DOI 10.29042/2017-2018-2024

Helix Vol. 7(5): 2018-2024

Clustering of users on web log data using Optimized CURE Clustering

[*1]P. Dhanalakshmi, [2]K. Ramani, [3]B. Eswara reddy

[1, 2] SVEC, A.Rangampet, [3]JNTU Kalikiri

Received: 6th June 2017, Accepted: 15th June 2017, Published : 1st Septemebr 2017

Abstract:

Web usage mining is one of the essential frameworks to find domain knowledge from interaction of users with the web. This domain knowledge is used for effective management of predictive websites, creation of adaptive websites, enhancing business and web services, personalization and so on. In non-profitable organization's website it is difficult to identify who are users, what information they need, and their interests change with time. Web usage mining based on log data provides a solution to this problem. The proposed work focuses on clustering of web users on web log data by developing an optimized clustering algorithm. The performance of web usage mining is also compared based on k-means, DBSCAN and CURE with proposed algorithm.

Keywords: clustering, K-Means, Density based clustering, Optimized clusters, Heap tree, KD tree, CURE Clustering, Manhattan distance measure, weblog, web usage mining.

1. Introduction:

Data Mining helps to extract only relevant information from these large repositories. Web is a huge repository of text documents and multimedia data. Mining useful data from the web is known as web mining and it may include analysis of web content, structure and also understanding usage of web . Usage of internet has magnified number of web users enormously increasing web log data. To understand the web user behavior one has to study web log data. It has three stages such as preprocessing, identifying usage patterns and analyzing the obtained patterns. Web server accounting all users' actions of the web site as web servers Logs. The majority of the log files have Common log file (CLF) format and each log contains IP Address, user name, date, time stamp, visited URL, Request type. web log mining is a process of analyzing and discovering user access patterns and their behavior using weblog data which is available may be in proxy, web servers or client browsers. Clustering of these web users in E-commerce sites becomes an essential approach for many applications: web site personalization and structure optimization, advertising purposes and reconstruction of websites. The web log data obtained by a web server can be applied to clustering process, where the users are identified based on IP address and their usage based on session duration. Users are clustered based on Navigation patterns of all web pages related to one website. The main goal of this paper is developing an optimized clustering algorithm for clustering of web users section 2 covers a related work of clustering of web users on web usage data, section 3 is about existing work and section 4 gives the details of proposed work and section 5 describes the results and section 6 discussed conclusion along with future work.

2. Related work:

In recent days clustering of online users becomes an active research area in web mining. Clustering the users in web log mining is performed to group or the similar navigation patterns done by the users in a given website [1]. The Literature Survey says that developed an efficient clustering technique of web logs for web pages based on the hit count [2]. It comprises data pre-processing, user and session identification, path completion and clustering is performed using Farthest First Technique (FFC used for clustering. But in this method occurrence of outliers is more while clustering and not suitable for smaller datasets [3] did clustering using a Basic Probability Assignment (BPA) method , but it is also not effective for large data sets[4] proposed fuzzy clustering and presented a Dempster-Shafer theory to model the web user navigation behavior. Tasawar et al [5]., proposed a session based web users clustering method using similarity measure and click history of users and accessed web Pages for a structured information in the web site. Chaofeng Li et. Al [6] developed an algorithm to cluster the user sessions from web server data by using Robust clustering for deciding initial point for each cluster and number of clusters are based on different applications. Dariusz Krol et al.[7] presented an algorithm for analyzing user behavior by clustering the sessions and it is represented as vectors where rows represent web page and column represents number of times the web page is visited using Hard C-means algorithm. Clustering the user ratings of any products in online sites using K-Nearest neighboring (KNN) approach[8] , which will improve the performance of a website and for increasing the scalability of the recommendation to the new users. M.Parimala et.al[9]proposed a clustering method based on Expectation and maximization(EM) to find the user similar interests by creating a frame

work with fuzzy c-means technique. This expectation and maximization will improve the accuracy of clustering. But this method is suitable for small data sets with static data only. From the above discussion it is understood that majority of clustering techniques are not scalable to large datasets. Hence, a Clustering method which is suitable to variable sized data set and can arrange elements in optimum way is needed.

