International Journal of Informatics, Information System and Computer Engineering

An Efficient Fuzzy Clustering Algorithm for Mining User Session Clusters on Web Log Data

Moksud Alam Mallik[1,2], Nurul Fariza Zulkurnain[1]*

[1]International Islamic University Malaysia, Kuala Lumpur, Malaysia.
[2]VNR Vignana Jyothi Institute of Engineering & Technology, Hyderabad, India.
*Corresponding Email: [1]alammallik_m@vnrvjiet.in

ABSTRACTS

Data mining is extremely vital to get important information from the web. Additionally, web usage mining (WUM) is essential for companies. WUM permits organizations to create rich information related to the eventual fate of their commercial capacity. The utilization of data that is assembled by Web Usage Mining gives the organizations the capacity to deliver results more compelling to their organizations and expanding of sales. Client access patterns can be mined from web access log information using Web Usage Mining (WUM) techniques. Because there are so many end-user sessions and URL resources, the size of web user session data is enormous. Human communications and non-deterministic browsing patterns increment equivocalness and dubiousness of client session information. The fuzzy set-based approach can solve most of the challenges listed above. This paper proposes an efficient Fuzzy Clustering algorithm for mining client session clusters from web access log information to find the groups of client profiles. In addition, the methodologies to preprocess the net log data as well as data cleanup client identification and session identification are going to be mentioned. This incorporates the strategy to do include choice (or dimensionality decrease) and meeting weight task assignments.

ARTICLE INFO

Article History:
Received 18 Dec 2021
Revised 20 Dec 2021
Accepted 25 Dec 2021
Available online 26 Dec 2021

Keywords:
Data Mining, Web usage mining (WUM), Data Preprocessing, Fuzzy Clustering.

1. INTRODUCTION

Data mining, the extraction of hid judicious information from immense informational collections, is a staggering new development with the phenomenal potential to help associations revolve around the fundamental information in their data stockrooms. Information mining instruments anticipate future examples and work on them, allowing associations to make proactive data-driven decisions. Utilizing a blend of AI, measurable investigation, demonstrating methods, and data set innovation, information mining discovers designs and unobtrusive connections in information and construes decisions that permit the forecast of future outcomes. Data mining (information disclosure from information) is the extraction of fascinating for example non-immaterial, verifiable, ahead-of-time dark, and conceivably important examples or information from a huge proportion of information. It changes locally very well and may be alluded to as information revelation (mining) in data sets (KDD), information, extraction, Information, design investigation, and so forth (Han et al., 2012; Zahid et al., 2011; Cooley et al., 1997). Web mining is defined as the disclosure and evaluation of useful data from the World Wide Web in a broad sense. There are two sections to web mining: Web content mining and web utilization mining are two types of web mining.

Web use mining is the automated disclosure of user access patterns from Web servers. Every business collects a significant amount of data on a daily basis in its operations. Web servers generate this information, which is saved in server access logs. Examining server access log data helps the organization to focus on lifetime estimation of customers, showcasing strategies for products, effective promotional campaigns, etc. It also helps in rebuilding websites to represent the organization and promote their products and services in a better way in WWW. Web mining is by and large isolated into two parts. The first part is secondary in space; it converts web data into an appropriate exchange structure. This combines exchange ID preparation and information inclusion. The subsequent part is space self-sufficient applications like general information mining and example coordinating with methods like clustering (Cooley et al., 1997).

Preprocessing, information extraction, and examination outcomes are all included in WUM. The preprocessing stage of Web-use mining aims to convert unprocessed web log data into a large number of customer profiles. Each of these profiles receives a plan or a number of URLs related to a customer session. The preprocessing stage in Web-use mining changes the harsh snap stream data into a get-together of customer profiles. Each of these forms contains a set of URLs that correspond to a client session. For different preprocessing activities, such as data fusion and cleaning, user and session identification, and so on, several algorithms and heuristic methods are used. Convergence of log files from several web servers is referred to as data fusion. Data cleaning incorporates assignments, for example, eliminating unnecessary references to inserted objects, style documents, illustrations, or sound records, and disposing of references because of bug routes. By doing away with an undesirable substance like this we can lessen the size of the input file and make the mining

errand efficient. So, during preprocessing we will clean the data, identify the user by using the IP address and identify the user session by using time-oriented heuristics. We can assign weight to URLs based on the number of times they are accessed in different sessions also weight can be assigned to a session according to the number of URLs present in it (Zahid et al., 2011; Ansari et al., 2012; Babuy et al., 2011). When user sessions are found we can utilize them for clustering. Little sessions will be removed because it shows disturbances in the data. Rather than straightforwardly removing it, we can utilize a fuzzy set_theoretic way to deal with it. Direct elimination of minimal estimated sessions may achieve a loss of a gigantic proportion of data. So, we can relegate weight to all sessions considering the number of URLs got to by the session (See Figure 1).

