ISSN (e): 2250 – 3005 || Volume, 08 || Issue, 6|| Jun – 2018 ||
International Journal of Computational Engineering Research (IJCER)

# Optimal Centroid Estimation by Automatic Parameter Initialization for Partition Based Clustering Algorithm

**D.Ashok Kumar[1] , P. Velayudham[2]**
*1 Head, Department of Computer Science, Government Arts College, Trichy-620022.*
*2Department of Computer Science, Government Arts College, Trichy-620022.*
*Correspondence Author: D.Ashok Kumar*

**ABSTRACT**
Mining knowledge from large amounts of data is known as data mining. The k-means algorithm is the efficient partition clustering algorithm in the large data sets. This algorithm is iteratively partitions a large dataset into k groups in the area of starting points such that define the objective function in the terms of minimize the sum of squares for total data the solutions is locally optimal. Thus the algorithm is success in finding globally optimal partitions depends on starting values. The main challenge of partition algorithm is to set the initial parameter. The automatic parameter initialization approach has been proposed by fast partition algorithm with efficient centroid estimation for the high dimensional data. The proposed method a detailed assessment of the performance and commonly used well performed parameter initializing methods over the datasets of higher dimensions, number of observations for the Square errors, groups and clustering complexity. Performance is evaluated both in the terms of the ability to best recovery of true groupings as well as the achieved minima of the objective function. It is based on shifting the center of the large cluster toward the small cluster, and re-computing the membership of small cluster points, the experimental results reveal that the proposed algorithm produces satisfactory results.
**KEYWORDS:** Clustering Analysis, Partition Clustering, K-Means Algorithm, High Dimensional Data, Automated Clustering.

---

Date of Submission: 21-05-2018 Date of acceptance: 05-06-2018

---

## I INTRODUCTION

Clustering is a key task in the process of unsupervised classification. The classification means to follows a procedure that assigns data objects to a set of classes. Unsupervised means clusters does not depends on the predefine classes. Thus the complexity of diversity has been forced society to organize things based on their similarities. This is done such that patterns in the same cluster are alike, and patterns belonging to two different clusters are different. Clustering has been a widely studied problem in a variety of application domains including data mining and knowledge discovery, data compression and vector quantization, pattern recognition and pattern classification [4], neural networks, artificial intelligence, and statistics. Clustering algorithms can be broadly classified into hierarchical and partitioning clustering algorithms [12], [14]. Hierarchical algorithms decompose a database D of n objects into several levels of nested partitioning, represented by a Dendrogram, i.e., a tree that iteratively splits D into smaller subsets until each subset consists of only one object. There are two types of hierarchical algorithms [17]; an agglomerative that builds the tree from the leaf nodes up it means bottom up, whereas a divisive builds the tree from the top down. Partitioning algorithms construct a single partition of a database D of n objects into a set of k clusters. Optimization based partitioning algorithms typically represent clusters by a prototype [5]. Objects are assigned to the cluster represented by the most similar prototype. An iterative control strategy is used to optimize the whole clustering such that, the average squared distances of objects to its prototypes are minimized. The prototypes, one can distinguish the different k-series partition algorithms (k-means, k-modes and k-medoids). In k-means algorithm the prototype called the centroid is the mean value of all objects belonging to a cluster. It has minimal objective function thus the added advantage for k-means is a widely studied partitioning algorithm.

The k-means have the number of advantages over other clustering techniques, it also has some drawbacks; it converges often at a local optimum [1], the final result depends on the initial starting centers. Many researchers introduce some methods to select good initial starting centers [1],[3],[4],[18],[19]. Other researchers try to find

the best value for the parameter k that determines the number of clusters or the value of k must be supplied by the user [8] and [10]. In recent years, many improvements have been proposed and implemented in the k-means algorithm with different centroid estimation models called random selection, mean distance and inter cluster distance [2]. Takayasu [5] proposed a k-means for spherical Cluster. This is fully based on size of the cluster. Thus we need to split the large cluster into small cluster. This is for a time consuming process and a chance to missing the data. Z Khan [7] proposed an enhanced k-means to improve the accuracy and efficiency of the k-means clustering algorithm. This is two step processes the first step is find the initial center and the second step is to assign the data points to cluster. It is very critical to assign the data points and more time consuming process. H Ismkhan [4] proposed a method to find better initial centroid selection and reduced time complexity. But it does in work well in high dimensional data. The computational complexity of the original k-means algorithm is very high, specifically for massive data sets [1],[3]. Various methods have been proposed in the literature to enhance the accuracy and efficiency of the k-means clustering algorithm. The author proposed an enhanced method to overcome the above said problems for finding the better initial centroid and to provide an efficient way of assigning the data points to suitable clusters with reduced time complexity for the high dimensional datasets. It means the proposed algorithm produce a good results with limited time pried.

