adjust the optimization criterion of any distinct algorithm to take the constraints on interpretation to improve the efficiency of clustering. We show that our proposed algorithm finds fairer clustering by evaluation on the NASA dataset by balancing the constraints of the problem.

1. INTRODUCTION

A substantial number of event logs are generated when users interact with Internet services. The system captures a considerable log file of user behavior. This allows for significant, valuable insight into application usage patterns as well as product customization and recommendations. The challenge of user session classification is particularly critical, with the goal of classifying types of sessions depending on usage patterns. The classification of user sessions is a crucial topic since it leads to essential and actionable knowledge about the program. For example, if we find out that a user prefers casual browsing to exploration, we can create a customized main page with tabs or links that make it easier to move around.

The key issue encountered with the collected data at the server-side is that the sequential navigation patterns are to be defined and separated. The problem is that certain people who visit a website because they're on the same system, use a browser to search the site [1]. Therefore, an IP address is easy to evaluate for the user. When they press the "forward" and "back" keys, it does not register them in the weblog file. This is strong proof that a page can have more than one entry. Identical data must be deleted from the data file. Sessions must be allocated to each user within a fixed time span. In this session, they must assess the time of the user. These problems can occur in web mining and pattern mining problems [2-4]. In order to have reliable predictions they should be made promptly. This has undoubtedly increased the market for recommended systems. Recommendation systems are mechanisms by which they may remove unknown or obsolete information. These have the potential to predict whether users will like an item based on their profile. Traditional methods for semi-supervised clustering are mostly used in different ways [5, 6]. A current clustering algorithm that takes into account the social and demographic factors of a region can be changed.

Some clustering algorithms can properly deal with pair-wise constraints like COPKMeans. The next approach is to use the distance metric based on the constraints. One can merge these two methods and craft a hybrid solution. Time-related problems have been accounted for in previous work by primarily recognizing the duration of a visitor's browsing and the sequential order of his/her visits (the so-called "clickstreams") [7-9]. This research is not detailed enough to identify users' perceived needs. Current applications involve the time locality of page visits in order to limit distortions of the data (e.g., on a yearly, monthly, or even daily scale). Consideration of website access time along with page preference when assessing clusters of users is vital because the time in which web users visit a page is a significant determinant for describing a specific user's needs and preferences. Many crucial circumstances are determined by (machine learning) algorithms. These range from playing promotional videos to clients, to granting home loans, to following up on criminals. It is crucial that the models are fair, free of prejudice, and that they maximize the team's efforts. Here they followed a hierarchical agglomerative process or hierarchical clustering, also known as an algorithm [10-12]. This ICBC approach differentiates various trends based on frequency of web pages. This list all of the pages in this session

known to be session representatives that are any linked patterns. Based on the web structure, every URL in the web log data is converted into tokens. The series of URLs browsed by a user for 15 minutes is viewed as a session, which might indicate a person's navigation pattern. To analyze the presence of sequences in navigation patterns, sessions from various users are aggregated using the hierarchical agglomerative clustering (HAC) approach. A session is recognized as a representative from each cluster because it has the greatest number of available pages in the sequence; other sessions in the cluster are a subset of the session representative. Such navigation patterns can help predict the most relevant sites for the user's request.

In this paper, we construct an effective method of knowledge discovery by integrating time and fairness, and we fix the shortcomings of the previous system. Our proposed work was employed to obtain the patterns that exist in the databases of website administrators in order to construct the best recommendation framework. As a result, our improved constraint-based clustering (ICBC) method is reliant on greedy Fair Hierarchical Agglomerative Clustering (FHAC) as described in Section 3, that has a notable property of linking norms as illustrated in Algorithm 2.

2. RELATED WORKS

Several strategies for clustering web user sessions have been proposed in the literature. Several strategies for clustering online user sessions have been proposed in the literature. The authors presented approaches to enhance the k-means algorithm in ref. [13] by establishing predefined centroids and using a clustering technique that yields the very same cluster for each and every run. The authors [14] described a strategy for predicting the next page-based clustering and an Artificial Neural Network. The k-means technique is used to cluster web user sessions. The accuracy of the prediction is mostly determined by the quality of the clustering. The authors [15] suggested a fuzzy clustering approach to cluster online user sessions when the sessions may correspond to more than a cluster, demonstrating improved prediction accuracy for typical user profiles.