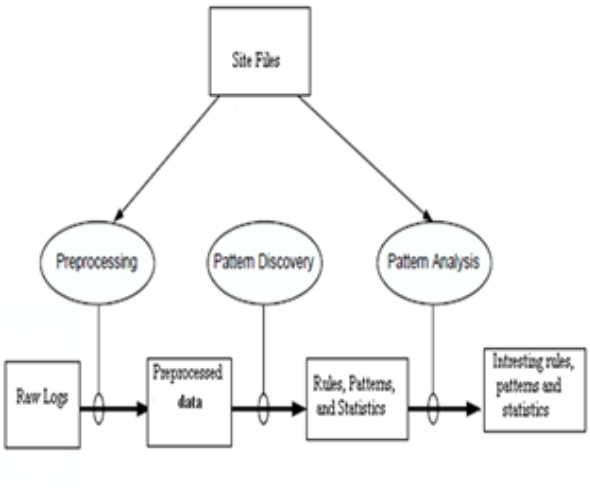

Figure 1. Structure of web usage mining

After this, we can apply the fuzzy clustering algorithm to recognize user session clusters. Fuzzy membership is promoted by fuzzy clustering. In this case, a single informational index can be used by many groups. It suggests that one informational collection can find a place with a few bunches all the while. Every informational index will have a degree of enrollment in each group; Some groups will have a high level of participation, while others will have a low level of enrollment. The value of participation will range from zero to one. The total assessment of the participation of one meeting to each bunch of habitats will be one. Data fuzzy clustering ought to oversee fit reality. For instance, if an informational index is on the limit between at least two bunches fluffy grouping will give it halfway participation among bunches (Bezdek et al., 1984). In fuzzy clustering, each datum point has relegated participation worth to every one of the clusters. If the membership value is zero the data is not a piece of that cluster. No zero value shows that the data is attached to that cluster. Membership value will be always between zero and one. Here we can discover similar user access patterns i. e. same URL patterns by applying the Fuzzy clustering algorithm. The output of this step will be separate user session clusters it (Zahid et al., 2011; Ansari et al., 2012; Babuy et al., 2011).

The Literature Review is found in Section 2, the proposed algorithm is found in Section 3, the test results are found in Section 4, and the conclusion and future improvements to this research are found in Section 5.

2. LITERATURE SURVEY

Digitized information is easy to capture and storing it is very cheap. So gigantic measure of data has been put

away in distinctive sorts of databases and other types of storage. The data storage frequency is developing at an exceptional rate. This developing data is amassed in various huge data storages. This sort of circumstance requires intense apparatuses to grasp knowledge from this ocean of information. With the exceptionally high development of data sources open on the World Wide Web, it has wound up continuously indispensable for clients to use customized instruments in finding the needed information resources, and to follow and dissect their utilization designs. So, there is a necessity to create server-side and client-side tools that mine knowledge adequately (Cooley et al., 1997). Web usage mining is the revelation of client access designs from web servers. How clients are getting to a webpage is critical to building the use of the site by clients. There are three steps to it. Preprocessing, pattern extraction, and examination of the results. Different forms of sounds are removed during the preprocessing stage. The user and session identification process will be completed in this stage. A wide variety of pattern extraction techniques are available like clustering, path analysis, etc based on the needs of the analyst. Once web usage patterns are discovered there are different types of techniques and tools to analyze and understand them. A gigantic amount of unessential data is available in input web access logs. Many user sessions and URL resources makes the dimension of web-user session data very high. Human interactions and nondeterministic browsing patterns increase the ambiguity and vagueness of user session data. The World Wide Web is a massive, dynamic data source that is both architecturally complex and constantly evolving. As a result, it is a fertile ground for data mining and web mining. Using various information mining methodologies, web mining can be utilized to extract valuable information from the internet. The majority of web information is unlabeled, dispersed, heterogeneous, semi-coordinated, time-moving, and multi-dimensional. The following categories of data can be found on the internet:

(i) The substance of real Web pages

(ii) Intra-page constructions of the website pages.

(iii) Inter-page structures decide linkage structures between website pages.

(iv) We use information depicting web

(v) User profiles incorporate demographic and enrolment data about users.