The proposed partition clustering algorithm attempts to determine k partitions that optimize a new criterion function with any one of the centroid estimation [2] method. The average square error criterion, defined in (1), is the most commonly used ($m_i$ is the mean of cluster $_{Ci}$, n is the number of objects in the dataset and x is the selected object) well define objective function.

$$E = {}^{1}/_{n} \sum_{i=1}^{k} \sum_{ixc_j} (x - m_{i)})^2 \qquad (1)$$

The average square-error is a good measure of the within cluster variation across all the partitions. Thus, the average square error clustering tries to make the k clusters as compact and separated as possible, and works well when clusters are compact clouds that are rather well separated from one another . The square-error method could split large clusters to minimize the square-error. The efficiency of the original k-means algorithm heavily relies on the initial centroid. The k-means clustering algorithm required the cluster count(k) as the main input data. The centroid values are used to measure the transaction relevancy. There are different centroid optimization techniques [5] are used. The proposed system is design to partition clustering algorithm with different centroid estimation models with include the average square error criterion function. Here we will review partition clustering algorithms and some variants of it in section II, discuss the proposed idea in section III, present some experimental results in section IV and conclude with section V.

## II CLUSTERING ANALYSIS

The k-means algorithm, we assign each item to the cluster whose centroid is nearest distance is calculated by using the Euclidean distance [2],[5],[9]. The time complexity of this progress is O(nkj). n refers to the number of total items, and k refers to the number of clusters initially set and j refers the dimension of data objects. The original k-means is a time consuming process for the large data sets. The k-means algorithm uses the mean value of the objects in a cluster as the cluster center. Suppose that a dataset of n objects x1, x2 ,..., xn such that each object is in Rd , the problem of finding the minimum variance clustering of the dataset into k clusters is that of finding k points mi , i = 1, 2,..., k , in Rd such that Equation (1) is minimized. The basic processes of the k-means algorithm:

1. Initialization: Select a set of k starting points mj , j = 1, 2,….. k. the selection may be done in random manner or according to some heuristic.
2. Distance calculation: For each object xi, $1 \leq i \leq n$ compute its Euclidean distance to each cluster centroid mj , $1 \leq j \leq k$, and then find the closest cluster centroid.
3. Centroid recalculation: For each $1 \leq j \leq k$ recomputed cluster centroid mj as the average of the data points assigned to it.
4. Convergence condition: Repeat step 2 and 3 until convergence. Before the k-means algorithm converges, step2 and step3 are executed number of times, say j, where the positive integer j is known as the number of k-means iterations. The precise value of j varies depending on the initial starting clusters centroid even on the same data set. The computational time complexity of the algorithm is O(nkj), Where n is the total number of objects in the dataset, k is the required number of clusters we identified j is the number of iterations, $k \leq n$, $j \leq n$.

The k-means algorithm can be thought of as a gradient descent procedure which begins at the starting clusters centroids and iteratively updates these centroids to minimize the objective function in equation (1). It is known that, k-means will always converge to a local minimum. When we analyze the k-means we find that, the main advantages of this algorithm are; (i) its efficiency, (ii) this algorithm is very easy to implement and (iii) speed of convergence. On the other hand, its main drawbacks are (i) the final result depends on the initial starting centers, (ii) to choose a proper number of clusters k is a domain dependent problem, (iii) this algorithm is applicable