Each user session is represented as an *n*-dimensional vector of URL references. Let *n* number of unique URLs present in the preprocessed log file are signified as

$$U = \{u_1, u_2, \ldots, u_n\} \tag{1}$$

Similarly, *m* number of user sessions discovered through preprocessing of web log data are signified as

$$S = \{s_1, s_2, \ldots, s_m\} \tag{2}$$

In Eq. (2) each user session $s_i \in S$ can be represented as:

$$S = \left\{ w_{u_1}, w_{u_2}, \ldots, w_{u_m} \right\}$$

At this point, w_{u_i} can be binary or non-binary depending on whether the URL in the session or any other feature of the URL is present or not. w_{u_i} is the time a user spends on URL item u_i for this item.

Clustering is a model for targeting similar users where it can be categorized into 2 types of clusters, namely Page Clusters and Usage Clusters. When users are combined, they exhibit identical navigational patterns. Such information is effective for inferring demographic information by searching through the history of previous users. The clustering of pages would take away pages that have shared material together. This knowledge is important for things like helping search engines and Web assistance providers find out. In both cases, it can generate permanent or dynamic HTML pages to suggest links to the user in order to fulfil the user's query data requirements. They commonly used the k-means clustering algorithm for partitioning the data. The goal of k-means is to minimize the dissimilarity between vector means in order to minimize the within-cluster sum of squares. In Eq. (3), the objective function J is shown to be based on the Euclidean distance between each data point vector x_i and its cluster centre v_j,

$$J(X,V) = \sum_{j=1}^{k} J_i(x_i, v_j) = \sum_{j=1}^{k} \left(\sum_{j=1}^{m} u_{ij} \cdot d^2(x_i, v_j) \right) \tag{3}$$

Fuzzy-C-means, Minibatch K-means and DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [16, 17] is a density-based data clustering algorithm since it finds a number of clusters from the estimated density distribution of corresponding nodes.

According to the study, several authors focused on refining the k-means clustering method for session clustering. The researchers increase cluster quality by identifying a set of clusters depending on the application field and by adjusting centroids to identify web log data consumption patterns. However, this enhancement doesn't really extract the presence of a page sequence in the navigational pattern. To define navigation patterns, extract the sequence of page occurrences. Hierarchical Agglomerative Clustering (HAC) is an algorithm used to put session representatives into groups [18, 19].

3. PROPOSED MODEL

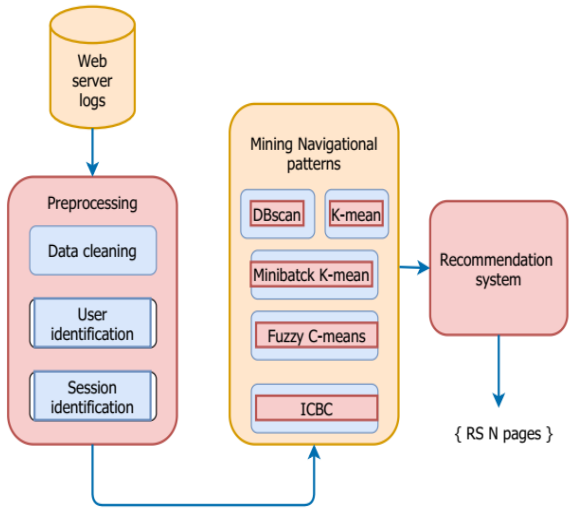

Figure 1. Proposed model of the system

The main features of the proposed system, as shown in Figure 1, are that it extracts the web log files and preprocesses log files by considering fairness and time constraints applied in various steps like data cleaning, user identification, and session identification.

The combination of several models builds this system, like preprocessing, mining navigational patterns, and finally, a recommendation model to suggest the best quality sessions with the help of clustering algorithms. The NASA 1995 dataset is used to train this system, and all five clustering algorithms are used to test it.