Web Usage Mining (WUM) takes a gander at the aftereffects of customer relationships with a web worker, including weblogs, click streams, and informational index trades at a website or a social event of related areas. WUM performs three guideline steps: preprocessing, design extraction, and results in examination it (Zahid et al., 2011; Ansari et al., 2012; Babuy et al., 2011). Giovanna use a LODAP (Log Data Preprocessor) tool to do preprocessing of web log data (Castellano et al., 2007; Nasraoui et al., 2000). To investigate Web log information, we use LODAP, a product device that cycles web access information to eliminate immaterial log passages, recognize gets made by clients,

and gather client gets into client meetings. Every client meeting contains access data (number of visits, season of visit, and so on) about the pages seen by a client; as a result, it depicts that client's navigational behavior. The term "user identification" refers to the process of identifying unique users from online log data. Generally, the log document in Extended Common Log design gives simply the PC's IP address and the client specialist. User registration-required websites will include additional user login information that can be utilized to identify users. Each IP address will be treated as a user if the user login information is not available. After this, we have to recognize user sessions. Here we will partition the web log data file into diverse parts known as user sessions. Every session is considered a single visit to a website. Identification of client meetings from the weblog record is a convoluted errand. This information can be used as a contribution to an assortment of information mining calculations it (Zahid et al., 2011; Ansari et al., 2012; Babuy et al., 2011). For clustering user sessions, we employ the Fuzzy c-Means clustering technique. Here we need to randomly select initial cluster centers. The similarity measure is done based on the page visit time using fuzzy intersection and union. Even after preprocessing noise is still present in the web log data.

Olfa defined the similarity between user sessions where compute preprocessing and segmentation of web log data into sessions. Preprocessing of web log data and cluster user sessions can achieve using the fuzzy clustering technique. This will affect the clustering result and similarity measures (Olfa Nasraoui et al., 2008).

Zahid explains an existing web usage mining framework. It uses the fuzzy set-theoretic approach in preprocessing and in clustering. It improves mining results when compared with the crisp approach in preprocessing and clustering. Because the fuzzy approach matches more with a real-world scenario. It is using the fuzzy c-means algorithm for clustering (Zahid et al., 2011; Ansari et al., 2011).

Using a fuzzy c-Means clustering technique, Castellano hopes to divide website users into different groups and generate session clusters. Preprocessing should remove noise up to maximum because it will affect remaining operations like session identification and clustering the sessions. The fuzzy set-based approach can solve most of the challenges listed above. FCM needs an initial random selection of clusters. This work focuses on designing "an efficient Fuzzy Clustering Algorithm for Mining User Session Clusters from Web Access Log Data". It improves the quality of clusters discovered (Castellano et al., 2006).

3. METHOD: PROPOSED SYSTEM

Here a new efficient fuzzy clustering algorithm that can proficiently mine client session clusters from web access log information is proposed. The calculation manages the least of medians while choosing group focuses. The strategy lessens mean squared mistakes and takes out the impact of anomalies.

3.1. Input Data

The essential information sources utilized in Web utilization mining are the worker log documents, which incorporate Web server access logs and application server logs. The input server log data is downloaded from the site https://filewatch. net. Filewatcher is a FTP search engine that monitors more than two billion files on more than 5,000 FTP servers. The downloaded file name is "pa. sanitized access. 20070109. gz". A sample server log file entry is given below (Table 1).

Table 1. Sample server log file entry

1168300919. 015	The time of the request
1781	The elapsed time for HTTP request
17. 219. 121. 198	IP Address of the client
TCP_MISS/200	HTTP reply status code
1333	bytes send to the server in response to the request
GET	the requested action
http://www. quiethits. com/hitsurfer. php - DIRECT/204. 92. 87. 134	URI of the item being mentioned, customer client name, the hostname of the machine where we got the solicitation,
text/html	content-type of the object.

3.2. Data Mining

Every hour, well-known websites generate gigabytes of online log data. Managing such massive records is a difficult task. Log record sizes can be reduced by performing information cleansing, allowing mining assignments to be lifted. When a user requests for a web page enters or clicks on a URL usually a single request will cause several URLs to be generated like figures, scripts, etc. So all URLs with a graphic extension should be removed. Web robots are also identified and their queries are removed during data cleaning. In weblog data, a web robot (also called as Web Wanderers, Crawlers, or Spiders) generates numerous request lines automatically. Robot's request is unwanted because it is not generated by the user, it is generated by the machine. So, we should remove robot requests as removing them will increase the accuracy of clustering results. Here we employed two methods for extracting robot requests. The first one is checking for an entry in "robots. txt" in

web log data and the second one is removing HEAD requests (Zahid et al., 2011; Ansari et al., 2012; Babuy et al., 2011). Next is the removal of URLs with query strings. Normally URL with query strings is used for requesting extra details from within the web page within the same session. Since they are unnecessary, we will remove them as well (Zahid et al., 2011; Ansari et al., 2012; Babuy et al., 2011).

