A Comparative Study on Number of Clusters Determination and Efficient Centroid Initialization for K-Means Algorithm

Manal A. Abdel-Fattah
Information Systems Department
Faculty of Computers and Information, Helwan University
Cairo, Egypt
manal_8@hotmail.com

Yehia M. Helmy
Business Information Systems Department
Faculty of Commerce and Business Administration, Helwan University
Cairo, Egypt
ymhelmy@yahoo.com

Sara M.Mosaad
Business Information Systems Department
Faculty of Commerce and Business Administration, Helwan University
Cairo, Egypt
sara.mosaad87@gmail.com

***Abstract*— We live in a world were data are generated from different sources, and it is really very easy and cheap to collect and store such a huge amount of data. But the real benefit is not in the data itself, the benefit arises with the ability to extract valuable knowledge from it and the capability of processing such data in a tolerable elapse time with the machine learning algorithms. With the increasing need to analyze large amounts of data to get useful insights, it is essential to develop complex parallel machine learning algorithms that can scale with data and number of parallel processes, K-Means is one of the most popular and robust clustering algorithm. In spite of its wide use, K-means algorithm has certain drawbacks. the number of clusters which is not known in advance particularly for real world data sets and another drawback is the random selection of the cluster initial centroids. In this paper, we compare the different approaches and indices used in determining the optimal number of clusters along with the different ways for selecting the clusters' initial centers. Finally, comparative study is formulated which defines the major alternatives available to reach an efficient and stable k-means algorithm.**

Keywords*— *Data Clustering, k-means Algorithm, Number of Clusters, Initial Centroid Selection.

I. INTRODUCTION

In the past decade Data generation has seen tremendous growth and managing such huge amount of data is a big challenge. So, clustering, an unsupervised data mining technique can be applied in such circumstance effectively as it divides the data into smaller groups called clusters based on the level of similarity between the objects whereas the elements of one cluster are similar to each other, and dissimilar from the objects of other clusters. Arguably K-means is one of the most popular clustering algorithm that is simple to implement. It works in an iterative manner to assign data points of a dataset to the cluster centers which are chosen randomly, and by the end of each iteration the cluster center is updated. The process continues for a number of iterations until there is no change in the centers [1]. Although of the simplicity and robustness of the k-means algorithm, it is not free from weakness such that; The number of cluster must be determined beforehand randomly. So the user need to specify the number of clusters, Different initial centroids lead to different clustering results, as by using the same data, if it is inputted in a different way it may produce different cluster. Unable to handle noisy data and outlier. It Cannot be applied directly to categorical data it is applied only on numerical data. That's why studying its properties is of interest not only to the machine learning, data mining or classification communities but also important to the increasing numbers of practitioners in fields of bioinformatics, marketing research, customer management, big data and other application areas. And as different initial k objects lead to different clustering results even by using the same data along with the majority of the hierarchical algorithms have quadratic or higher complexity in the number of data points so they are not suitable for large datasets, while the partitioning algorithms have lower complexity. So, In this paper two of the most controversial issues in clustering is addressed where, a comparison of various options for selecting

the optimal number of clusters and how to select the cluster's initial centroid.in the most popular partitioning method, K-Means is addressed. specially on the initial centroid selection methods with linear time complexity N. as the k-means clustering algorithm itself has linear complexity N, which is the most important reason for its popularity. That's why an initial centroid selection method for k-means should not diminish this advantage of the k-means algorithm.
The rest of the paper is organized as follows. Section 2 presents the background. Section 3 presents related work of k-means number of clusters determination methods and indices, Linear initial centroid selection methods. Section 4 presents recommendations for practitioners. Section 5 presents the conclusion and future work.

II. BACKGROUND

A. Clustering

Clustering of objects is as ancient as the human need for describing the salient characteristics of men and objects and identifying them with a type and therefore it embraces various scientific disciplines starting from mathematics and statistics to biology and genetics, and each of them uses different terms in order to describe the topologies formed using this analysis. From biological “taxonomies”, to medical “syndromes” and genetic “genotypes” to manufacturing "group technology". the problem is identical which is forming categories of entities and assigning individuals to the proper groups within it [2]. In other words, Clustering is a technique of finding similar characteristics among the data set which are always hidden in nature and grouping them into groups, called as clusters [3].