Data preprocessing:

The major aspect of data preprocessing is to unify the log files of the web server so that we can identify the usage patterns. We could achieve this in 3 phases: (i) Data cleaning serves as a platform to duplicate or mislabel data through the fusion of several data sources. Incorrect data is inaccurate, while outcomes and algorithms can fit properly. The major steps in the data cleaning phase may be recommended not at all, as the procedures differ from one dataset to another. Our dataset excludes redundant entries that have been logged with the file for suffixes, including jpg, gif, and tiff. (ii) The sound, streaming video, and robot files also deleted from the logs of the Web Server is normalized by specifying the various syntactically analogue URLs at the user identification phase. We refer to the time interval between the following sites as the time between visits to web pages. (iii) And then, in the identification phase of the session, the emphasis had been on the users' behaviors, originating with their entry into the web and ending with their leaving. We then identify the sessions in our log file depending on the visitor's request. If a visitor often goes to a new page, these requests may not help to figure out how they use the site. Algorithm 1 is used to show how to find sessions in webpages.

Algorithm 1 to find sessions

begin
1. Organize web pages rendering to user id and required timestamp
2. **for** every sequence of webpage
3. **do** apply fixed threshold for fragmented the web page data sequence
4. **if** we find the sequence of webpage is single
5. add the sequence web page to corresponding session list
6. **else**
7. **if** we find the shared pattern to be shared among the other subsequence
8. add subsequence to the session list
9. **else**
10. skip the sequence of web page
11. **end if**
12. **end for**
13. **end**

User Identification: In this step, the admittance of each user takes place by considering their individual IP addresses. There are many problems concerning how many users may use the same device simultaneously. Here, we are only interested in identifying valid user requests rather than specific or frequent user requests. By convention, similar multimedia content that was accessed by multiple IP addresses is removed. During this phase, host IP addresses that are assigned on the internet and can be accessed by the public are also found.

Session Identification: Content can vary depending on the specific type of content and navigational patterns of the users. There are two primary issues in this phase:

i) When a user accesses a specific page that includes the desired specific content, the web page is downloaded from the website. For every hit on the web page, whether related or not to the user, each hit is recorded in a web log file.

ii) Second, a few frequently accessed multimedia contents, like icons, buttons, banners, etc., are matched with similar sessions. To fix this flaw, an index table with labels for the multimedia content will be used to filter the results from the previous step based on the labels in the index table.

For instance, it takes an image that has the extension NBA.jpg and then collects the log files that consist of this image and similar images for other extensions. This outline will provide information such as main media content accesses, most accessed web sites, and the media accesses that distinguish sites from one another.

Mining Navigational pattern:

The frequent pattern mining algorithm is an important issue and is most relevant in the field of data mining. In this paper we propose a pattern for user navigation leveraging frequent item sets and dealing with different navigation pattern mining algorithms such as DBScan, K-mean, Minibatch K-mean and Fuzzy C-means.

Proposed Improved constraint-based clustering (ICBC):

In this paper, we compared different algorithms of navigation models with our improved constraint-based clustering (ICBC) algorithms. Thus, through the integration of time and fairness, we achieve an effective process of knowledge discovery in which we address the pitfalls of the existing system. This work was used to extract the patterns identified in website administrators' databases for the purpose of developing the right recommendation framework.

For the purpose of exploring fairness in the setting of hierarchical clustering, we recommend the inclusion of constraints and estimate the cluster quality as the set of constraints it satisfies. This quality estimation enables us to search for the unsupervised algorithm, as defined in the Algorithm 2, and its parameter settings. At first glance, with the dataset $X \in R^{n \times m}$, we officially considered vanilla HAC. Then HAC is represented by a binary tree T with a height $\left\lceil \frac{n+1}{2} \right\rceil$ on data X in which its level represents a disjoint collection that can fusion among its subclusters. Each node that is T at every stage often entails a subcluster of points. A HAC algorithm initially allows each of the n samples of X to be subclusters in a singleton and afterwards selects a combination of two sub-clusters. The lowest T level is indeed the leaves, which comprise all the n points of X. The root of T is the only X containing node/cluster. Let C_1, C_2,….C_s be another level of T subclusters. At that moment $C_1 \cup C_2 \cup, ..., \cup C_S = X$, then the Euclidean distance between adjacent points x and y is $d(x, y)$.

