Segmentation of Pathological MR Images for Discovery of Useful Patterns Using Various Clustering Techniques

Mohammed Zakir Bellary[1],Dr B.Aziz Musthafa[2] and Dr Zahid Ansari[3]
[1]Dept. of Electronics and Communication
P.A. College of Engineering
Mangalore, India

[2]Dept of Computer Science Engineering
Bearys Institute of Technology
Mangalore, India
[3]Dept of Computer Science Engineering
P.A. College of Engineering
Mangalore, India

Abstract— The Huge demand of Magnetic Resonance Imaging (MRI) in the world of medical field has helped the doctors to analyze and detect relevant disease. MRI is not only a reliable technique for the assessment of pathology but also a useful tool for monitoring the progression of disease. The capability of MRI to provide a fast three dimensional visualization has resulted in this technology being extensively used for the diagnosis and therapy of pathologies of various organs which are then used during surgery. Clustering techniques are widely used in image segmentation to capture similar interests and trends among users accessing a data image. Clustering aims to divide a data set into groups or clusters where inter-cluster similarities are minimized while the intra cluster similarities are maximized. This paper reviews some of the popularly used clustering techniques: *k*-Means, *k*-Medoids, Fuzzy C-Means, Leader and DBSCAN. Performance and validity results of each technique are presented and compared.

Keywords-pathology;MRI; k-means clustering; k-medoids clustering; leader clustering; DBSCAN

I. INTRODUCTION

Clustering techniques are widely used in data to capture similar interests and trends among users accessing a Web site. Clustering aims to divide a data set into groups or clusters where inter-cluster similarities are minimized while the intra cluster similarities are maximized. Details of various clustering techniques can be found in survey articles [1]-[3]. The ultimate goal of clustering is to assign data points to a finite system of k clusters. Union of these clusters is equal to a full dataset with the possible exception of outliers. Clustering groups the data objects based only on the information found in the data which describes the data objects and the relationships between them.

Some of the main categories of the clustering methods are [4]: i) *Partitioning* methods, that create k partitions of a given data set, each representing a cluster. Typical partitioning methods include k-means, k-medoids etc. In *k-means* algorithm each cluster is represented by the mean value of the data points in the cluster called centroid of the cluster. On the other hand and in *k-medoids* algorithm, each cluster is represented by one of the data point located near the center of the cluster called medoid of the cluster. Leader clustering is also a partitioning based clustering techniques which generates the clusters based on an initially specified dissimilarity measure, ii) *Hierarchical* methods create a hierarchical decomposition of the given set of data objects. A hierarchical method can be classified as being either agglomerative or divisive, based on how the hierarchical decomposition is formed. iii) *Density- based* methods form the clusters based on the notion of density. They can discover the clusters of arbitrary shapes. These methods continue growing the given cluster as long as the number of objects or data points in the "neighborhood" exceeds some threshold. They can also filter out noise and outliers. DBSCAN is a typical density-based method that grows clusters according to a density-based connectivity analysis. iv) *Grid-based* methods quantize the data object space into a finite number of cells that form a grid structure. All the clustering operations are performed on the grid structure. v) *Model-based* methods, that discover the best fit between data points given a mathematical model. Mathematical model is usually specified as a probability distribution.

The remainder of the paper is organized as follows. Section II presents a overview of MRI images of different medical images using clustering techniques and the underlying concepts. Section III presents each of the *k*-Means, *k*-Medoids, Leader and DBSCAN clustering techniques in detail along with the underlying mathematical foundations. Section IV describes the experimental results of each technique, followed by a comparison of the results. A brief conclusion and future work are presented in Section V.

II. DATA MINING USING CLUSTERING

A number of clustering algorithms have been used for medical MR image segmentation. A typical clustering starts with the matrix representing the image pixels and partitions this multi-dimensional space into k groups of segments that are close to each other based on a measure of distance or similarity among the vectors (such as Euclidean or Manhattan distance). Clusters obtained in this way can represent image segments based on their common characteristics. In order to determine similarity between an image pixel and target segment, the centroid vector corresponding to each segment is computed which is the representation of that image segment. Some of the research works related to medical image segmentation using clustering are described below.

Bandyopadhyay et al. have also performed analysis of medical images especially the MRI of brain using K-means and DBSCAN Clustering methods. The main benefit of using DBSCAN over K-means is that DBSCAN is efficient to handle noise points as outlier points but k means cannot handle noise with such a high efficiency. The DBSCAN method can find arbitrarily shaped or Non-convex shaped clusters .Tha factors get increase in size as the database increases exponentially in k-means and linearly in DBSCAN.[5]

Chebbout et al. have analyzed high quality of Natural images like fruits and vegetables etc in terms of parameter such as Connectivity, Dunn index,and Silhoutte width in different color spaces such as RGB,HSV and CIE L*A*B*. The K-Means and SOM clustering algorithms performed the high quality of image clustering in HSV and CIE L*a*b* rather than the RGB colour space in silhouette width. The Results demonstrate a significant variation in clustering quality where K-Means and SOM algorithms produced clusters of maximum compactness, maximum separation and minimum connectedness in the HSV and CIE L*a*b colour space, while RGB yielded poorer results as compare to earlier. [6].

Ajala Funmilola et al. have compared the benchmark such as mode of operation or iteration ,time taken and accuracy of three different clustering methods i.e. k-means, fuzzy c-means fuzzy k-c-means used for medical image segmentation by comparing the above three methods. Fuzzy segmentation methods are of considerable benefits, because they could retain much more information from the original image than hard segmentation methods. FCM allows pixels to belong to multiple clusters with varying degrees of membership with flexibility. This is a soft segmentation method that has been used extensively for segmentation of MR images [7].

Dhanachandra et al. have performed segmentation method on the image of Malaria infected blood cell by using K-means clustering, Fuzzy C-means clustering, and mountain clustering and subtractive clustering method. They proposed algorithm in which it consists of partial contrast stretching, subtractive clustering, *k*-means clustering and median filter. RMSE and PSNR are checked and observed that they have small and large value respectively, which were the condition for good image segmentation quality. It has better performance result [8].