The input file is 30. 6MB in size and has 2,06,914 entries. After removing URLs with graphic contents, the log file has 72,498 entries which are almost one third of the input file. After removing the web robot request, we have 72,305 entries. After removing URLs with query string,we have 59,054 entries in the log file. Then we will encrypt IP Address to hide the user's identity and to have ease in future processing and the IP address will be put away in a map with its encoded id. Furthermore, each URL will be appointed a unique number and it will be put away in a URL map along with its number (Zahid et al., 2011; Ansari et al., 2012; Babuy et al., 2011). The Data cleaning algorithm is demonstrated in the following scheme:

1. Step 1: Remove each line of the input file one by one.
2. Step 2: Remove all URLs with suffixes recorded in the above suffix list.
3. Step 3: Remove all URLs produced by web robots.
4. Step 4: Remove URLs with query strings.
5. Step 5: Take out the IP address and store it on a map.
6. Step 6: Code URL with URL number and store it on a map.
7. Step 7: Sort each line based on the IP Address encryption code.
8. Step 8: Print in the required fields to a yield file.

The output file after applying the above algorithm will be as shown in Table 2. The output file is sorted in ascending order based on the encoded value of the IP Address (Table 2).

Table 2. Output file after data cleaning

IP	Time	Elapsed Time	Bytes	URL
IP1	1168300931. 828	142	1599	1
IP1	1168300935. 244	501	1617	2
IP1	1168300936. 604	1	1617	3
IP1	1168300941. 345	2	1593	4
IP1	1168300957. 585	186	1585	6
IP1	1168300985. 665	145	1563	10

3.3. User Identification

After cleaning input web log data, we can distinguish users. Since the log file doesn't contain user login information, we consider each IP as a user. Next, we separate all solicitations identifying with the individual user. The algorithm for user identification is shown in the following scheme (Zahid et al., 2011; Ansari et al., 2012; Babuy et al., 2011).

Step 1: Split every line in the input file into obliged fields.

Step 2: Store it(i. e. obliged fields) in a Map M1 with IP Address as the key and another Map M2 as the worth. Key of the Map M2 is the time and worth is whatever is left of the fields.

Step 3: Sort the internal map m2 considering the time key.

Step 4: Print the content of the map M1 to the yield record.

The organization of the yield document produced after user identification is shown in Table 3.

3.4. Session Identification

Client Session distinguishing proof is the technique of dividing the customer activity log of each customer into sessions, each addressing alone visit to the site. Sites without client verification data generally depend on heuristic strategies for sessionization. The sessionization heuristic guides in isolating the genuine game plan of exercises performed by one customer in one visit to the site. Keeping in mind the end goal to recognize client sessions we can try different things with two distinctive Time-Oriented Heuristics (TOH) as portrayed underneath (Zahid et al., 2011; Ansari et al., 2012; Babuy et al., 2011):

TOH1: The time term of a session should not surpass a limit α. Let the timestamp of the main URL demand, in a session be, T1. If another URL asks for a session with timestamp Ti it is allotted to the same session if and only if Ti-T1≤ α. The principal URL asking for with timestamp bigger than T1 +α is taken as the first request of the following session (Zahid et al., 2011; Ansari et al., 2012; Babuy et al., 2011):

1. Step 1: The given steps ought to be finished for every line in the information input file.
2. Step 2: If the Line contains User Id, then UserId =User Id of the line.
3. Step 3: Print Line to output file under this User Id and the first session of same User Id.
4. Step 4: In case that L is the first accessed log of the user then T1 = Line. time else T2 = Line. time.
5. Step 5: If T2-T1≤ α at that point print Line under the same session to the file.
6. Step 6: If it is not as in the previous step i. e. Step 5 then output

User Id and corresponding line under a new session, T1 = Line. time.

Detailed information is shown in Table 3.

Table 3. Algorithm to create User Sessions taking into account TOH1.

User	Time	Elapsed Time	Bytes	URL
IP1	1168300931. 828	142	1599	1
	1168300935. 244	501	1617	2
	1168300936. 604	1	1617	3
	. . .	. .	. .	. .
IP2	1168300953. 645	648	260	5
	1168300990. 665	143	260	14

TOH2: The time spent on a page visit should not surpass a limit *a*. Let a URL that is most recently given to a session having a timestamp T_i. The next URL's request fits in with the same session if and only if T_{i+1}-$T_i \leq \alpha$ where T_{i+1} is the timestamp of the new URL's request. This URL is now the first of the following session (Zahid et al., 2011; Ansari et al., 2012; Babuy et al., 2011). In our implementation for the interim, we are utilizing TOH1. We have chosen 30 minutes as the estimation of the limit time. The algorithm for user session identification is shown in Table 4 and the output file of session identification are shown in Table 4.