Case 1 (α-Proportional Fairness):

Let all the protective groups be set, in which the protected

group. Then if an X data sample is of a certain group g, then the vector g at this index has 1, otherwise 0. In addition, in cluster where $C \subset X$, and I is an index set with indices of the points in X, i.e., the optimal proportion of the representatives of group g in C is preserved when the following requirements are present for cluster C and covered group g that holds.

Case 2 (Fairness Cost (FC)):

Let the value of $HC(X)$ be the output of a certain hierarchy on X. The fairness cost with the k clusters of the HC(X) tree then calculates how close each point cluster is to the ideal g ratio for each protected group (g in F) (C_i where i = 1,2,., k).

The fairness cost can be mathematically formulated as:

$$\Sigma_{i=1}^{k}\Sigma_{g \in F}|\delta_g^{C_i} - \varphi_g|$$

Case 3 (Hierarchical Fairness Cost (HFC):

The HFC is the total of the FC in all intermediate clusters generated by the HAC mechanism. Let HC(X) = T be the total performance of such hierarchical X clusters. After this, we begin with n singleton clusters as leaf nodes for the entire tree of HAC T and combine them one at a time to arrive at a cluster (T root) containing all nodes. The intermediate fusion steps of the HAC method should then be 1 to A, and a set SA should maintain the set of related clusters linked to stage A. This is then determined by the hierarchical fairness cost as:

$$\Sigma_{i=1}^{A}\Sigma_{C \in S_i}\Sigma_{g \in F}|\delta_g^{C} - \phi_g|$$

Working of Algorithm 2: The aim of our fair algorithm is to minimise the fairness cost (FC) or the hierarchical fairness cost (HFC), where αl is any constant value to balance the cluster levels to obtain the αl proportional fairness. We first run through k iterations, then return to the center. We are still tightening the more stringent proportional constraints and loosening the less restrictive exclusion constraints, and we are steadily heading toward k-clusters. Our improved constraint-based clustering (ICBC) algorithm depends on greedy Fair Hierarchical Agglomerative Clustering (FHAC) has a remarkable feature of linkage norms shown in Algorithm 2. This Algorithm 2 is like the vanilla HAC assessment in a lot of ways, but there are some big differences. The major difference between them is in integrating fairness constraints and maintaining minimum distance among clusters. The major difference lies in allowing standards such as equal selection and fairness without negotiating completeness, which results in a fairer output tree. We consider an interesting solution for improving the runtime of Algorithm 2 by maintaining a distance matrix $R^{n \times n}$ among all the clusters by calculating the linkage norm on that condition as an input to the algorithm. In order to boost runtime, one should keep the distance between clusters equal. Moreover, dmin should be regarded as the initialization of the distance when measuring the distances between the clusters to combine into a level.

Algorithm 2: Proposed FHAC Process

$Input: X, F, k, D(.,.), Z_\alpha : R \to R, Z_\alpha : R \to R$

$Output$: Fair $HAC_{tree} T_{fair}$

1: set $C \leftarrow X$

2: $compute D_{n1,n2} = D(n1,n2), \forall (n1,n2) \in X \times X$

3: $while: |C| \geq do$

4: $P_g = 0, \forall g \in F$

5: $d_{\min} \leftarrow \infty$

6: $\alpha \leftarrow z_\alpha \left(|C| - k\right)$

7: $\beta \leftarrow z_\beta (n - |C|)$

8: for each $(c_i, c_j) \in C, s.t. c_i \neq c_j$ do

9: for each $g \in$ F do

10: $\delta_g^{ci+cj} \leftarrow \dfrac{\delta_g^{c_i}|c_i| + \delta_g^{c_j}|c_j|}{|c_i| + |c_j|}$