3. Existing System:

Web user clustering is one of the essential task in Web usage analysis.The goal of clustering is to Group data points that are close (or similar) to each other and identify such groupings (or clusters) in an unsupervised manner. Information of web user clusters has been widely used in many applications, such as solution of website structure design and optimization. There are many clustering techniques employed for finding the interested patterns among users. One among such technique is clustering of web users through Matrix Influence Degree (MID), which contains influence degree of each web page corresponding to the user. Simple k-Means [10] is applied over the obtained MID to obtain the clusters.

Density based clustering algorithm[11] is an efficient algorithm for clustering of web users because it uses density distribution , epsilon neighbourhood and density reachability measures for creating number of clusters. In Density based clustering all data points in the cluster satisfies 2 properties.

Property1: All data points are equally density connected.

Property2: With in a cluster if any one data point is density connected to other data point it is also a part of that cluster. The algorithm starts with randomly selects any one point as initial point that has not been visited. Clusters are formed with finding epsilon neighbourhood for randomly selected point if it consists of more number of points then clusters are initiated then it considers for all the data points finds the epsilon neighbourhood which data point is having more than that one forms another cluster. For each core-point c create an edge from c to every point p in the ε-neighborhood of c [12]. If any representative point does not have core point then it terminates.similarly this process is continues for all unvisited points until all clusters are formed. Let X be the set of nodes that can be reached from c by going forward to create cluster X U{c}.And another clustering is graph clustering which is also used to group the similar web users in different density and size. CHAMELEON is a one type of graph clustering where data points are considered as objects and similarity between the nodes are considered as weights. Cluster similarity is calculated by using relative closeness and interconnectivity. Using these two measures it constructs an k-nearest neighbour graph. CURE is a novel clustering algorithm [13]for large databases. It first selects constant number of scattered points which captures the shape and extent of a cluster. By selecting any fraction point x these scattered points shrunk to centroid of a cluster. Scattered points towards fraction x now considered as a representative points. In each step, CURE algorithm merges the closest representatives in to a cluster. It overcomes the problems of centroid based approach clustering methods. CURE algorithm also suitable to cluster the data elements of non-spherical shape.

Contribution of the paper as follows:

- Time consuming to build the model over large records and increases over the size of MID.
- Web user clusters are dependent on the initial point selected, which have large influence over the cluster sizes.
- More number of outliers can cause problem of ambiguity.

DBSCAN Algorithm does not suitable for high dimensional data.

4. Proposed Algorithm:

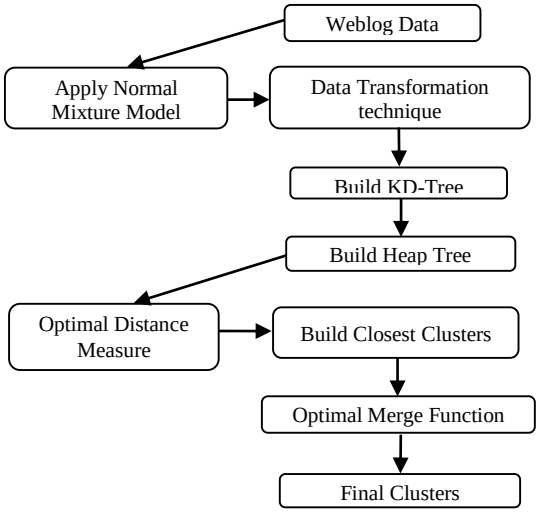


Fig 1: Optimized CURE clustering

In the above architecture model, a real-time web log data is given as input to the normal distribution based mixture method and then to data transformation. This data transformation procedure is used to form the uniformly distributed initial clusters for KD-tree

[14][15]. In the next step, KD-tree and Heap tree are built on the transformed initial cluster objects [16]. Then closest clusters are computed and merged using the optimal distance measure and optimal merge function. Finally, specified web-log clusters are identified in the KD-tree using the Heap tree model.