only when mean is defined, (iv) it is sensitive to outliers and (v) this algorithm is Good only for convex shaped, similar size and density clusters. For the first four disadvantages, there are a lot of efforts have been done to overcome these problems, we review of them, the proposed method handles the above problems in several variants of the k-means algorithm have been proposed. The purpose is to improve efficiency or find better clusters; improved efficiency is usually accomplished by either reducing the number of iterations to reach final convergence or reducing the total number of distance calculations. Therefore, choosing a good set of initial cluster centers is very important for the algorithm. However, it is difficult to select a good set of initial cluster centers randomly. Bradley and Fayyad [17] have proposed an algorithm for refining the initial cluster centers. The main idea of their algorithm is to select m subsamples from the data set, apply the k-means on each subsample independently, keep the final k centers from each subsample provided that empty clusters are not be allowed, so they obtain a set contains mk points to apply the k- means on this set of m times; at the first time, the first k points are the initial centers. At the second time, the second k points are the initial centers, and so on and the algorithm returns the best k centers from this set. To choose a proper number of clusters k is a domain dependent problem. To resolve this problem, some methods World Academy of Science, Engineering has been proposed to perform k-clustering for various numbers of clusters and employ certain criteria for selecting the most suitable value of k. [12] and [11]. For example in [11] the authors depend on the fact that, the k-means method aims to minimize the sum of squared distances from all points to their cluster centers, this should result in compact clusters. So they use the distances of the points from their cluster centers to determine whether the clusters are compact or not. For this purpose, they use the intra-cluster distance measure, which is simply the distance between two points and its cluster center and take the average of all of these distances, as defined in equation (2).

$$\text{Intra} = {}^{1}/_{n} \sum_{i=1}^{k} \sum_{ixc_j} (x - m_{ij})^2 \qquad (2)$$

Thus the intra-cluster distance is the average squared error that the k-means method minimizes and also the author measure the inter-cluster distance, or the distance between clusters, which should to be as large as possible. To calculate this as the distance between cluster centers, and take the minimum of these values, defined as in equation (3)

$$\text{Inter} = \min\left(\|m_i - m_j\|^2\right) \qquad (3)$$

i =1, 2 ,..., k −1 and j = i +1 ,..., k

To take only the minimum of these values as want the smallest of this distance to be maximized, use these measures to help them to determine if they have a good clustering, thus minimize the ratio between them, defined as in equation (4).

Validity=Intra/Inter (4)

Therefore, the clustering which gives a minimum value for the validity measure will give the ideal value of k in the k-means algorithm. This algorithm consists of two consecutive stages, which are repeated several times. In the first stage, they perform k-means algorithm with multiple initial starting points, and pick the best centers, and in the second stage, the algorithm assign a factor for each point and iteratively removes the points which are far from their clusters centers.

## III PARTITION CLUSTERING ALGORITHM

The aim of clustering procedures is to partition a heterogeneous multi-dimensional data set into groups of more homogenous characteristics. The formation of clusters is based on the principle of maximizing similarity between patterns of the same cluster and simultaneously minimizing the similarity between patterns belonging to distinct clusters. Similarity or proximity is usually defined as a distance function on pairs of patterns and based on the values of the features of these patterns.

The number of clusters k is an input to the k-means clustering algorithm. Clusters are described by centroids which are cluster centers, in the algorithm. In our implementation of k-means the initial centroids consist of the clustering results from average of square errors with the closest Euclidean distance. K-means algorithm is a partitioning prototype based supervised clustering algorithm. The input of k-means algorithm is a dataset consisting of n vectors and if output is k-cluster which together to form a mutually exclusive and extensive partitioning of the dataset. The k- means algorithm expects the user to specify k. so it is called as a supervised clustering algorithm. The k-means algorithm is simple and has nice convergence but there are number of problems to make the clustering. The k-means algorithm is applicable only when mean is defined, also this problem is solved by introducing the k-modes algorithm [7]. This is an extended version of the k-means with some modification to be suitable for categorical data. The cause that the k-means algorithm cannot cluster categorical objects is its dissimilarity measure and the method used to solve the clustering problem. These barriers have been removed by making the following modifications to the k-means algorithm. 1). It used a simple matching dissimilarity measure for categorical objects. 2). Replacing means of clusters by modes. 3).

Using a frequency-based method to find the modes to solve the problem. Thus the k-means algorithm is more sensitive.