11: $if\ |\delta_g^{c_i + c_j} - \varphi_g| \leq \alpha$ then $P'_g = 1, else P'_g = 0$

12: end for

13: $if \Sigma_{g \in F} P'_g \geq \Sigma_{g \in F}$ Pg $then$

14: $if\ d_{\min} + \beta > D_{c_i, c_j}$ then

15: $d_{\min} \leftarrow D_{c_i, c_j}$

16: $P_g \leftarrow P'_g, \forall g \in F$

17: $(c_1^m, c_2^m) \leftarrow (c_i, c_j)$

18: end if

19: end if

20: end for

21: merge $c_1^m \leftarrow c_1^m + c_2^m$

22: update C with newly merge clusters

23: recompute $D_{c_1, c_2} = D(c_1, c_2), \forall (c_1, c_2) \in C \times C$

24: update T_{fair} with merge

25: end while

26: reture T_{fair}

As is apparent, Algorithm 2 reaches an $O(fn^3)$ symptotic time complexity analogous to vanilla HAC ($O(n^3)$) because we typically limit f for applications in the real world. In addition, the advantage of using Algorithm 2 for fair HAC is that it reduces any cost function that is necessary to ensure the hierarchical clustering mechanism is fair. We specified the fair clustering for k clusters in the previous section, and the algorithm can minimise the generic HFC for the entire tree, by suitable parameterization of Z_α and Z_β.

4. RESULTS AND DISCUSSION

In this section, we test the proposed model on the NASA dataset for the month of August 1999, which is used to validate these strategies. When the dataset is cleaned by eliminating image files, robots, and sound files, we perform the detection of users and the identification of heuristic sessions depending on the threshold time. After completing data cleaning:

Table 1 displays the sample data collection. Prior to data cleaning, and after it lists data cleaning, Table 2 shows the

amount of documents found.

We run FHAC (Algorithm 2) and approximate hyper-parameters, as described above, and then assess how fair the final clusters are for HAC single-linkage and FHAC single-linkage vanilla. In addition, since the above clustering algorithms lead to various cluster formations, the evaluation of their quality is significant. We analysed our findings based on: (i) Dunn's Index; (ii) Jaccard Index iii) The Rand Index, and iv) The Silhouette Index After this, we report the execution time in milliseconds to determine the processing time.

The below Figure 2 displays the plot graph of Dunn's index value for the cluster number (k = 1 to 5). The primary goal is to increase the distances connecting the inter-clusters and minimize the intra-clusters. The proposed ICBC shows the highest value of the Dunn index at 0.93 for cluster number k= 5, whereas the K-mediod shows poor performance with the lowest value of the Dunn index at 0.59.

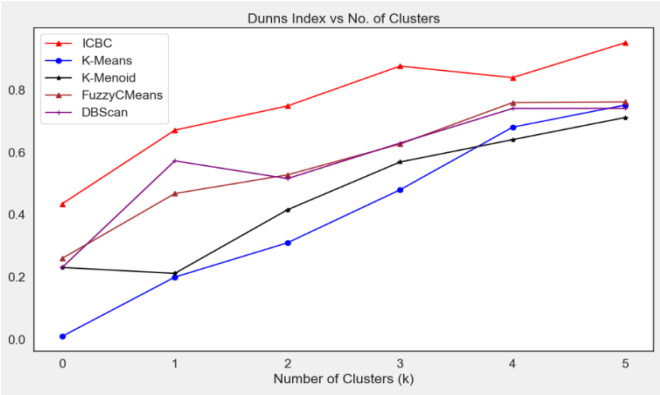

Figure 2. Comparison of Dunn's Index versus number of Clusters k

Figure 3 displays the Jaccard index value graph plot as the feature of cluster number k. The index value of Jaccard varies from 0 to 1, and the above shows the validity of the cluster. The graph shows explicitly that for this indicator, ICBC and Fuzzy Cmeans clustering (with a threshold of dissimilarity ε = 1.0) provide the max values and therefore outperform other approaches.