Norouzi et al. have suggested different medical Image Segmentation Methods and Algorithms and Applications on MR images of knee bone by using K-means, Fuzzy-C means and Expectation maximization. Computation time in k-means is less compared to classification and its an unsupervised technique but selecting the clusters randomly is cumbersome.FCM is good in medical image analysis to correct corrupted images but significant enough experience is required. In Expectation maximization we compute means and variance that help us to separate different tissues from each other but In EM,it is based on iteration that help us to iterate continuously until the stop condition is true [9].

Abdel-Maksoud et al. have proposed a hybrid technique that gets the benefits of the K-means clustering for image segmentation in the aspects of minimal computation time. In addition, it can get advantages of the Fuzzy C-means in the aspects of accuracy. In order to have more visualization clearly, Intensity adjustment and 3D evaluation can be done by using 3D slicer [10].

Adhikari et al. have applied csFCM algorithm on Brain MR images in order to have superior performance in terms of qualitative and quantitative studies such as, cluster validity functions, segmentation accuracy, tissue segmentation accuracy and receiver operating characteristic (ROC) curve on the image segmentation results and also this proposed algorithm produce superior segmentation results even in the presence of noise and intensity inhomogeneity in MRI data but The presence of high percentage of IIH(Intensity Inhomogeniety)along with noise in MRI data may degrade the performance of the FCM algorithm, when it is compared to other algorithm [11].

Folkesson, Jenny, et al. have discussed that Segmentation using I-FCM clustering of trabecular bone from MR images which will be a challenging task due to spatial resolution limitations, signal-to-noise ratio constraints, and signal intensity inhomogeneities. I-FCM computation time is less compare to Bone enhancement FCM clustering technique. The method shown to be more precise (reproducible), more highly correlated with HR-pQCT and to better discriminate between participants with and without vertebral fractures than the other evaluated approaches. The method may thus have potential to become a valuable component in imaging-based studies related to the prediction of fracture risk and to the effect of treatment in osteoporosis [12].

Aprovitola et al. have done literature review on Knee bone segmentation from MRI by using various techniques among that the widest used is clustering algorithm A modified fuzzy C-means clustering incorporating second order features that were adopted to provide bone enhancement at multiple scales. Such an approach accounted for the partial volume effects in MR images and was demonstrated to be robust with respect to noise. The other Clustering algorithm requires moderate user interaction also the lack of flexibility [13].

Pham et al. have proposed the Current methods in medical image segmentation and applied on Brain MR Images. Basically the clustering methods train themselves, using the available data without training data and that they are unsupervised method. The fuzzy c-means algorithm generalizes the K-means algorithm, allowing for soft segmentations based on fuzzy set theory. The EM algorithm applies the same clustering principles with the underlying assumption that the data follow a Gaussian mixture model. It iterates between computing the posterior probabilities and computing maximum likelihood estimates of the means, covariance, and mixing coefficients of the mixture model [14].

Zhang et al. have proposed an algorithm which is realized by modifying the objective function in the conventional fuzzy C-means (FCM) algorithm using a kernel-induced distance metric and a spatial penalty on the membership functions which is tested on T-1 weighted Brain MR images and also synthetic images .This novel FCM (KFCM) is more robust clustering approach than any other FCM. Experimental results on both synthetic and real MR images shows that the proposed algorithms have better performance when noise and other artifacts are present than the standard algorithms [15].

Castellanos et al. have proposed some nonlinear filtering processes such as the morphological filters which are able to reduce some inherently associated noise with less degradation. The method used is Adaptive fuzzy Leader clustering algorithm (AFLC) which is applied on ultrasound

image of 12 weeks old fetus where the head, legs, umbilical cord, and one arm. It has been found that there is significant Noise reduction in the image i.e. speckle noise without degradation of the image [16].

Castellanos et al. have also proposed a Segmentation of magnetic resonance images using a neuro-fuzzy algorithm i.e. Adaptive fuzzy Leader clustering algorithm.(AFLC) .In this technique segmentation method such as clustering is used on Brian MRI to improve the performance. By using AFLC, the MRI of brain has shown improved performance in misclassification rates(MCR) by using a vigilance parameter such as pixel intensity which results in improved segmentation.MCR is defined as the number of pixel misclassified by the algorithm divided by the total number pixels in an image [17].

Al-Dmour et al. have proposed the semi-supervised algorithm with Standard Fuzzy Clustering (SSSFC) which has advantages over FCM in terms of simplicity and low computational cost. Also, SSSFC is used to enhance the results being obtained by the FCM with pre-defined membership matrix that has been tested on Brain MR Images. The main drawbacks of FCM is its sensitivity to noise and outliers, and high computational cost compared to standard fuzzy clustering [18].

L. Juang et al. have proposed the Color-converted segmentation with K-means clustering algorithm for tracking objects in medical images is per-formed to be very promising for MRI applications. The brain regions related to a tumor or lesion can be exactly separated from the colored image. This proposed method will be able to help pathologists distinguish exactly lesion size and region. [19]

III. Data Clustering Techniques

In this section a detailed discussion of each clustering technique and its underlying mathematical model is presented.

A. k-Means Clustering Algorithm:

The *k*-Means clustering or Hard *c*-Means clustering algorithm [16] is one of the most commonly used methods for partitioning the data. Given a set of *m* data points $X = \{x_i \mid i = 1 \cdots m\}$, where each data point is a *n*-dimensional vector, *k*-means clustering algorithm aims to partition the *m* data points into *k* clusters ($k \le m$) $\boldsymbol{C} = \{c_1, c_2, \ldots, c_k\}$ so as to minimize an *objective function* (or a cost function) $J(V, X)$ of dissimilarity [21], which is the within-cluster sum of squares. In most cases the dissimilarity measure is chosen as the Euclidean distance. The objective function is an indicator of the distance of the *n* data points from their respective cluster centers. The objective function *J*, based on the Euclidean distance between a data point vector x_i in cluster *j* and the corresponding cluster center v_j, is defined in (2).

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

where, $J_i(x_i, v_j) = \sum_{i=1}^{m} u_{ij} . d^2(x_i, v_j)$,

is the objective function within cluster c_i,

$u_{ij} = 1$, if $x_i \in c_j$ and 0 otherwise.