3.5. Dimensionality Reduction

Removing to separate the logs references to low bolster URLs (i. e. that are not bolstered by a predetermined number of user sessions) can give a powerful dimensionality decrease system while enhancing clustering. To implement this, we are removing URLs that occur only once (Zahid et al., 2011; Ansari et al., 2012; Babuy et al., 2011) (see Table 4).

Table 4: Output file of Session Identification

User Session	Time	Elapsed Time	Bytes	URL
IP1S1	1168300931. 828	142	1599	1
	1168300935. 244	501	1617	2
	1168300936. 604	1	1617	3
IP1S2	1168302738. 407	81	1623	482
	1168302745. 477	138	1559	483
	. . .	. .	. .	. .
IP2S49	1168300953. 645	648	260	5
	. . .	. .	. .	. .

3.6. Session weight assignment

The session files can be divided for the clustering process in order to remove small sessions with the purpose of removing variation from the data. In any event, deleting these little measured sessions directly may result in the loss of a vital measure of information, especially if the number of these small sessions is significant. Here we allot weights to every one of these sessions considering the number of URLs got to by the sessions. Session weight assignment is done based on the following equation (Zahid et al., 2011; Ansari et al., 2012; Babuy et al., 2011).

W_{s_i}=0,if$|s_i| \leq 1$

W_{s_i}=1, if$|s_i| \geq 1$

where $|s_i|$ is the number of URLs accessed in a particular session.

3.7. Development of user session matrix

Here we represent sessions using a matrix. Every row denotes a session, and the column denotes a URL. If a URL arrives in a session, then the entry for that URL in the specific session will be more prominent than zero. It will be many events of that URL in that session. If URL is not present, then that entry will be zero. Sessions are referred to by utilizing a sparse matrix in row-major form. It reduces processing time up to a great extent. After all, we are dividing to standardize the session matrix for every column by its greatest value (Zahid et al., 2011; Ansari et al., 2012; Babuy et al., 2011).

For fuzzy clustering structures, the Fuzzy C-Means technique is commonly employed nowadays. So, in order to compare our new algorithm against the previous system, we used FCM (Zahid et al., 2011; Ansari et al., 2012; Babuy et al., 2011; Bezdek et al., 1984).

3.8. Implementation of Proposed System.

The suggested system can be implemented as a fast Fuzzy clustering technique for mining user session clusters from web log data, as described in section 5 titled "Session Clustering." The following is the primary part of the processing:

At first, we will take one meeting say s1 and discover the distance between this meeting to every other meeting (say s2; s3; s4; …; sn)multiplied by the enrollment capacity of s1 to bunch focus 1(v1). Next, we will sort these qualities into rising requests and take the middle. The above step will be done for all sessions s_1; s_2; s_3; s_4; …; s_n. Now these medians obtained from the above steps will be sorted and the least value will be taken. The session relating to the least worth will be taken as the main group community in this round. All above advances will be proceeded for bunch focus 2 up to group focus c(v1; v2; v3; …; vc). In this way, we will get new arrangements of bunch focuses in one round. New group communities will be determined up to a particular number of rounds till we get ideal bunch habitats.

3.9. Modification in Proposed System.

Here for every cluster center, we will be selecting the smallest value of medians. However, the issue is that abruptly we are getting the same smallest median in each iteration. So, in each cycle, we are getting the same cluster center repeatedly. So, we rolled out a little improvement in this algorithm. Instead of selecting the least median in each round, we will choose the smallest median in the first round, the second smallest median in the second round, the third smallest median in the third round, and so on. By actualizing in this manner, we are demonstrating indicators of progress in the suggested algorithm's execution, which is superior to FCM.

3.9.1. Fuzzy Membership function

Expect to be X = {x1; x2; :::; xm} is the arrangement of information focuses or sessions. Each point is a vector of the structure I = 1… m , xi = {xi1; xi2; :::; xin}. Let V ={v1; v2; :::; vc} is a bunch of n dimensional vectors compares to c group habitats and each bunch place is a vector of the structure 8j = 1:::n , vj = {v1j; v2j; :::; vnj}. Let uij addresses enrollment of information point(or meeting) xi in bunch j. The m×c enrollment framework U = [uij] shows portion of sessions to different bunch communities. It fulfills following models.