Figure 4 displays the Silhouette index value graph plotted as a feature of clusters k. The values of silhouette vary from -1 to 1. A value near 1 means that a rather suitable cluster applies to the data point. A value near zero shows that a data point can also be allocated to some other nearest cluster, since all clusters are equidistant. If the value is near –1, this implies that the data is incorrect and is between the clusters. Our findings show all strategies produce 0 to 1 value. ICBC and Fuzzy C-means have values equal to one and thus outperform the other approaches.

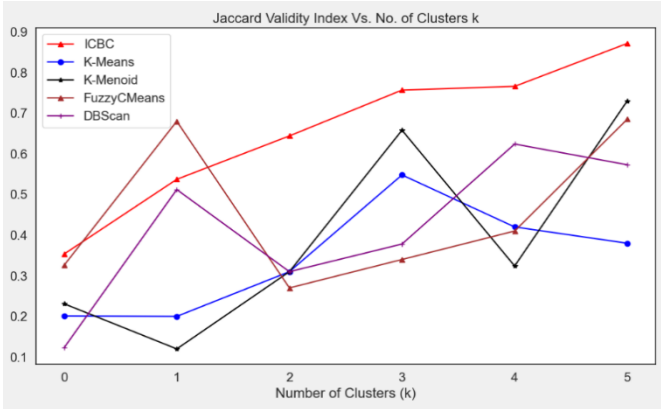

Figure 3. Comparison of Jaccard Validity Index versus number of Clusters k

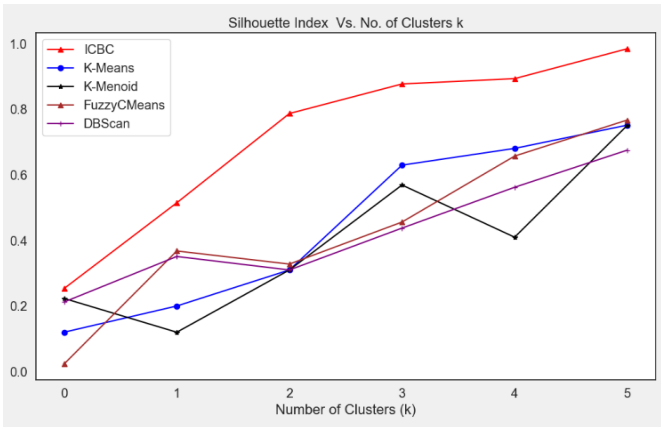

Figure 4. Comparison of Silhouette Validity Index versus number of Clusters k

Figure 5 displays the Rand validity index value as a feature of the cluster number k. The index values differ from 0 to 1. There is high consensus among the clustering groups and the normal classes for this index. The clustering of ICBC and Fuzzy c-mean gives the highest values for this index, so it is better than the others.

Table 1. Sample data set after performing data cleaning

IP Address	Session Time	URL	Status Code
154.20.320.21	24/Aug/1995:04:35:52-40012	Head/missions/shuttle2.2 /missions.html HTTP/2.0	200
125.41.115.40	23/Aug/1995:04:37:28-0400	GET/shuttle2/missions1.1/missions.html HTTP/1.1	200 8677
169.128.95.111	23/Aug/1995:04:38:32-0400	GET/shuttle12/missions.1/sts-78/missions-sts-78.html HTTP/1.0	200 59705
171.29.60.225	25/Aug/1995:04:38:46-0400	GET/history1/apolo.11-14/apolo-14.html HTTP/2.0	305 0
171..29.61.228	25/Aug/1995:04:39:58-04001	GET/shuttle22/missions34/sts-79/movies.s/imovies.html HTTP/2.0	305 0

Table 2. Number of records identified before and after data cleaning

Dataset	Period	No.of entries	No. of entries after cleaning	Identified users
Nasa weblog	August 1995	10,48,576	93980	56385