$d^2(x_i, v_j)$ is the disatnce between x_i and v_j

Euclidian distance between various data points and cluster centers can be calculated using (3).

$$d^2(x_i, v_j) = \left\| \sum_{k=1}^{n} x_k^i - v_k^j \right\|^2 \qquad (3)$$

where, n is the number of dimensions of each data point

x_k^i is the value of k^{th} dimensions of x_i

v_k^j is the value of k^{th} dimensions of v_j

The k-means clustering first initializes the cluster centers randomly. Then each data point x_i is assigned to some cluster v_j which has the minimum distance with this data point. Once all the data points have been assigned to clusters, cluster centers are updated by taking the weighted average of all data points in that cluster. This recalculation of cluster centers results in better cluster center set. The process is continued until there is no change in cluster centers.

Algorithm: *k*-Means clustering algorithm for partitioning, where each cluster's center is represented by the mean value of the data points in that cluster.

Input: *k*, the number of clusters and Set of *m* data points $X=\{x_1, \ldots, x_m\}$.

Output: Set of *k* centroids, $V=\{v_1, \ldots, v_k\}$, corresponding to the clusters $C=\{c_1, \ldots, c_k\}$, and membership matrix $U=[u_{ij}]$.

Steps:

1) Initialize the *k* centroids $V=\{v_1, \ldots, v_k\}$, by randomly selecting *k* data points from *X*.

2) **repeat**

   i) Determine the membership matrix *U* using (8), by assigning each data point x_i to the closest cluster c_j.

   ii) Compute the objective function $J(X,V)$ using (6). Stop if it below a certain threshold ε.

   iii) Recompute the centroid of each cluster using (9).

3) **until** Centroids do not change

The partitioned clusters are defined by a $m \times k$ binary membership matrix $\boldsymbol{U}$, where the element u_{ij} is 1, if the *i*th data point x_i belongs to the cluster *j*, and 0 otherwise. Once the cluster centers $V = \{v_1, v_2, \ldots, v_k\}$, are fixed, the membership function u_{ij} that minimizes (2) can be derived as follows:

$$u_{ij} = \begin{cases} 1; \text{ if } d^2(x_i, v_j) \le d^2(x_i, v_{j^*}) \; j \ne j^*, \forall \; j^* = 1, \cdots, k \\ 0; \text{otherwise} \end{cases} \qquad (4)$$

The equation (4) specifies that assign each data point x_i to the cluster c_j with the closest cluster center v_j. Once the membership matrix $U=[u_{ij}]$ is fixed, the optimal center v_j that minimizes (2) is the mean of all the data point vectors in cluster *j* :

$$v_j = \frac{1}{|c_j|} \sum_{i, x_i \in c_j}^{m} x_i \quad (5)$$

where,

$|c_j|$, is the size of cluster c_j and also $|c_j| = \sum_{i=1}^{m} u_{ij}$

Given an initial set of k means or cluster centers, $V = \{v_1, v_2, \ldots, v_k\}$, the algorithm proceeds by alternating between two steps: i) Assignment step: Assign each data point to the cluster with the closest cluster center. ii) Update step: Update the cluster center as the mean of all the data points in that cluster. The input to the algorithm is a set of m data points $X = \{x_i \mid i = 1 \cdots m\}$, where each data point is a n-dimensional vector, it then determines the cluster centers v_j and the membership matrix $\boldsymbol{U}$ iteratively as explained in Fig. 1.

The k-means algorithm provides locally optimal solutions with respect to the sum of squared errors represented by the error objective function. Since it is a fast iterative algorithm, it has been applied to a variety of areas [20].

The attractiveness of the k-means lies in its simplicity and flexibility. However, it suffers from major shortcomings that have been a cause for it not being implemented on large datasets. The most important among these are i) k-Means scales poorly with respect to the time it takes for large number of points; ii) The algorithm might converge to a solution that is a local minimum of the objective function. The main disadvantage of this algorithm lies in its sensitivity to initial positions of the cluster centroids [22].

Since the performance of the k-Means algorithm depends on the initial positions of the cluster centeroids, it is recommended to execute the algorithm multiple times, each with a different set of initial centroids.

B. K-Medoids Clustering Algorithm:

k-Medoid is a classical partitioning technique of clustering that clusters the data set of m data points into k clusters. It attempts to minimize the squared error, which is the distance between data points within a cluster and a point designated as the center of that cluster.

Figure 1. k-Means Clustering Algorithm

In contrast to the k-means algorithm, k-Medoids algorithm selects data points as cluster centers (or medoids). A medoid is a data point of a cluster, whose average dissimilarity to all the other data points in the cluster is minimal i.e. it is a most centrally located data point in the cluster [23]-[25].

Given a set of m data points $X = \{x_i \mid i = 1 \cdots m\}$, where each data point is a n-dimensional vector, k-mdoids clustering algorithm aims to partition the m data points into k clusters ($k \leq m$) $\boldsymbol{C} = \{c_1, c_2, \ldots, c_k\}$ so as to minimize an objective function representing the sum of the dissimilarities between each of the data points and its corresponding cluster medoid. Let $\boldsymbol{M} = \{m_1, m_2, \ldots, m_k\}$ be the set of medoids corresponding to C. The objective function $J(X, M)$ is defined in (7)

$$J(X,M) = \sum_{j=1}^{k} \left(\sum_{i=1}^{m} u_{ij} . d^2(x_i, m_j) \right), \quad (7)$$

where,

x_i is the i^{th} data point

m_j is the medoid of cluster c_j

$u_{ij} = 1$, if $x_i \in c_j$ and 0 otherwise.