4.1 Optimized CURE Clustering

D: Log dataset

m: Required number of clusters for cluster initialization

Step 1:

for i=1 to m do

$u_1 \leftarrow$ Mean of initial data objects.

$\sigma_i^2 \leftarrow 1$

$\phi_i \leftarrow 1/m$

End for

For k=1 to N do

For i=1 to m do

$$z(n,i) \leftarrow \phi_i \left[2.\pi.\sigma_i\right]^{-D^2/2} . e^{\frac{1}{2\sigma_i^2}} \left\| X_n - \mu_k \right\|^2 / \min\{D_k\}$$

End for

$$Z(n) \leftarrow \frac{z(n,i)}{\sum_k z(n,i)}$$

End for

// Update cluster parameter

For i=1 to m do

$\phi_i \leftarrow z(n,k)/N$

$$\mu_i \leftarrow \frac{\sum_n \left[z(n,k).X_n\right]}{\sum_n z(n,k)}$$

$$\sigma_i^2 \leftarrow \frac{\sum_n \left[z(n,k).\left\|X_n - \mu_i\right\|\right]}{\sum_n z(n,k)}$$

End for

Until convergence of initial m clusters.

Step 2:

D' ← Optimal initial clusters in step -1

G ← buildKD-tree(D')

H ← buildHeap(D')

m ← Number of clusters

For k=1 to m clusters do

if size(H)>m then

p ← min(H)

q ← p.nearest;

remove(H,q)

s ← OptimalMerge(p,q)

Remove_Represent(G,p)

Remove_Represent(G,q)

insert_represent(G,s)

s.nearest ← x //arbitary cluster in H

for each object x ∈ H do

if distance(S,x)< distance (s,s.nearest_pt) then

if s.nearest_pt ∈ p or x.nearest ∈ q

if distance(x,x.nearest_pt)<distance(x,s)

x.nearest_pt ← nearest _nearest(G, x, distance(x,s))

else

x.nearest ← s

Relocate(H,x)

end if

end if

else if distance(x,x.nearest)> distance (x,s) then

x.nearest ← s

Relocate(H,x)

end if

Insert(H,s)

end for

In the Optimized CURE Clustering, step 1 represents the weblog data transformation procedure to maintain the data distribution as normal for initial cluster initialization. In the initial step, total objects are partitioned into 'm' parts. For each object in the 'm' partitions, data object is transformed using the normal mixture measure z and then it is normalized. In the step

2, 'm' initial clusters are given to KD-tree and Heap tree for data clustering. Initially, all the objects in the heap is considered as single cluster and representative object. For each object in the m^{th} cluster, extract top element from the heap tree and it is represented as p. In the next step, find the nearest element p from the heap tree as q. Here, p and q are merged using the optimal merging function. A new cluster's' is represented as the union of p and q. In the next steps, the nearest object to the cluster's' is identified as minimum distance computation in the heap tree. This process is repeated until 'm' clusters or heap tree is empty.

Algorithm 2: OptimalMerge(p,q)

$s \leftarrow p \cup q$

$$\text{s.meanprob} \leftarrow \sum_{\min(p,q)} |p_i - q_i| \left\{ \frac{|p|.p}{|p|+|q|} \right\}$$

```
Temp ← {}
for i=1 to |R| do
    maxD ← 0
    for each object obj in cluster s do
        if  i=1
         then
             minD ← distance(obj,s.mean)
         else
              minD ← min{dis tan ce(obj, t) : t ∈ Temp
         if (minD>=maxD) then
                 maxD ← minD
                 maxPt ← obj
        end if
    end for
    Temp ← Temp ∪ {max Pt}
    end for
for each object obj in Temp do
      s.represent ← s.represent ∪ {|(1 − η) | obj + η(s.mean − obj)}
//  η scaling factor
return  s
end
```

$$\text{Distance(U,V)} \leftarrow \frac{\max|\mu_U(x_i)-\mu_V(x_i)|}{\min|\sigma_U^2(x_i)-\sigma_V^2(x_i)|}.\text{Manhat tan}(u,v)$$

In the optimal merge method, two cluster objects are taken as input to form the new cluster. In the merging process, the average inter and intra cluster measure is computed on the input two clusters to form a new cluster's'. Using this measure, the farthest point in the cluster is identified from the heap tree as representative point. Finally, outliers are removed using the scaling factor equation in the merged cluster.