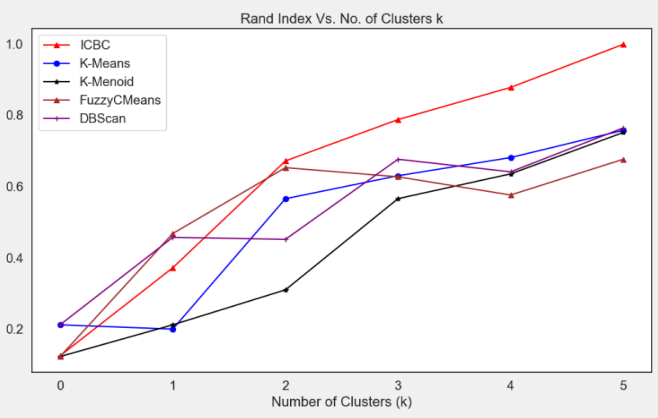

Figure 5. Comparison of Rand Validity Index versus number of Clusters *k*

Table 3 shows comparison of execution time of five Clustering Algorithms used in this current study.

Figure 6 displays the execution time of all the clustering algorithms. It is evident that ICBC outperforms in terms of execution time of 0.00746 sec as compared to other approaches.

Table 3. Comparison of execution time of clustering algorithms

Model used	Time Taken to Cluster (sec)
K-means Clustering Algorithm	0.03700
Minibatch K-means	0.256894
Fuzzy C-means	0.203484
DBScan	0.162139
Proposed ICBC	**0.00746**

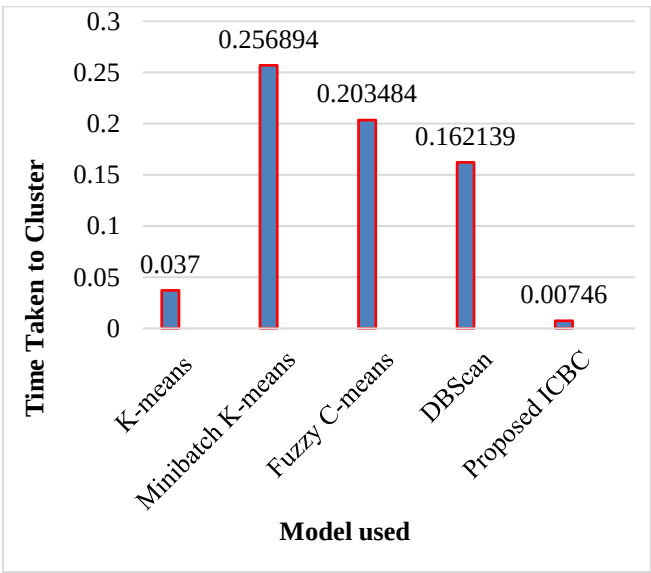

Figure 6. Execution time of clustering algorithms

5. CONCLUSIONS

Clustering sessions establish clusters of representative sessions and their subset of navigation patterns. This retrieves all viable, relevant user request pages for better prediction. This study describes alternative strategies for clustering user navigation sessions. We used several validity indexes to evaluate clusters. Documenting clustering method execution times evaluates time performance. Using collected data, proposed clustering and sequential pattern mining algorithms produce the best patterns. Online recommendations for web pages will be made using developed patterns. Our ICBC model (Algorithm 2) is independent of class linkage criteria. We show, using NASA datasets and multiple linkage criteria and evaluations, that Algorithm 2 is more fair than vanilla HAC methods, with an asymptotic time complexity of $O(fn^3)$. Future research might look towards identifying session categories using simply the initial few actions, allowing for tailored experiences within the same single visit. It might also be interesting to keep track of which demographic groups of users use the service in a certain way and then connect changes in user behavior to the release of new products and services.