**Algorithm 1: Optimal Centroid Estimation for k-means Clustering Algorithm**

Require: D = {d1, d2, d3,..., di,..., dn } // Set of n data points.
di = { x1, x2, x3,..., xi,..., xm } // Set of attributes of one data point.
k // Number of desired clusters.
Ensure: A set of k clusters.
Steps:
1: In the given data set D, if the data points contain the both positive and negative attribute values then go to step 2, otherwise go to step 4.
2: Find the minimum attribute value in the given data set D.
3: For each data point attribute, subtract with the minimum attribute value.
4: For each data point calculate the distance from origin.
5: Sort the distances obtained in step 4. Sort the data point's accordance with the distances.
6: Partition the sorted data points into k equal sets.
7: In each set, take the middle point as the initial centroid.
8: Compute the distance between each data point di (1<= i<= n) to all the initial centroids cj (1 <= j <= k).
9: Repeat.
10: For each data point di, find the closest centroid cj and assign di to cluster j
11: Set Cluster Id[i]=j. // j:Id of the closest cluster.
12: Set Nearest Dist. [i]= d(di, cj).
13: For each cluster j (1 <= j <= k), recalculate the centroids.
14: For each data point di,
14.i). Compute its distance from the centroid of the present nearest cluster.
14.ii). If this distance is less than or equal to the present nearest distance, the data point stays in the same cluster.
Else
14.ii.a) For every centroid cj (1<=j<=k) compute the distance d(di,cj).
End For;
Until the convergence criteria is met.

**Algorithm 2: Proposed Optimal Centroid Estimation for k-means Algorithm**

Require: D = {d1, d2, d3,..., di,..., dn } // Set of n data points.
di = { x1, x2, x3,..., xi,..., xm } // Set of attributes of one data point.
k // Number of desired clusters.
Ensure: A set of k clusters.
Steps:
1. In the given data set D, if the data points contain the both positive and negative attribute values then go to step 2, otherwise go to step 4.
2.: Find the minimum attribute value in the given data set D.
3: For each data point attribute, subtract with the minimum attribute value.
4: For each data point calculate the distance from origin.
Divide the distance from sum of square error, and take the average distance.
5: Sort the distances obtained in step 4. Compare the average with the Sort the data points.
6: If average distance<=distance then assign the average distance as an initial centroid.
Compute the distance between each data point di
(1<= i<= n) to all the initial centroids cj(1<= j<= k).
Else
Set Cluster Id[i]=j. // j:Id of the closest cluster.
Set Nearest Dist.[i]= d(di, cj).
For each cluster j (1 <= j <= k), recalculate the centroids.
Compute its distance from the centroid of the present nearest cluster.
Divide the distance from sum of square error and take the average distance.
End If
Repeat until the convergence criterion is met.

## IV EXPERIMENTAL RESULTS

### DATASETS DESCRIPTION

Iris Plants Database: Iris is a most useful UCI repository dataset in data mining. The iris data set consists of three classes.

a. Iris setosa.
b. Iris versicolour
c. Iris virginica

Latter are not linearly separable from each other. Predicted attributes are class of iris plant. Number of instances are 150 for 50 in each of three classes. This dataset have the four attributes. Now to present some experimental evaluation of the proposed algorithm compare the existing algorithm, which reveal a great improvement in the k-means algorithm when the dataset contains large variance in their size. We find that, the existing method produces the exact clusters in dataset 1 and dataset 2, because there is a separation between clusters. But there is an error at clusters discovered from dataset 3, note that, for the existing system there is no separation between clusters for large data set. But our proposed system has good separation for high dimensional data with leads to less time. From Table 1 shows, there are some points misclassified.

**Table 1 comparison between the results from the existing OCEP and the proposed OCEP method.**

| Data sets | Actual Cluster Objects | Exist OCEP Clusters Objects | Proposed OCEP Clusters Objects | Process Time/ seconds | |
|---|---|---|---|---|---|
| | | | | Exist OCEP | Proposed OCEP |
| Data Set1 | 110 | 110 | 110 | 0.9384 | 0.00890 |
| | 45 | 45 | 45 | | |
| | 45 | 45 | 45 | | |
| Data Set2 | 88 | 88 | 88 | 0.9084 | 0.0063 |
| | 56 | 56 | 55 | | |
| | 56 | 56 | 56 | | |
| Data Set3 | 90 | 92 | 92 | 0.9029 | 0.0059 |
| | 60 | 55 | 55 | | |
| | 50 | 53 | 53 | | |

The qualitative evaluation of quantitative results are done using the best metrics for data clustering like Inter-cluster distance, Intra-cluster distance, elapsed time for clustering. Inter-cluster distance μ means the distances between different clusters, and it should be maximized i.e distance between their centroids. The inter-cluster distance μ for K clusters $C_1, C_2,\ldots,C_K$ with centroids $Z_i$, i=1...K is given in equation (5).

$$\mu(C_1, C_2, \ldots C_k) = \sum_{i=1}^{k} \sum_{j=i+1}^{k} |Z_i - Z_j| \qquad (5)$$