B. K-means Algorithm

K-means clustering algorithm is a kind of partitioning clustering methods, a typical distance-based clustering algorithm, using distance as similarity evaluation. K-means algorithm is simple for large-scale data mining with high efficiency and scalability and fast with a more intuitive geometric meaning. It has been widely used in pattern recognition, image processing and computer vision. At the same time, the satisfactory results are obtained [4]. It performs the following steps to form the clusters [5]

K-means Algorithm

Input: k: the number of clusters or groups **D:**data set of 'n' objects

Output: Formed k clusters.

Algorithm:

1. Input k value and dataset.
2. If k == 1, then Exit.
3. Else
4. Select k objects from D to the closest centroid.
5. Assign each point d_1 in D to the closest centroid.
6. Calculate and update new cluster centroids.
7. Repeat from step 5 until centroids no longer move.

Here are the strength and weakness points of the k-means algorithm [6][7]:

Strength of k-Means algorithm:

i. Relatively efficient O(knt) where k is the number of clusters, n is the number of objects, t is the number of iteration.
ii. Very easy to understand and implement.
iii. Objects automatically assigned to its clusters.
iv. Often it terminates at local optimum.

Weakness of k-Means algorithm:

i. The number of cluster, K, must be determined beforehand. So the user need to specify k (number of clusters)
ii. We never know the real cluster, using the same data, if it is inputted in a different way it may produce different cluster. So different initial k objects lead to different clustering results.
iii. Unable to handle noisy data and outlier.
iv. Not suitable for non-convex shapes.
v. Cannot be applied directly to categorical data only numerical data.

vi. When the numbers of data are not so many, initial grouping will determine the cluster significantly.
vii. We never know which attribute contributes more to the grouping process since we assume that each attribute has the same weight.

This paper focus on two issues; first, how to get the optimal number of clusters for a dataset. Second, how to select the initial centroids in an optimal way

III. Related Work

With the tremendous growth in big data technology, it became hard to handle the complex big data by using the traditional algorithms. So learning algorithms are required to be enhanced in order to handle the huge and heterogeneous datasets. As the results obtained from such analytical techniques provide effective solutions for many real world problems in various domains such as banking, healthcare, agriculture, etc. Various researches are conducted to gather information about enhancements in k-means algorithm. This research gives an overall idea about the enhancements in k-means clustering algorithms specially in explaining the different ways of determining the optimal number of clusters and the different ways for selecting the cluster initial centroids.

Determining the optimal number of clusters in a data set is a fundamental issue in partitioning clustering, such as k-means clustering, which requires the user to specify the number of clusters k to be generated. But unfortunately there is no definitive answer to this question as the optimal clustering is somehow subjective and depends on the method used for measuring similarities and the parameters used for partitioning. several approaches have been proposed to determine the number of clusters for k-mean clustering algorithm. We focus on three different approaches; variance-based approach, structural approach, consensus approach Approaches for determinting the optimal number of clusters[8]

This section focuses on comparison of representative set of methods for determining the number of clusters for K-Means algorithm.
K-Means is an unsupervised clustering algorithm which is applied to a data set of N entities (I) each entity has a set of features (V), and the entity-to- feature matrix $Y=(y_{iv})$, where $v \in V$ at entity $i \in I$. The algorithm produces a non-overlapping partition $S=\{S_1, S_2,\ldots, S_K\}$ which is referred to as clusters, each with a specified centroid from a set $C=\{c1, c2,\ldots, cK\}$.The below criterion, minimised by the method, is the within-cluster summary distance to centroids:

$$W(S, C)=\sum_{k=1}^{k} \sum_{i \in sk} d(i, c_k) \quad (1)$$

where d is typically the Euclidean distance (in the former case is referred to as the **square** error criterion) or the Manhattan distance (in the former case is referred to as the **absolute** error criterion).
Variance based approach: comparing values W_K (the smallest square error criterion (W(S, C)) among those found at different K-Means initializations) at different K, among many indexes that is based on W_K to determine the number of clusters, we focus on the following four; as a representative set in our literature.