5. Experimental results

Experimental results are performed on real-time weblog dataset taken from the http://www.vtsns.edu.rs. Various statistical measures are performed on the proposed model to compare the cluster quality and miscluster rate. Recall, Precision and Accuracy are used to evaluate the cluster accuracy of the proposed model to the existing models.

Sample Cluster distance measure between object and representative point are shown below

Distance value 3.5820569289930266

Data Object :16,83.136.179.11,,'[16/Nov/2009:03:27:23 +0100] ','GET/ispit_odbijeni.php HTTP/1.1',200,1662,'URL-link''

Representative Point 493,'91.240.109.81 ',--,'[02/Jan/2010:16:00:52 +0000]','GET/noviSajt/administrator/history/historyFrame.html HTTP/1.0',200,4053,'URL-link''

Distance value 3.5858671951048278

Data Object :17,83.136.179.11,--,'[16/Nov/2009:03:37:32 +0100] ','GET/ispit_odbijeni.php HTTP/1.1 ',200,6554,'URL-link''

Representative Point 493,'91.240.109.81 ',--,'[02/Jan/2010:16:00:52 +0000]','GET/noviSajt/administrator/history/historyFrame.html HTTP/1.0',200,4053,'URL-link''

Distance value 3.4754754754754753

Data Object :18,82.208.255.125,--,'[16/Nov/2009:03:37:44 +0100] ','GET/ispit_rezultati.php HTTP/1.1 ',200,4053,'URL-link''

Representative Point 493,'91.240.109.81 ',--,'[02/Jan/2010:16:00:52 +0000]','GET/noviSajt/administrator/history/historyFrame.html HTTP/1.0',200,4053,'URL-link''

Distance value 3.4912701705773483

Data Object :19,83.136.179.11,--,'[16/Nov/2009:04:13:43 +0100] ','GET/ispit_rezultati.php HTTP/1.1 ',200,3669,'URL-link''

Representative Point 493,'91.240.109.81 ',--,'[02/Jan/2010:16:00:52

+0000]','GET/noviSajt/administrator/history/historyF
rame.html HTTP/1.0',200,4053,'URL-link''

Distance value 3.4902691695763473

Data Object :20,82.117.202.158,--
,'[16/Nov/2009:04:17:26 +0100]
','GET/ispit_odbijeni.php HTTP/1.1
',200,3669,'URL-link''

Representative Point 493,'91.240.109.81 ',--
,'[02/Jan/2010:16:00:52
+0000]','GET/noviSajt/administrator/history/historyF
rame.html HTTP/1.0',200,4053,'URL-link''

Distance value 3.581863191100824

Data Object :21,82.208.255.125,--
,'[16/Nov/2009:05:17:39 +0100]
','GET/ispit_raspored_god.php HTTP/1.1
',200,6554,'URL-link''

Representative Point 493,'91.240.109.81 ',--
,'[02/Jan/2010:16:00:52
+0000]','GET/noviSajt/administrator/history/historyF
rame.html HTTP/1.0',200,4053,'URL-link''

Distance value 3.4714714714714714

Data Object :22,82.208.255.125,--
,'[16/Nov/2009:05:17:41 +0100]
','GET/ispit_raspored_god.php HTTP/1.1
',200,4053,'URL-link''

Representative Point 493,'91.240.109.81 ',--
,'[02/Jan/2010:16:00:52
+0000]','GET/noviSajt/administrator/history/historyF
rame.html HTTP/1.0',200,4053,'URL-link''

Distance value 3.5750499219860195

Data Object :23,'82.208.207.41 ',--
,'[16/Nov/2009:05:18:40 +0100]
','GET/ispit_rezultati.php HTTP/1.1
',200,1662,'URL-link''

Representative Point 493,'91.240.109.81 ',--
,'[02/Jan/2010:16:00:52
+0000]','GET/noviSajt/administrator/history/historyF
rame.html HTTP/1.0',200,4053,'URL-link''