$\sum_{j=1}^{c} u_{ij} = 1;\ \forall i = 1 \ldots m$

$0<\sum_{i=1}^{m} u_{ij}<m,\ \forall j = 1 \ldots c$

The participation esteem is determined by utilizing the accompanying equation (Bezdek et al., 1984).

$$Uij=\frac{\frac{1}{d_{ij(xi,vj)}^{2}}^{\left(\frac{1}{m-1}\right)}}{\sum_{k=1}^{c}\frac{1}{d_{ij\,(xi,vk)}^{2}}^{\frac{1}{(m-1)}}} \ldots\ldots\ldots\ldots\ldots\ldots(1)$$

The initial cluster centers are chosen at random from the available sessions. Then, using the equation for u_{ij}, the membership value of each cluster is calculated. The following equation can be used to calculate the Euclidean distance between various data points and cluster centers (Malik et al., 2021; Bezdek et al., 1984).

$$d_{ij(xi,vj)}^{2}=\sum_{k=1}^{n} d_{ik\left(x_{k,}^{i} v_{k}^{j}\right)}^{2} \ldots\ldots\ldots\ldots(2)$$

Where n is the number of dimensions of each data point, $x_{k,}^{i}$ is the value of k^{th} dimension of x_i , and v_k^j is the value of k^{th} dimension of v_j which is the jth cluster center.

3.9.2. Cluster Center calculation

The following formula calculates new cluster centers in each iteration step:

Di = Median{$(d_{ij}(s_k$ - $s_i)$ *$u_{ij})$}; ∀i ≠ k; k = 1…n

p = Argmin{(Di:n); ∀i = 1…n} vj = sp

At first, we will take one meeting say s1 and discover the distance between this meeting to every other session (say s2; s3; s4; …; sn) duplicated by the enrollment capacity of s1 to group focus 1(v1). Next, we will sort these qualities in climbing requests and take the middle. The above advance will be accomplished for all meetings s1; s2; s3; s4; …; sn. Presently from these medians got from above advances least worth will be taken. The

meeting relating to the least worth will be taken as the primary group place in this round . All above advances will be proceeded for bunch focus 2 up to group focus c(v1; v2; v3; ::::vc). In this manner, we will get new arrangements of the group focuses in one round. New group habitats will be determined up to a particular number of rounds till we get ideal bunch communities.

3.9.3. Objective function calculation

The target work is utilized to quantify the exhibition of the grouping calculation. The bunch habitats which are having less incentive for the target capacity will give smaller groups or better grouping results. The presentation record is determined utilizing the accompanying target work:

$$Oi=\sum_{j=1}^{c}(uij * dij(vj - si)) \ldots\ldots\ldots (3)$$

3.9.4. Algorithm Termination

The means in these areas will have proceeded till a predetermined number of steps or till we get the base incentive for the goal work. The relating worth of the group places will be taken as last.

4. RESULTS AND DISCUSSION

For both FCM and the proposed Fuzzy algorithm (FCLM), we will give similar data information. That is, we will give a similar basic bunch habitat and ascertain the Xie-Beni Index, Partition Coefficient, FS Index, Deviation, Compactness, and Separation of the group focuses overall. Likewise, we will process the fluffy smallness and fluffy deviation of individual group habitats. After that, we will look at file esteems obtained by both FCM and the proposed calculation. The calculation which is getting less regard for legitimacy record performs preferred other over Partition coefficient. For parcel coefficient, the calculation which gets a higher worth performs better (Its most outrageous worth is "one"). Detailed results are shown in Figures 2-10.