$d^2(x_i, m_j)$ is the Euclidean disatnce between x_i and m_j

$$d^2(x_i, m_j) = \left\| \sum_{k=1}^{n} x_k^i - m_k^j \right\|^2 \quad (8)$$

where, n is the number of dimensions of each data point

x_k^i is the value of k^{th} dimensions of x_i

m_k^j is the value of k^{th} dimensions of m_j

The partitioned clusters are defined by a $m \times k$ binary membership matrix $\boldsymbol{U}$, where the element u_{ij} is 1, if the ith data point x_i belongs to the cluster j, and 0 otherwise. Once the cluster medoids $M = \{m_1, m_2, \ldots, m_k\}$, are fixed, the membership function u_{ij} that minimizes (7) can be derived as follows:

$$u_{ij} = \begin{cases} 1; \text{ if } d^2(x_i, m_j) \leq d^2(x_i, m_{j^*}) \; j \neq j^*, \forall \; j^* = 1, \cdots, k \\ 0; \text{otherwise} \end{cases} \quad (9)$$

The equation (9) specifies that assign each data point x_i to the cluster medoid m_j. Once the membership matrix $U=[u_{ij}]$ is fixed, the new cluster medoids m_j that minimizes (7) can be found using (10)

$$m_j = \arg\min_{xi \in c_j} \sum_{x_l \in c_j} d(x_i, x_l) \quad (10)$$

Algorithm: k-Medoids Clustering

Input: Set of m data points $X=\{x_1, \ldots, x_m\}$.

Output: Set of k medoids, $M=\{m_1, \ldots, m_k\}$, corresponding to the clusters $C=\{c_1, \ldots, c_k\}$, and membership matrix $U=[u_{ij}]$ that minimizes the sum of the dissimilarities of all the data points to their nearest medoid.

Steps:

4) Initialize the k medoids $V=\{v_1, \ldots, v_k\}$, by randomly selecting k data points from X.

5) **repeat**

iv) Determine the membership matrix U using (9), by assigning each data point x_i to the closest cluster c_j.

v) Compute the objective function $J(X,M)$ using (7). Stop if it below a certain threshold ε.

vi) Recompute the medoid of each cluster using (10).

Figure 2. k-Medoids Clustering Algorithm

The basic strategy of k-Medoids clustering algorithms is to discover k clusters in m objects by first arbitrarily selecting a representative data point (the Medoid) as the center for each cluster. Each remaining data point is clustered with the medoid to which it is the most similar. The algorithm takes the input parameter k, the number of clusters to be partitioned among a set of m objects. The most common realisation of k-medoid clustering is the Partitioning Around

Medoids (PAM) algorithm and is as described in Fig 2. It is more robust to noise and outliers as compared to *k*-means because a medoid is less influenced by outliers or other extreme values than a mean. It minimizes the sum of pair-wise dissimilarities instead of a sum of squared Euclidean distances as in case of k-means. Both methods require the user to specify *k*, the number of clusters.

C. *Leader Clustering Algorithm:*

The leader clustering algorithm [26] is based on a predefined dissimilarity threshold. Initially, a random data point from the input data set is selected as leader. Subsequently, distance of every other data point with the selected leader is computed. If the distance of a data point is less than the dissimilarity threshold, that data point falls in the cluster with the initial leader. Otherwise, the data point is identified as a new leader. The computation of leaders is continued till all the data points are considered. It should be noted that the result of the clustering depends on the chosen distance threshold. The number of leaders is inversely proportional to the selected threshold.

Given a set of m data points $X = \{x_i \mid i = 1 \cdots m\}$, where each data point is a n-dimensional vector. The Euclidean distance between the i^{th} data point $x_i \in X$ and j^{th} leader $l_j \in L$ (where L is a set of leaders) is given by :

$$d^2(x_i, l_j) = \left\| \sum_{k=1}^{n} x_k^i - l_k^j \right\|^2 \quad (11)$$

where, n is the number of dimensions of each data point

x_k^i is the value of k^{th} dimensions of x_i

l_k^j is the value of k^{th} dimensions of x_j

Fig. 3 below describes the leader clustering algorithm

D. *DBSCAN Clustering Algorithm*

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [25] is a density-based data clustering algorithm because it finds a number of clusters starting from the estimated density distribution of corresponding nodes.

Given a set of m data points $X = \{x_i \mid i = 1 \cdots m\}$, where each data point is a n-dimensional vector. The Euclidean distance between the two data points $x_p \in X$ and $x_q \in X$ is given by

$$d^2(x_p, x_q) = \left\| \sum_{k=1}^{n} x_k^p - x_k^q \right\|^2 \quad (12)$$

where, n is the number of dimensions of each data point

x_k^p is the value of k^{th} dimensions of x_p

m_k^q is the value of k^{th} dimensions of x_q

In this algorithm concept of a cluster is based on the notion of "ε-neighborhood" and "density reachability". Let the ε-neighborhood $N_\varepsilon(x_p)$ of a data point x_p, is defined as below:

$$N_\varepsilon(x_p) = \left\{ x_q \in X \mid d^2(x_p, x_q) \le \varepsilon \right\} \quad (13)$$

where, ε is the neighborhood distance

Let η be the minimum number of points required to form a cluster. A point x_q is directly density-reachable from a point x_p, if x_q is part of ε-neighborhood of x_p and if the number of points in the ε-neighborhood of x_p are greater than or equal to η as specified in (13).

Algorithm: Leader Clustering

Input: i) Set of m data points $X=\{x_1, \ldots, x_m\}$,
ii) α, the dissimilarity threshold.

Output: Set of clusters $C = \{c_1, \ldots, c_k\}$,

Steps:

1) $C = \phi, L = \phi, j = 1$ // Initilize the cluster and leader sets
2) $l_j = x_1$ // Initialize x_1 as the first leader
3) $L = L \cup l_j$
4) $c_j = c_j \cup x_1$
5) $C = C \cup c_j$
6) **for each** $x_i \in X$ where $i = 2, \ldots m$
7) **begin**
8) $j = \arg\min_{j,\ lj \in L} d(x_i, l_j)$
9) **if** $d^2(x_i, l_j) < \alpha$ **then**
10) $c_j = c_j \cup x_i$
11) **else**
12) $j = j + 1$
13) $l_j = x_i$
14) $L = L \cup l_j$
15) $c_j = c_j \cup x_i$
16) $C = C \cup c_j$
17) **endif**
18) **end**

Figure 3. Leader Clustering Algorithm

Let η be the minimum number of points required to form a cluster. A point x_q is directly density-reachable from a point x_p, if x_q is part of ε-neighborhood of x_p and if the number of points in the ε-neighborhood of x_p are greater than or equal to η as specified in (13)

$$x_q \in N_\varepsilon(x_p) \quad (14)$$
$$\left| N_\varepsilon(x_p) \right| \ge \eta$$

where η is the minimum number of points required for a cluster

x_q is called density-reachable from x_p if there is a sequence $x_1, \ldots, x_n$ of points with $x_1 = x_p$ and $x_n = x_q$ where each x_{i+1} is directly density-reachable from x_i. Two points x_p and x_q are said to be density-connected if there is a point x_o such that x_o and x_p as well as x_o and x_q are density-reachable.