REFERENCES

[1] Sisodia, D., Verma, S., Vyas, O. (2016). Augmented intuitive dissimilarity metric for clustering of Web user sessions. Journal of Information Science, 43(4): 480-491. https://doi.org/10.1177/0165551516648259

[2] Mallaiah, S., Manjula, S. (2015). Web Mining. https://doi.org/10.3850/978-981-09-4426-1

[3] Nguyen, T.T., Nguyen, B.H., Nguyen, K.P. (2013). Parallelizing the improved algorithm for frequent patterns mining problem. Intelligent Information and Database Systems, pp. 156-165. https://doi.org/10.1007/978-3-642-36546-1_17

[4] Han, J.W., Pei, J., Yan, X.F. (2004). From sequential pattern mining to structured pattern mining: A pattern-growth approach. Journal of Computer Science and Technology, 19: 257-279. https://doi.org/10.1007/BF02944897

[5] Dang, Y.Z., Xuan, Z.G., Rong, L.L., Liu, M. (2010). A novel initialization method for semi-supervised clustering. Knowledge Science, Engineering and Management, pp. 317-328. https://doi.org/10.1007/978-3-642-15280-1_30

[6] Xia, Y. (2009). A global optimization method for semi-supervised clustering. Data Min. Knowl. Discov., 18: 214-256. https://doi.org/10.1007/s10618-008-0104-3

[7] Zuo, M.H., Ou, C., Liu, H.W., Liang, Z.Y., Angelopoulos, S. (2020). Dynamic competition identification through consumers' clickstream data. Academy of Management Proceedings. https://doi.org/10.5465/AMBPP.2020.19916abstract

[8] Scholz, M. (2017). R Package clickstream - Analyzing clickstream data with Markov chains. Journal of Statistical Software, 74(4): 1-17. https://doi.org/10.18637/jss.v074.i04

[9] Huynh, H.M., Nguyen, L.T.T., Vo, B., Oplatkova, Z.K., Hong, T. (2019). Mining clickstream patterns using IDLists. 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC), 2019, pp. 2007-2012. https://doi.org/10.1109/SMC.2019.8914086

[10] Emmendorfer, L., Canuto, A. (2020). A generalized average linkage criterion for hierarchical agglomerative clustering. Applied Soft Computing, 100: 106990. https://doi.org/10.1016/j.asoc.2020.106990

[11] Aljohani, A., Edirisinghe, E.A., Lai, D.T.C. (2020). An effective and efficient constrained ward's hierarchical agglomerative clustering method. Intelligent Systems

and Applications, pp. 590-611. https://doi.org/10.1007/978-3-030-29516-5_46
[12] Bellanger, L., Coulon, A., Husi, P. (2020). PerioClust: a simple hierarchical agglomerative clustering approach including constraints. Data Analysis and Rationality in a Complex World, pp. 1-8. https://doi.org/10.1007/978-3-030-60104-1
[13] Pandian, P., Srinivasan, S. (2016). A unified model for preprocessing and clustering technique for web usage mining. J. Multiple Valued Log. Soft Comput., 26: 205-220.
[14] Snopek, J., Jelinek, I. (2022). Web access predictive models. International Conference on Computer Systems and Technologies - CompSysTech' 2005.
[15] Wan, M., Jönsson, A., Wang, C., Yang, Y. (2011). A random indexing approach for web user clustering and web prefetching. New Frontiers in Applied Data Mining pp. 40-52. https://doi.org/10.1007/978-3-642-28320-8_4
[16] Id, I., Mahdiyah, E. (2018). Modifikasi DBscan (density-based spatial clustering with noise) pada objek 3 dimensi. Jurnal Komputer Terapan, 3(1): 41-52. https://doi.org/10.13140/RG.2.2.22346.67529
[17] Lee, K.K. (2018). A comparative study between of fuzzy C-means algorithms and density based spatial clustering of applications with noise. International Journal of Engineering and Technology (UAE), 7: 131-133. https://doi.org/10.14419/ijet.v7i3.33.18592
[18] Basha, M., Kaliyamurthie, K.P. (2016). An effective scalable text pattern matching using hierarchical agglomerative clustering (HAC). International Journal of Pharma and Bio Sciences, 2016: 36-43.
[19] Kumar, I., Mishra, M.K., Mishra, R.K. (2021). Performance analysis of NOMA downlink for next-generation 5G network with statistical channel state information. Ingénierie des Systèmes d'Information, 26(4): 417-423. https://doi.org/10.18280/isi.260410