Intra-cluster distance $\upsilon$ is the sum of distances between objects in the same cluster, and it should be minimized. The intra-cluster distance μ for K clusters $C_1, C_2, .. C_K$ centroid $Z_i$, i=1..K is given in equation (6).

$$\vartheta(C_1, C_2, \ldots C_k) = \sum_{i=1}^{k} \sum_{j=i+1}^{k} X_j C_j |X_i - Z_j| \qquad (6)$$

### TIME COMPLEXITY

The Time complexity of K-means algorithm is O(nkj) Where n is the number of data points , k is the number of cluster and the j is the number of iterations. As per the proposed method the average square error reduce the time complexity half of our existing O(nkj). The cluster processing time is counted in the term of seconds.

### CLUSTER VALIDATION

$$d_{min} = min_{ij} \|z_i - z_j\| \qquad (7)$$

$d_{min}$ is the minimum distance between the cluster centers. Where n is the number of objects, k is the number of clusters, and $Z_i$ is the cluster centre of cluster $C_i$, The smaller values is indicate that the clusters are more compact and larger is indicated the clusters are well separated.

From Table 2, it is observed that the Intra-cluster distance μ i i= 1,2,..,5 for various distance measures in proposed OCEP outperforms the existing OCEP. Three distance measure executions are done here because existing OCEP clustering vary based on the initialization of centroids for proposed OCEP clustering.

**TABLE 2: INTRA DISTANCE FOR 3 CLUSTERS**

| Distance Measure | μ1 | | μ2 | | μ3 | | Average μ | |
|---|---|---|---|---|---|---|---|---|
| | E-OCEP Clusters | P-OCEP Clusters | E-OCEP Clusters | P-OCEP Clusters | E-OCEP Clusters | P-OCEP Clusters | E-OCEP clusters | P-OCEP clusters |
| Sq.Eucli | 118.9900 | 118.8041 | 118.8047 | 119.2190 | 119.6118 | 118.9901 | 119.1355 | 119.0044 |
| CB | 119.5043 | 118.9586 | 119.5043 | 118.9586 | 118.772 | 118.6962 | 119.2602 | 118.8711 |
| Euclidean | 118.8047 | 118.9291 | 118.8047 | 119.2872 | 119.618 | 118.9901 | 119.0758 | 119.0688 |

During OCEP clustering the Squared Euclidean Distance, City Block Distance and Euclidean Distance are used to prove the performance of clustering. Total average intra-distance for all types of distance measures of existing OCEP squared Euclidean is 119.1355 and 119.0044 for proposed OCEP that means 0.1311 smaller for μ on an average for proposed OCEP, the distance measure of existing OCEP for City block is 119.2602 and 118.8711 for proposed OCEP that means 0.3891 smaller for μ on an average for proposed OCEP and the distance measure of existing OCEP for Euclidean is 119.0758 and 119.0688 for proposed OCEP that means 0.007 smaller for μ n an average for proposed OCEP. The proposed OCEP clustering is best compared to existing OCEP clustering is depicted in the Figure 1.

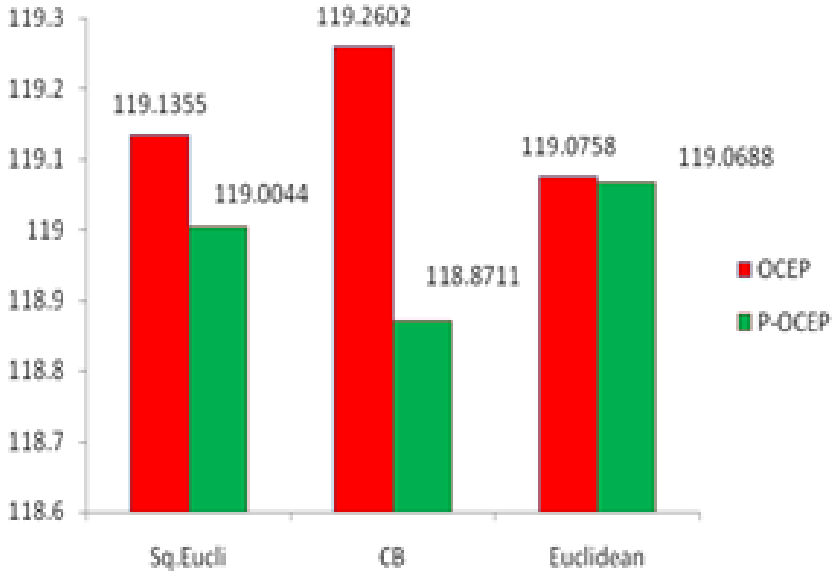


**Figure 1. Average performance of Intra distance for various dist**

From the figure 1, we can understand that proposed OCEP clustering is better than the existing OCEP clustering. The Inter distance for various distance measures on Iris database is listed in the following table 3.