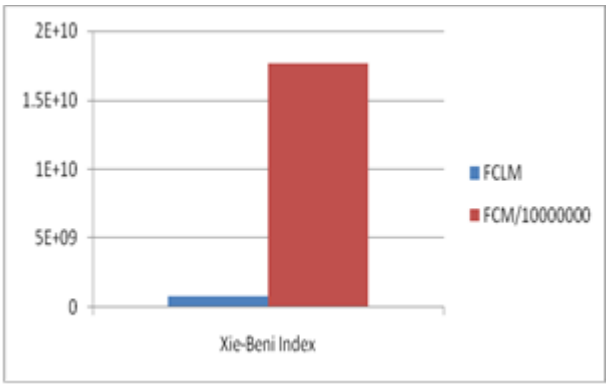

Figure 2. Effect of Xie-Beni index on value

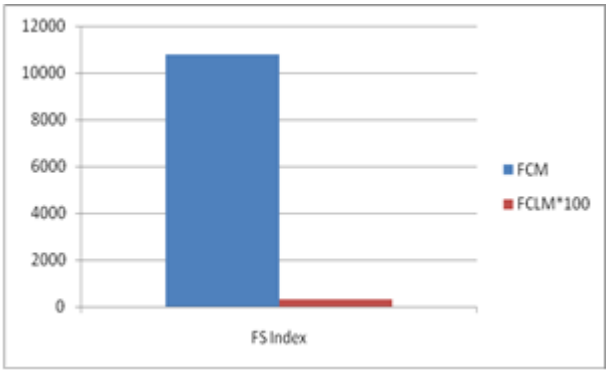

Figure 3. Effect of FS index on value

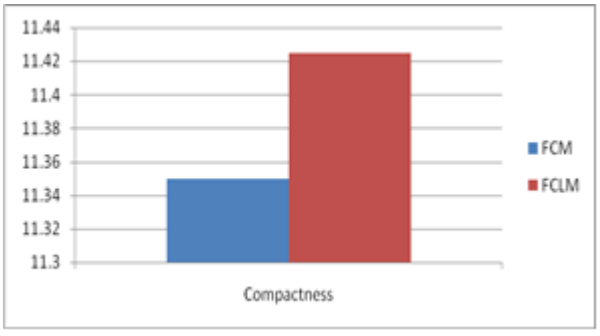

Figure 4. Compactness

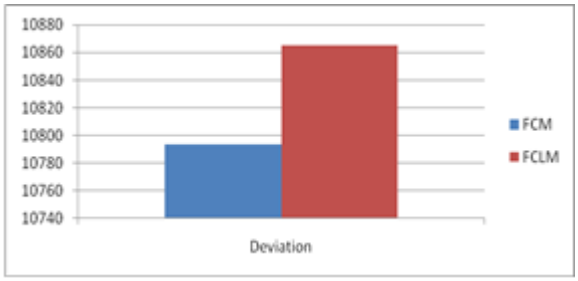

Figure 5. Deviation

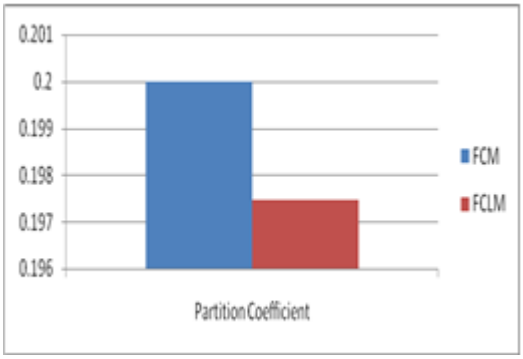

Figure 6. Partition Coefficient

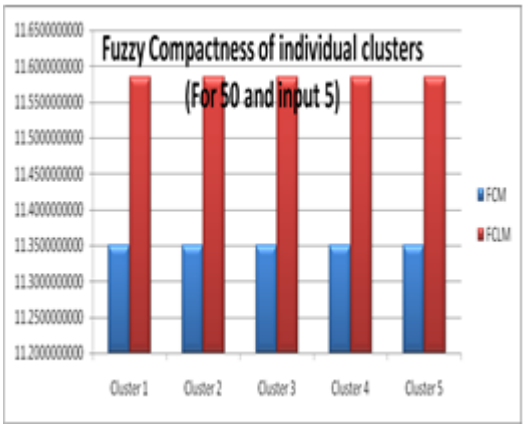

Figure 7. Fuzzy Compactness of indvidual clusters

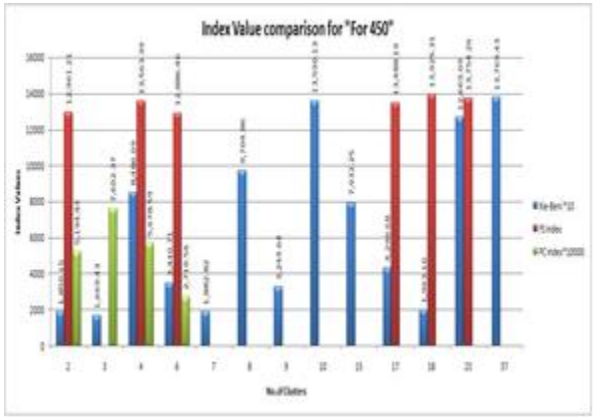

Figure 8. Index value comparison for "for 450"