A cluster of data points satisfies two properties: i) All the data points within the cluster are mutually density-connected.

ii) If a data point is density-connected to any data point of the cluster, it is part of the cluster as well.

Algorithm: DBSCAN

Input: i) Set of m data points $X=\{x_1, \ldots, x_m\}$,

ii) ε (epsilon), the neighborhood distance and

iii) η , the minimum number of data points required to form a cluster.

Output: Set of clusters $C = \{c_1, \ldots, c_k\}$,

Steps:

1) $C = \emptyset.$; $i = 0$;
2) **for each** $x_p \in X$ *and* $x_p.visited = false$
3) **begin**
4) $x_p.visited = true$
5) $N_p = N_\varepsilon(x_p)$ using (13)
6) **if** $|N_\varepsilon(x_p)| < \eta$ **then**
7) $x_p.noice = true$
8) **else**
9) $i = i + 1$
10) $C = C \cup c_i$
11) $c_i = c_i \cup x_p$
12) **for each** $x_q \in N$
13) **begin**
14) **if** $x_q.visited = false$ **then**
15) $x_q.visited = true$
16) $N_q = N_\varepsilon(x_q)$
17) **if** $|N_\varepsilon(x_q)| < \eta$ **then**
18) $N_p = N_p \cup N_q$
19) **if** $x_q \notin c_j \; \forall j = 1 \le j \le i$ **then**
20) $c_i = c_i \cup x_q$
21) **endif**
22) **endif**
23) **endif**
24) **end**
25) **endif**
26) **end**

Figure 4. DBSCAN Algorithm

Input to DBSCAN algorithm are i) ε (epsilon) and ii) η , the minimum number of points required to form a cluster. The algorithm starts by randomly selecting a starting data point that has not been visited. If the ε-neighborhood of this data point contains sufficiently many points, a cluster is started. Otherwise, the data point is labeled as noise. Later this point might be found in a sufficiently sized ε-neighborhood of a different data point and hence could become part of a cluster. If a data point is found to be part of a cluster, all the data points in its ε-neighborhood are also part of that cluster and hence added to the cluster. This process continues until the cluster is completely found. Then, a new unvisited point is selected and processed, leading to the discovery of a next cluster or noise. Fig. 4 describes the DBSCAN algorithm.

Although DBSCAN can cluster objects given input parameters such as ε and η, but it is the responsibility of the user to select these parameter values. Such parameter settings are usually empirically set and difficult to determine, especially for high-dimensional data sets.

IV. Experimental Results

A. *Image Segmentation Using Clustering Techniques:*

Segmentation of pathological MR Images is performed with the help of different clustering algorithms in order to discover useful segments from the images. These algorithms are i) *k*-Means ii) *k*-Medoids iii) Leader and iv) DBSCAN. Since the above clustering algorithms result in different clusters it is important to perform an evaluation of the results to assess their quality. We evaluated our results based on DB index and C Index which are two quality measures to evaluate the quality of the discovered clusters. These validity measures a described below:

Davies-Bouldin Validity Index: This index attempts to minimize the average distance between each cluster and the one most similar to it. It is defined as [27]:

$$DB = \frac{1}{k}\sum_{i=1}^{k} \max_{1 \le j \le k, j \ne i} \left(\frac{diam(c_i) + diam(c_j)}{dis(c_i, c_j)} \right) \quad (16)$$

An optimal value of the k is the one that minimizes this index.

C Index: It is defined as [28]:

$$C = \frac{S - S_{\min}}{S_{\max} - S_{\min}}, \quad (17)$$

Here S is the sum of distances over all pairs of objects form the same cluster. Let m be the number of those pairs and S_{min} is the sum of the m smallest distances if all pairs of objects are considered. Similarly S_{max} is the sum of the m largest distances out of all pairs. The interval of the C-index values is [0, 1] and this value should be minimized.

The results of application of various clustering algorithms are presented in the following subsections.

1) *k-Means Algorithm:*

We conducted multiple runs of k-Means algorithm by selecting the input parameter k (number of clusters) ranging from k = 2, …, 60. For each of these runs we computed the value of the clustering error function (J) using (2) which represents the sum of the squared error. We also computed the execution timings, Dunn's index, DB index and C index for all of the above runs. Table III describes the results after the application of *k*-Means clustering algorithm.

TABLE III

K-Means Clustering Results

Clusters	SSE (J)	DB Index	C Index	Execution Time(ms)
10	583.54	1.3395	0.1229	49
20	443.06	1.3456	0.1060	110
30	357.24	1.2228	0.0769	142
40	284.08	1.1045	0.0610	164
50	279.29	1.1345	0.0651	278

Clusters	SSE (*J*)	DB Index	C Index	Execution Time(ms)
60	260.64	0.8846	0.0783	188

One of the problems associated with the k-Means algorithm is that it may produce empty clusters depending on the initial centroids chosen. Graph in Fig. 6 Describes the number of empty clusters generated for different values of *k*. M.K. Pakhira [29] has proposed a modified k-means algorithm to avoid the empty clusters. K-medoids algoritm also rectifies this problem.

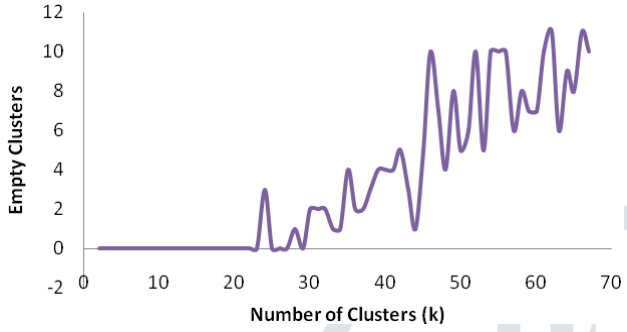

Figure 5. No. of Empty Clusters Vs. No. of Initial Clusters *k*

2) k-Medoids Algorithm:

We conducted the multiple runs of *k*-Medoids algorithm by selecting the input parameter *k* (number of clusters) ranging from k = 2, 60. For each of these runs we computed the value of the clustering error function (*J*) using (7), which represents the sum of the squared error. We also computed the execution timings, Dunn's index and DB index and C index for all of the above runs. Table IV describes the results after of *k*-Means clustering algorithm.