**TABLE 3: INTER DISTANCE FOR 3 CLUSTERS**

| Distance Measure | $\upsilon_1$ | | $\upsilon_2$ | | $\upsilon_3$ | | Average $\upsilon$ | |
|---|---|---|---|---|---|---|---|---|
| | E-OCEP Clusters | P-OCEP Clusters | E-OCEP clusters | P-OCEP Clusters | E-OCEP Clusters | P-OCEP Clusters | E-OCEP clusters | P-OCEP Clusters |
| Sq.Eucli | 1.9499 | 1.8349 | 1.8660 | 1.9531 | 1.4344 | 1.7248 | 1.7501 | 1.8376 |
| CB | 1.5113 | 1.5810 | 1.2422 | 1.5611 | 1.8693 | 1.8624 | 1.5409 | 1.6681 |
| Euclidean | 1.7132 | 1.9641 | 1.6351 | 1.8816 | 1.7737 | 1.7248 | 1.7073 | 1.8568 |

The Table 3 shows that the Inter-distance $\upsilon$ for various distances in proposed OCEP out performs OCEP. Total average inter-distance for all types of distance measures of OCEP for squared Euclidean is 1.7501 and 1.8376 for proposed OCEP that means 0.0875 improvements for $\upsilon$ on an average for proposed OCEP, the distance measure of OCEP for City-block is 1.5409 and 1.6681 for proposed OCEP that means 0.1272 improvements for $\upsilon$ on average for proposed OCEP and the distance measure of OCEP for Euclidean is 1.7073 and 1.8668 for proposed OCEP that means 0.1595 improvements for$\upsilon$ on an average for proposed OCEP. The Improved OCEP clustering is best compared to existing OCEP clustering is depicted in the Figure 2.

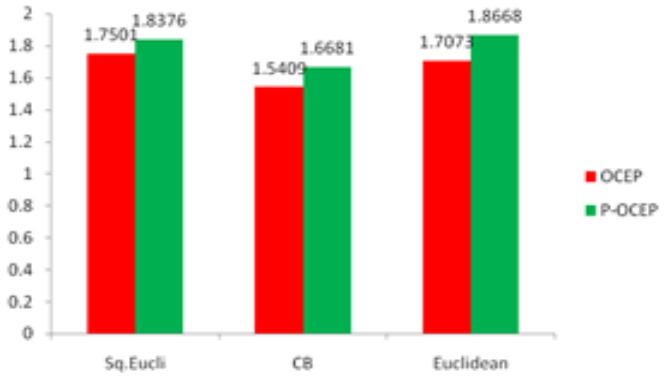


**Figure 2. Average for Inter distance**

From the Figure 2, we can understand that the inter distance for various distance measures are best for proposed OCEP compared to existing OCEP. The process time for executing the existing OCEP and proposed OCEP process time is tabulated in Table 4.

**TABLE 4: PROCESSING TIME IN SECONDS**

| **Distance Measure** | **$\mu_1$** | | **$\mu_2$** | | **$\mu_3$** | | **Average $\mu$** | |
|---|---|---|---|---|---|---|---|---|
| | **Exist OCEP Clusters** | **Proposed OCEP Clusters** | **Exist OCEP Clusters** | **Proposed OCEP Clusters** | **Exist OCEP Clusters** | **Proposed OCEP Clusters** | **Exist OCEP Clusters** | **Proposed OCEP Clusters** |
| Sq.Eucli | 1.0563 | 0.0126 | 0.9235 | 0.0061 | 1.0227 | 0.0066 | 1.00080 | 0.00843 |
| CB | 0.8878 | 0.0185 | 1.0269 | 0.0088 | 1.1328 | 0.0071 | 1.01583 | 0.01146 |
| Euclidean | 0.8839 | 0.0089 | 0.9029 | 0.0063 | 0.9384 | 0.0059 | 0.9084 | 0.00703 |