Table 1: Clusters to data distribution:

Cluster No	No of items	Distribution (%)
0	181	18%
1	152	15%
2	158	16%
3	114	11%
4	39	4%
5	77	8%
6	75	8%
7	89	9%
8	115	12%

Table 2: Cluster quality and miscluster rate of proposed model to the existing models, when datasize=500 instances

LogDatasize=500		
Algorithm	**Avg Cluster Quality**	**Miscluster Rate**
Kmeans	0.73	0.32
Kmediods	0.762	0.283
DBSCAN	0.816	0.24
CURE	0.897	0.182
Optimized CURE Clustering	0.934	0.106

Table 2, describes the performance of the cluster quality and mis-cluster rate of the Optimized CURE clustering to the existing models when the log datasize is 500 instances. From the table, it is clearly observed that proposed model have high cluster quality (>0.93) and mis-cluster rate(<0.12) compared to K-Means, K-Medoids, DBSCAN, CURE algorithms .

Table 3: Cluster quality and miscluster rate of Optimized CURE Clustering to the existing models, when datasize=1000 instances

logDatasize=1000		
Algorithm	**Avg Cluster Quality**	**Miscluster Rate**
Kmeans	0.727	0.337
Kmediods	0.773	0.319
DBSCAN	0.811	0.298
CURE	0.913	0.173
Optimized CURE Clustering	0.941	0.113

Table 3, describes the performance of the cluster quality and mis-cluster rate of the proposed model to the existing models when the log datasize is 1000 instances. From the table, it is clearly observed that

proposed model have high cluster quality(>0.94) and mis-cluster rate(<0.15) compared to K-Means, K-Medoids, DBSCAN, CURE algorithms.

Graphical representation of Cluster quality and miscluster rate when data size is 1000.

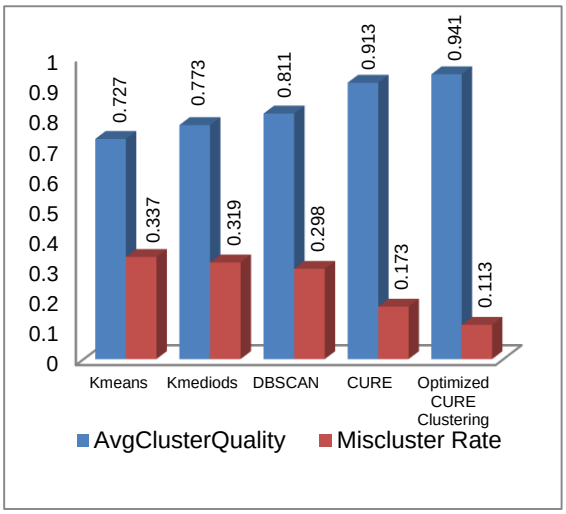
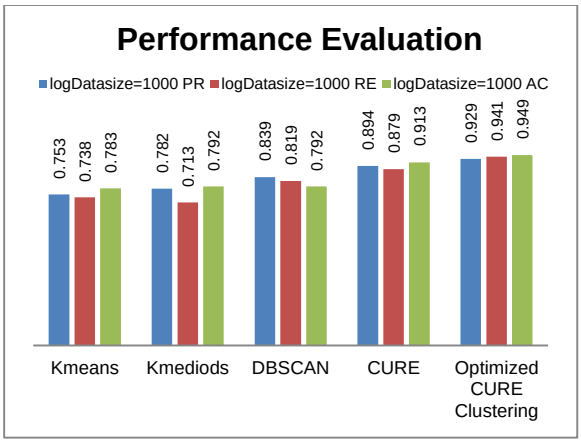


Table4: Performance metrics of Optimized CURE Clustering to the traditional models in terms of PR,RE,AC.

logDatasize=1000			
Algorithm	**PR**	**RE**	**AC**
Kmeans	0.753	0.738	0.783
Kmediods	0.782	0.713	0.792
DBSCAN	0.839	0.819	0.792
CURE	0.894	0.879	0.913
Optimized CURE Clustering	0.929	0.941	0.949

Table 4, describes the performance analysis of proposed model to the traditional models in terms of precision, recall and accuracy measures. From the table , it is observed that Optimized CURE Clustering has high computational accuracy ,precision and recall compared to the existing models.