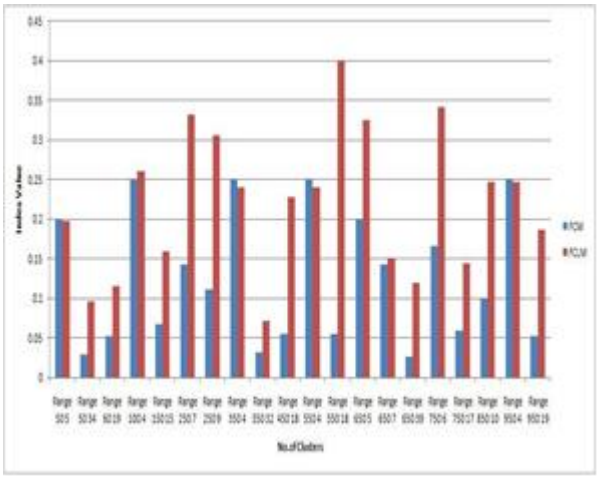

Figure 9. No of clusters

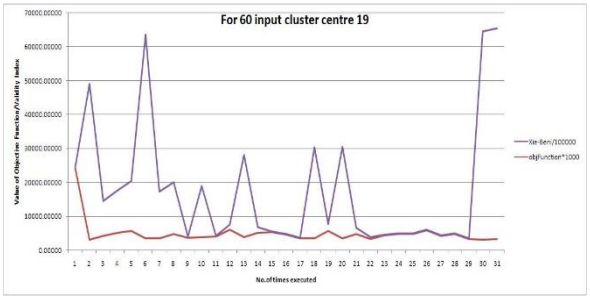

Figure 10. For 60 input cluster centre 19

5. CONCLUSION

The proposed fuzzy clustering algorithm further develops grouping. It decreases the impact of mean squared blunder and disposes of the impact of exceptions and consequently the clamor. So, we are improving bunching arrangements than in the fuzzy c-means algorithm. With better informational collections we can improve arrangements in our calculation. Both FCM and proposed fuzzy clustering algorithm exhibitions are affected by starting group place decisions. To get away from this adverse consequence we can apply the mountain thickness capacity to pick basic group habitats. So we will get fitting beginning group habitats. In

preprocessing additionally, we need to do many enhancements like applying new techniques to channel robot demands.

REFERENCES

Ansari, Z., Azeem, M. F., Babu, A. V., & Ahmed, W. (2015). A fuzzy approach for feature evaluation and dimensionality reduction to improve the quality of web usage mining results. *arXiv preprint arXiv:1509.00690*.

Ansari, Z., Azeem, M. F., Babu, A. V., & Ahmed, W. (2015). A fuzzy clustering based approach for mining usage profiles from web log data. *arXiv preprint arXiv:1509.00693*.

Ansari, Z., Babuy, A. V., Ahmed, W., & Azeemz, M. F. (2011, September). A fuzzy set theoretic approach to discover user sessions from web navigational data. In *2011 IEEE Recent Advances in Intelligent Computational Systems* (pp. 879-884). ieee.

Bezdek, J. C., Ehrlich, R., & Full, W. (1984). FCM: The fuzzy c-means clustering algorithm. *Computers & geosciences*, *10*(2-3), 191-203.

Castellano, G., Fanelli, A. M., & Torsello, M. A. (2006, September). Mining usage profiles from access data using fuzzy clustering. In *The 6th WSEAS international conference on simulation, modelling and optimization, Portugal*.

Castellano, G., Fanelli, A. M., Mencar, C., & Torsello, M. A. (2007, November). Similarity-based fuzzy clustering for user profiling. In *2007 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology-Workshops* (pp. 75-78). IEEE.

Castellano, G., Mesto, F., Minunno, M., & Torsello, M. A. (2007, July). Web user profiling using fuzzy clustering. In *International Workshop on Fuzzy Logic and Applications* (pp. 94-101). Springer, Berlin, Heidelberg.

Cooley, R., Mobasher, B., & Srivastava, J. (1997, November). Web mining: Information and pattern discovery on the world wide web. In *Proceedings ninth IEEE international conference on tools with artificial intelligence* (pp. 558-567). IEEE.

Han, J., Kamber, M., & Pei, J. (2012). Data Mining Concepts and Techniques 3rd Edition Vol. 3rd.

Mallik, M. A., Zulkurnain, N. F., Nizamuddin, M. K., & Aboosalih, K. C. (2021, February). An Efficient Fuzzy C-Least Median Clustering Algorithm. In *IOP Conference Series: Materials Science and Engineering* (Vol. 1070, No. 1, p. 012050). IOP Publishing.

Nasraoui, O., Frigui, H., Krishnapuram, R., & Joshi, A. (2000). Extracting web user profiles using relational competitive fuzzy clustering. *International journal on artificial intelligence tools*, *9*(04), 509-526.