Table 4 shows that the average performance of processed time using various distance measures. Total average time for executing the OCEP for squared Euclidean is 1.00080 and 0.00843 for proposed OCEP that means the proposed OCEP executes 0.99237 faster than the OCEP, executing the OCEP for Cityblock is 1.01583 and 0.01146 for proposed OCEP that means the proposed OCEP executes 1.00437 faster than the OCEP and executing the OCEP for Euclidean is 0.9084 and 0.00703 for proposed OCEP that means the proposed OCEP executes 0.90137 faster than the Existing OCEP. These average performances are depicted in the Fig 3

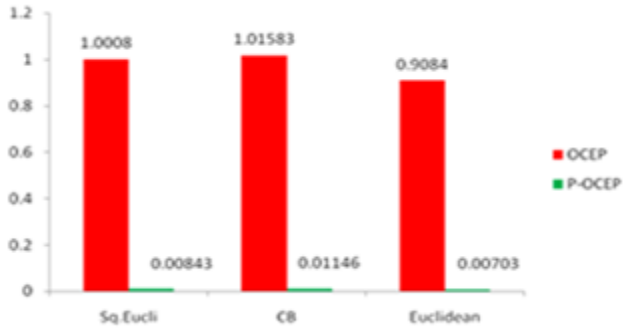


**Figure 3 Average process Time**

From the Figure. 3, we can understand that executing the proposed algorithm is much faster than the existing OCEP.

.

**TABLE (5) EXISTING OCEP VALIDITY MEASURE.**

| Distance Measure | Intra Cluster Distance | Inter Cluster Distance | Exist OCEP Validity |
|---|---|---|---|
| Sq – Equcli | 119.1355 | 1.7501 | 68.074 |
| CB | 119.2602 | 1.5409 | 77.396 |
| Euclidean | 119.0758 | 1.7073 | 69.745 |

**CLUSTER VALIDATION**

The cluster validation is done in based on the equation (4). The clustering which gives a minimum value for the validity measure will give the ideal value of k in the k-means algorithm

Table(5) shows that the existing OCEP algorithm validity is measure using various distance measures. As per Sq -Euclidean distance measure the cluster validity is 68.074, as per CityBlock the clustering validity is 77.396 and as per Euclidean distance measure the cluster validity is 69.396. The minimum clustering validity measure is the best for clustering partition. The validity status are depicted in the figure (4).

**TABLE( 6) - PROPOSED OCEP VALIDITY MEASURE.**

| Distance Measure | Intra Cluster Distance | Inter Cluster Distance | Proposed OCEP Validity |
|---|---|---|---|
| Sq-Equcli | 119.0044 | 1.8376 | 64.76 |
| CB | 118.8711 | 1.6681 | *71.26* |
| Euclidean | 119.0688 | 1.8568 | 64.12 |

Table 6 shows that the proposed OCEP algorithm validity is measure using various distance measures. As per Sq -Euclidean distance measure the clustering validity is 64.76, as per CityBlock the clustering validity is 71.26 and as per Euclidean distance measure the clustering validity is 64.12. The results of proposed OCEP have the minimum validity measure compare with existing OCEP. Thus the proposed OCEP outperform for the existing OCEP because the minimum clustering validity is the best for the clustering partition. The validity status are depicted in the figure(4)

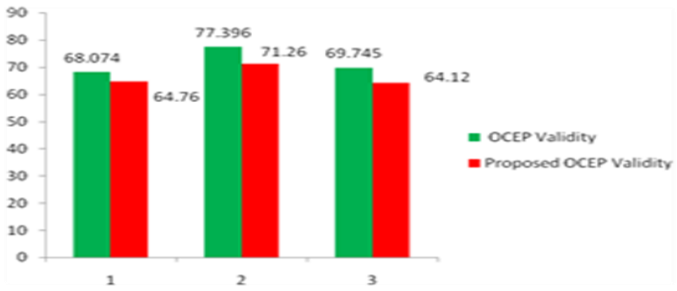


**Figure 4 Cluster Validity Measure**

## V CONCLUSION

In this paper, a new square error partition technique is added to the end of the OCEP algorithm. The objective of this procedure is to enhance the results of the OCEP. The experiment results are evidence for the proposed method improves the quality of clustering results with the limited time. Experiments are performed and the result shows that clustering data using the proposed OCEP is more efficient in terms of optimality measures and processing time than existing OCEP algorithm. This shows that the efficiency of the proposed partition clustering technique is superior to the existing OCEP algorithm, its widely used clustering technique where there is need to specify the required number of clusters. Further the authors are investigating the results of the proposed algorithm for real life data sets.