Graphical representation of performance metrics:



6. Conclusion :

In this paper, an improved CURE Clustering approach using Normal Mixture distribution model was implemented to cluster the web users on weblog dataset. Normal mixture model is used to transform the web log data as initial clusters. In this approach, KD tree followed by heap tree data structures are used to represent the clusters and representative points. In this model, we have optimized the initial clusters representation, merging process and distance measure of traditional CURE clustering algorithm. Also, Manhattan distance measure is used to find the nearest distance between the heap objects with merge operation to determine closest clusters. Experimental results show that proposed model has high computation efficiency in terms of cluster quality precision and error rate compared to traditional models. In future, this work can be extended to improve the real-time web log processing using parallel programming.

7. References:

[1] Alka Tripathi and Kireeti panwar, "Modified CURE Algorithm with enhancement to identify number of clusters", International journal of Artificial Intelligence and soft computing,pp. 226-240,2016.

[2] Kalaivani S, Vidyapriya V, An Efficient Clustering Technique for Weblogs,IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 2 Issue 7,pp.516-525,2015.

[3] Sumathi P, Uma Maheswari B, A New Clustering and Pre-processing for Web Log Mining, World Congress on Computing and Communication Technologies.IEEE,pp.25-29,2014.

[4] Frugui H, Joshi A, Krishnapuram R, Nasraoui , Extracting web user profiles using relational

competitive fuzzy clustering, International Journal on Artificial Intellegence Tools pp. 509-526,2000.

[5] Joshi A, Krishnapuram R, Nasraoui O , Low-complexity fuzzy relational clustering algorithms for web mining, IEEE Transaction of Fuzzy System 4 (9) pp. 596-607,2003.

[6] Li, C, Algorithm of Web Session Clustering Based on Increase of Similarities. In: Proceedings of International Conference on Information Management, Innovation Management and Industrial Engineering, pp. 316–319. IEEE, 2008.

[7] Jianxi Zhang , Lin Shang and Lunsheng Wang , Peiying Zhao , "Web usage mining based on fuzzy clustering in identifying target group", ISECS International Colloquium on Computing, Communication, Control, and Management, Vol. 4, pp. 209-212, 2009.

[8] Herlocker J, O'Conner, M.,Clustering Items for Collaborative Filtering. in Proceedings of the ACM SIGIR Workshop on Recommender Systems. , Berkeley, CA: ACM Press,1999.

[9] Parimala M , Poongothai K, and Sathiyabama S," Efficient Web Usage Mining with Clustering", IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 6, No 3, November 2011.

[10] Ahmed Alsayat, Hoda El-Sayed, "Social Media Analysis using Optimized K-Means" Clustering" pp.1-6,2016.

[11] Zahid Ahmed Ansari ,"Discovery of Web User Session Clusters Using DBSCAN and Leader Clustering Techniques", International Journal for Research in Applied Science & Engineering Technology, Volume 2 Issue 12, December 2014.

[12] Dias G, Ranathunga S, Udanta M, "Modelling website user behaviors by combining the EM and DBSCAN algorithms", IEEE, pp.168-177,2016.

[13] Katie Owens, Conor Mettenburg, Evan Cohen, Alex Ripley, Ruben Aghayan, and William Scherer," Using Online User Behavior to Predict Demographics", 2016 IEEE Systems and Information Engineering Design Conference ,pp.78-83,2016.

[14] Phoha V.V, Xie Y, Web User Clustering from Access Log Using Belief Function, in: Proceedings of the 1st international conference on Knowledge capture, pp. 202-208,2001.

[15] Simon Fong, Sohail Asghar , Tasawar Hussain, "A hierarchical cluster based preprocessing methodology for Web Usage Mining", 6th International Conference on Advanced Information Management and Service (IMS), pp. 472-477,2010.