TABLE IV

K-MEDOIDS CLUSTERING RESULTS

Clusters	Error (*J*)	DB Index	C Index	Execution Time(ms)
10	613.73	1.4426	0.1622	7
20	512.81	1.4689	0.1543	7
30	352.88	1.2018	0.05	5
40	315.63	0.9413	0.23572	6
50	257.83	2.35	0.03	7
60	254.13	2.85	0.06	9

We compared the k-Means and k-Medoids algorithms based on clustering error (*J* as defined in equations (2) and (7)), cluster validity using C index and the execution time.

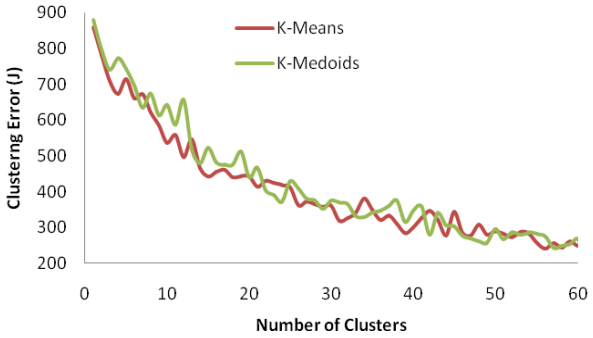

Figure 6. Clustering Error (*J*) Vs. No. of Clusters *k*

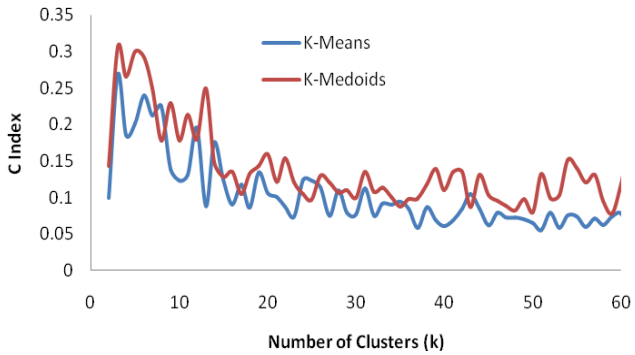

Figure 7. C Index Vs. No. of Clusters *k*

Our results (Fig. 7) show that the k-Means algorithm minimizes the clustering error (*J*) slightly better than the k-Medoids algorithm. C index values in graph plot of Fig. 8 indicates that the clusters of *k*-Means algorithm have better validity index than that of *k*-Medoids algorithm. On the other the execution timings of *k*-Medoids algorithms are faster than the that of *k*-Means algorithm as show in Fig.9.

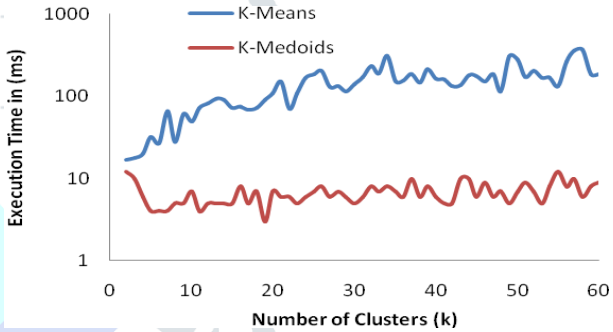

Figure 8. Execution Time in milliseconds Vs. No. of Clusters *k*

3) Leader Algorithm:

We conducted the multiple runs of Leader algorithm by selecting the input parameter ε (Dissimilarity Threshold) ranging from ε = 0.5, …, 3.5 in steps of 0.5. For each of these runs we computed the value of the clustering error. We also computed the execution timings, DB index and C index for all of the above runs. Table V describes the results after the application of Leader clustering algorithm.

TABLE V

LEADER CLUSTERING RESULTS

Epsilon (ε)	Error (*J*)	DB Index	C Index	Execution Time(ms)	No. of Clusters
1	26.19	0.3623	0.0021	3	115
1.5	76.81	0.5061	0.0348	2	86
2	216.62	0.5578	0.0588	2	56
2.5	398.81	0.7200	0.0801	1	33
3	467.07	0.9084	0.1878	2	26
3.5	624.87	0.8801	0.2407	1	14

Fig. 10 shows the results of Leader clustering. From the graph it is very clear that the number of discovered clusters is inversely proportional to the dissimilarity threshold ε.

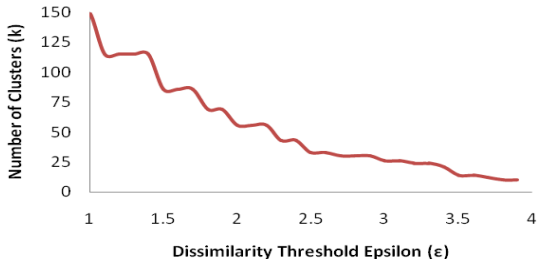

Figure 9. Number of clusters formed Vs. Dissimilarity Threshold ε

4) DBSCAN Algorithm:

We conducted the multiple runs of DBSCAN algorithm by selecting the input parameter ε (neighborhood disatnace) ranging from ε = 0.5, …, 3.5 in steps of 0.5. The other parameter η which indicates the minimum no. of points in a cluster is set in a range from η = 2, …, 10. For each of these runs we computed the value of the clustering error. We also computed the execution timings, DB index and C index for all of the above runs. Table VI describes the results after the application of DBSCAN algorithm for the value of η = 2.