## REFERENCES

[1]. KC Gull, AB Angadi - ICT Based Innovations A Methodical Study on Behavior of Different Seeds Using an Iterative Technique with Evaluation of Cluster Validity-pp 63-74 , AISC, volume 653) - 2018 - Springer
[2]. P Das, DK Das, S Dey - A New Class Topper Optimization Algorithm with an Application to Data Clustering , IEEE Transactions on Emerging Topics in Computing, Issue-99- 2018.
[3]. D Gribel, T Vidal - HG-means: A scalable hybrid genetic algorithm for minimum sum-of-squares clustering arXiv preprint arXiv:1804.09813, Apr 2018.
[4]. H Ismkhan - Pattern Recognition, An iterative clustering algorithm based on an enhanced version of the k-means, Elsevier-- 2018 .
[5]. Takayasu Moriyaa, Holger R. Rotha, Shota Nakamurab, Hirohisa Odac, Kai Nagarac, Masahiro Odaa, and Kensaku Moria, Unsupervised pathology image segmentation using representation learning with spherical k-means, Medical Imaging-2018 - spiedigitallibrary.org.
[6]. Xiaoping Qin, Shijue Zheng, Ying Huang and Guangsheng Deng "Improved k-means algorithm and application in customer segmentation" Asia- Pacific Conference on Wearable Computing Systems, 2018.
[7]. Z Khan, J Ni, X Fan, P Shi, K-Means Clustering Algorithm based on Adaptive Initial Patrameter Estimation Procedure for Image Segmentation, International Journal of Innovative Research-.2017.
[8]. W Hu, J Gao, J Xing, C Zhang, Semi-Supervised Tensor-Based Graph Embedding Learning and Its Application to Visual Discriminant Tracking , ,- IEEE transactions - 2017 - ieeexplore.ieee.org.
[9]. Su M.-C. and C.-H. Chou, "A Modified Version of the k-means Algorithm with a Distance Based on Cluster Symmetry," IEEE Trans. Pattern Analysis and Machine Intelligence,vol.23,no.6,pp. 674-680, June 2015.
[10]. KS Gyamfi, J Brusey, A Hunt - K-Means Clustering using Tabu Search with Quantized Means- arXiv preprint arXiv:1703.08440, 2017 .
[11]. MM Mafarja, S Mirjalili , Hybrid Whale Optimization Algorithm with simulated annealing for feature selection Neurocomputing, 2017 - Elsevier , 2017.
[12]. K Ng, C De Filippi, WF Stewart, A Perer . Clustervision: Visual Supervision of Unsupervised Clustering- - IEEE transactions, 2018 - ieeexplore.ieee.org-
[13]. SAA Shah, U Nadeem, M Bennamoun. "Efficient image set classification using linear regression based image reconstruction , 2017 - openaccess.thecvf.com.
[14]. F Schwenker, E Trentin , Pattern classification and clustering: A review of partially supervised learning approaches. 2014 - Elsevier
[15]. N Dhanachandra, YJ Chanu – "A survey on image segmentation methods using clustering techniques , European Journal of Engineering Research- ejers.org." 2017.
[16]. Leung.Y, Zhang.J and Xu.Z."Clustering by Space-Space Filtering," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, no.12, pp. 1396-1410, 2010.
[17]. Koheri Arai and Ali Ridho Barakbah, "Hierarchical k-means: an algorithm for Centroids initialization for k-means," department of information science and Electrical Engineering Politechnique in Surabaya, Faculty of Science and Engineering, Saga University Vol. 36, No.1, 2014.
[18]. Chen Zhang and Shixiong Xia, " K-means Clustering Algorithm with Improved Initial center," in Second International Workshop on Knowledge Discovery and Data Mining (WKDD), pp. 790-792, 2015..
[19]. N Dhanachandra, YJ Chanu : A New Approach of Image Segmentation Method Using K-Means and Kernel Based Subtractive Clustering Methods - International Journal publication- 2017 .

D.Ashok Kumar." Optimal Centroid Estimation by Automatic Parameter Initialization for Partition Based Clustering Algorithm." International Journal of Computational Engineering Research (IJCER), vol. 08, no. 06, 2018, pp. 35-43.