TABLE VI

DBSCAN RESULTS

Epsilon (ε)	Error (*J*)	DB Index	C Index	Execution Time(ms)	No. of Clusters
1	765.9	1.2694	0.6606	13	21
1.5	804.881	1.3665	0.1984	20	7
2	870.2758	0.8415	0.0766	24	2
2.5	879.5672	0.8348	0.0500	13	2
3	865.1479	1.0874	0.0442	16	3
3.5	869.23	0.9092	0.0463	17	3

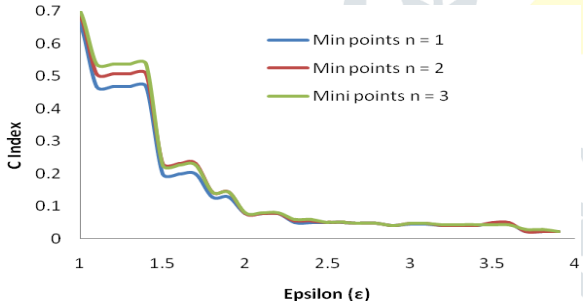

Figure 10. C Index Vs. Neighbourhood Distance ε

The graph plot in Fig. 11 displays the C index as a function of the neighbourhood distance ε, for different values of η (the minimum number of points in a cluster). The graph shows that the C index value improves as we increase the neighbourhood distance ε. It also improves if we decrease the value of η .

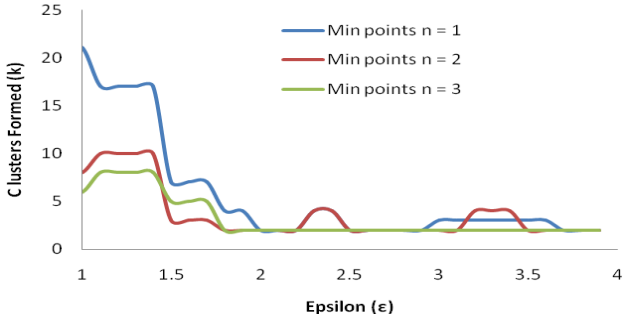

Figure 11. Number of clusters formed Vs. Neighbourhood Distance ε

The graph plot in Fig. 12 displays the number of clusters formed as a function of the neighbourhood distance ε, for different values of η (the minimum number of points in a cluster). The graph shows that the number of clusters formed decreases as we increase the neighbourhood distance ε. It also decreases if we increase the value of η.

The next two graphs compare the results of the Leader and DBSCAN techniques.

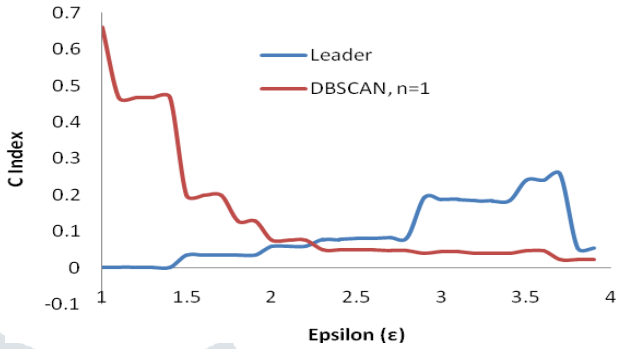

Figure 12. C Index Vs. Epsilon (ε)

The graph plot in Fig. 13 displays the C validity index value as a function of Epsilon (ε). Here ε is the dissimilarity threshold in case of Leader clustering and neighbourhood distance in case of DBSCAN. Our results show that in case of Leader clustering, validity index improves for lower values of dissimilarity distance ε. In case of DBSCAN, the validity index improves as increase the value of neighbourhood distance ε. Note that we have set the value of η to 1.

V. CONCLUSION AND FUTURE WORK

In this paper we have presented our framework for Medical MR image segmentation using k-Means, k-Medoids, Leader and DBSCAN clustering algorithms. We provided a detailed overview of these techniques. We also described the mathematical model and algorithm details related to the implementation of these clustering algorithms in order to discover the user sessions clusters. From the results presented in the previous section, we conclude the following points.

- K-means clustering produces fairly higher accuracy and lower clustering error as compared with k-medoids clustering algorithm.
- K-means algorithm may result in the formation of empty cluster while it is not the case with k-medoids algorithm.
- Our result shows that k-medoids algorithm gives reasonably better time performance than that of the k-means algorithm. The reason behind this is we are using a large data set. The k-Medoids algorithm requires to compute the distance between every pair of data objects only once and uses this distance at every stage of iteration. On the other for an optimal solution k-Means algorithm performs multiple runs and computes the distance between every data object and it's corresponding cluster center.
- Although Leader clustering algorithm does not require estimating the value of *k* at the beginning, it does require estimating the dissimilarity threshold ε.
- Number of clusters formed in Leader clustering is inversely proportional to the value of dissimilarity threshold ε.

- Leader clustering validity index (C index) improves as we increase the value of the dissimilarity threshold ε.
- DBSCAN algorithm can identify a data point as a noise or outlier.
- DBSCAN validity index (C index) improves as we decrease the value of the neighborhood distance ε.
- If we choose the same value for dissimilarity threshold in Leader clustering and neighbor distance in DBSCAN (while keeping η constant), the time performance of Leader clustering much faster than that of DBSCAN.

Another way is related with the use of fuzzy *c*-Mean clustering technique to discover the user session clusters. The reason behind this is, although the several clustering algorithms described are suitable in handling the crisp data which have clear cut boundaries, but in reality MRI data is not fully structured or semi-structured and contains the outliers and incomplete data, due to a wide variety of reasons of having imperfect precision in the boundaries of given MRI images. Therefore, MR images require modelling of multiple overlapping sets in the presence of significant noise and outliers. Fuzzy Clustering can be very useful for mining such semi structured, noisy and incomplete data.

REFERENCES

[1] P. Berkhin, "Survey of clustering data mining techniques," Springer, 2002.

[2] B. Pavel, "A survey of clustering data mining techniques," in Grouping Multidimensional Data. Springer Berlin Heidelberg, 2006, pp. 25–71.

[3] R. Xu and I. Wunsch, D., "Survey of clustering algorithms," Neural Networks, IEEE Transactions on, vol. 16, no. 3, pp. 645–678, May 2005

[4] M. K. Jiawei Han, Data Mining: Concepts and Techniques. Academic Press, Morgan Kaufmarm Publishers, 2001.

[5] Bandyopadhyay, Samir Kumar, and Tuhin Utsab Paul. "Segmentation of brain tumour from mri image analysis of k-means and dbscan clustering." *International Journal of Research in Engineering and Science* 1, no. 1 (2013): 48-57.

[6] Chebbout, Samira, and Hayet Farida Merouani. "Comparative study of clustering based colour image segmentation techniques." In *2012 Eighth International Conference on Signal Image Technology and Internet Based Systems*, pp. 839-844. IEEE, 2012.

[7] Ajala Funmilola, A., O. A. Oke, T. O. Adedeji, O. M. Alade, and E. A. Adewusi. "Fuzzy kc-means clustering algorithm for medical image segmentation." *Journal of Information Engineering and Applications, ISSN* 22245782 (2012): 2225-0506.

[8] Dhanachandra, Nameirakpam, Khumanthem Manglem, and Yambem Jina Chanu. "Image segmentation using K-means clustering algorithm and subtractive clustering algorithm." *Procedia Computer Science* 54 (2015): 764-771.

[9] Norouzi, Alireza, Mohd Shafry Mohd Rahim, Ayman Altameem, Tanzila Saba, Abdolvahab Ehsani Rad, Amjad Rehman, and Mueen Uddin. "Medical image segmentation methods, algorithms, and applications." *IETE Technical Review* 31, no. 3 (2014): 199-213.

[10] Abdel-Maksoud, Eman, Mohammed Elmogy, and Rashid Al-Awadi. "Brain tumor segmentation based on a hybrid clustering technique." *Egyptian Informatics Journal* 16, no. 1 (2015): 71-81.

[11] Adhikari, Sudip Kumar, Jamuna Kanta Sing, Dipak Kumar Basu, and Mita Nasipuri. "Conditional spatial fuzzy C-means clustering algorithm for segmentation of MRI images." *Applied soft computing* 34 (2015): 758-769.

[12] Folkesson, Jenny, Julio Carballido-Gamio, Felix Eckstein, Thomas M. Link, and Sharmila Majumdar. "Local bone enhancement fuzzy clustering for segmentation of MR trabecular bone images." *Medical physics* 37, no. 1 (2010): 295-302.

[13] Aprovitola, Andrea, and Luigi Gallo. "Knee bone segmentation from MRI: A classification and literature review." *Biocybernetics and Biomedical Engineering* 36, no. 2 (2016): 437-449.

[14] Pham, Dzung L., Chenyang Xu, and Jerry L. Prince. "Current methods in medical image segmentation." *Annual review of biomedical engineering* 2, no. 1 (2000): 315-337.

[15] Zhang, Dao-Qiang, and Song-Can Chen. "A novel kernelized fuzzy c-means algorithm with application in medical image segmentation." *Artificial intelligence in medicine* 32, no. 1 (2004): 37-50.

[16] Castellanos, Ramiro, Hiedra Castillo, and Sunanda Mitra. "Performance of nonlinear methods in medical image restoration." In *Nonlinear Image Processing X*, vol. 3646, pp. 252-260. International Society for Optics and Photonics, 1999.

[17] Castellanos, Ramiro, and Sunanda Mitra. "Segmentation of magnetic resonance images using a neuro-fuzzy algorithm." In *Proceedings 13th IEEE Symposium on Computer-Based Medical Systems. CBMS 2000*, pp. 207-212. IEEE, 2000.

[18] Al-Dmour, Hayat, and Ahmed Al-Ani. "MR brain image segmentation based on unsupervised and semi-supervised fuzzy clustering methods." In *2016 international conference on digital image computing: techniques and applications (DICTA)*, pp. 1-7. IEEE, 2016.

[19] L. Juang, and M. Wu,"MRI brain lesion image detection based on color-converted K-means Clustering segmentation,"Measurement, Vol. 43, no. 7,pp. 9419, 2010

[20] Zahid Ansari, Waseem Ahmed, M.F. Azeem and A. Vinaya Babu. "Discovery of Web Usage Profiles Using Various Clustering Techniques", *International Journal of Computer Information Systems*, ISSN: 2229 5208, vol. 1, no. 3, pp. 18-27, July 2011.

[21] Zahid Ansari, A. Vinaya Babu, Waseem Ahmed and M. F. Azeem, "A Comparative Study of Mining Web Usage Patterns Using Variants of k-Means Clustering Algorithm", *International Journal of Computer Science and Information Technologies*, ISSN: 0975-9646, vol. 2 no. 4, pp. 1407-1413. July 2011.

[22] Zahid Ansari, "Web User Session Cluster Discovery Based on k-Means and k-Medoids Techniques", *International Journal of Computer Science & Engineering Technology (IJCSET)*, ISSN:2229-3345, vol 5, no. 12, pp. 1105-1113,. December 2014.

[23] Zahid Ansari and Amjad Khan, "Fast Global k-Means Method To Discover User Session Clusters from Web Log Data", *International Journal of Computer Engineering and Applications (IJCEA)*, ISSN:2321-3469, vol. 8 no. 3, pp. 26-35. Dec.2014.

[24] Zahid Ansari, Mohammed Tajuddin, Syed Ab. Sattar, "Discovery of Web User Session Clusters Using Partitioning Based Clustering Techniques", *International Journal of Computer Technology and Applications (IJCTA)*, ISSN: 2229-6093 , vol 5, no. 6. , pp. 2049-2056, Nov-Dec 2014

[25] Zahid Ansari, Mohammed Fazle Azeem, A. Vinaya Babu, and Waseem Ahmed. Preprocessing users web page navigational data to discover usage patterns. In The Seventh International Conference on Computing and Information Technology, Bangkok, Thailand, May 2011.

[26] Zahid Ansari, "Discovery of Web User Session Clusters Using DBSCAN and Leader Clustering Techniques", *International Journal of Research in Applied Science & Engineering Technology (iJRASET)*, ISSN: 2321-9653 vol 2, no. 12, pp. 209-207. December 2014

[27] D.L. Davies, D.W. Bouldin. A cluster separation measure. 1979. IEEE Trans. Pattern Anal. Machine Intell. 1 (4). 224-227.

[28] Hubert, L. and Schultz, J. Quadratic assignment as a general data-analysis strategy. *British Journal of Mathematical and Statistical Psychology*, 29, 190-241, 1976.

[29] M.K. Pakhira, A Modified k-means Algorithm to Avoid Empty Clusters, *International Journal of Recent Trends in Engineering* Vol 1, No. 1. 2009.