- Calinski and Harabasz present A Fisher-wise criterion (CH) [9][10][11][12][13] where the criterion showed the best performance in the experiments by Milligan and Cooper (1985), and was subsequently utilized by some authors for choosing the number of clusters This method finds K maximizing

  $$CH=((T-W_K)/(K-1))/(W_K/(N-K)) \quad (2)$$

  Where;

  $$T=\sum_{i \in I} \sum_{v \in V} y_{iv}^2 \quad (3)$$

  is the data scatter.

- Sugar and James present Jump Statistic method [9][10] [11][13] where it utilizes the criterion *W* which extended according to the Gaussian distribution model where It is supported with a mathematical derivation stating that if the data can be considered a standard sample from a mixture of Gaussian distributions at which distances between centroids are great enough, then the maximum jump would indeed occur at *K* equal to the number of Gaussian components in the mixture. The distance between an entity and centroid in criterion (1) is calculated as:

  $$d(i, c_k)=(y_i-c_k)^T \Gamma_k^{-1}(y_i-c_k) \quad (4)$$

  and

  $$d_k=(\sum_k \sum_{i \in Sk} d(i,S_k))/M*N \quad (5)$$

  where *k* is the within cluster covariance matrix. where **a transformation power, typically** *M/2*. Then the jump is defined as:

$$JS= d_k^{-M/2} - d_{k-1}^{-M/2} \quad (6)$$

where

$$d_0^{-M/2}=0 \quad (7)$$

Where the maximum jump *JS(K)* corresponds to the right number of clusters.

- Tibshirani, et al. present the Gap Statistic method [9][11][14][12][15]where it compares the value of criterion (1) with its expectation under the uniform distribution, it takes a range of *K* values and finds W_k for each *K*. Then in order to model the reference values, a number, *B*, of uniform random reference datasets over the range of the observed data are generated, so that criterion (1) values W_{Kb} for each $b=1,\ldots,B$ are obtained. Where gap is defined as:

$$Gap(k)=1/B\sum_b \log(W_{kb})-\log(W_{k)}. \quad (8)$$

Then the average is:

$$GK=1/B\sum_b \log(W_{kb}) \quad (9)$$

And its standard deviation is defined as:

$$sd_k=[1/B\sum_b (\log(W_{kb})-GK)^2]^{1/2} \quad (10)$$

leading to:

$$s_k=sd_k\sqrt{1+1/B} \quad (11)$$

Thus The estimate of the optimal number of clusters is the smallest *K* such that $Gap(K) \geqq Gap(K+1)_{-sK+1}$

- Hartigan presents A heuristic/Hartigan rule [9][10][11][12][13]that utilizes the intuition when clusters are well separated, then for K<K*, where K* is the "right number" of clusters, a (K+1) cluster partition should be the K-cluster partition with one of its clusters split in two. This would drastically decrease W_K. On the other hand, at K>K*, both K and (K+1) cluster partitions will be equal to the "right" cluster partition with some of the "right" clusters split randomly, so that W_K and W_{K+1} are not that different.

  Hartigan proposed calculating

$$HT=(W_K/W_{K-1}-1)(N-K-1) \quad (12)$$

where N is the number of entities. Increasing K from K=2 and find the k which HT is less than a threshold 10 (the threshold 10 here is "a crude rule of thumb" based on the intuition that if K is less than the "right number" of clusters, then a (K+1)-cluster partition should be equal to a K-cluster partition with one of its clusters split in two.)

Structural Approach: comparing values of another characteristic of the cluster structure **(Within-Cluster Cohesion Versus Between-Cluster Separation)**. we focus on the following method; as a representative in our literature.

- Kaufman and Rousseeuw present the Average Silhouette Width[11][14][16][12] [9][10] [13] that evaluates the relative closeness/tightness between individual entities to their clusters and separation from the rest. First, the silhouette width is calculated for each entity, then the average silhouette width for each cluster and finally the overall average silhouette width for the total clustering.
  For each $i \in I$, Calculate *a(i)* =which is the average dissimilarity between i and all other entities of the cluster to which i belongs, then *b(i)* is the minimum of the average dissimilarity between i and all the entities in other clusters, and

$$s(i)=\frac{b(i)-a(i)}{max(a(i),b(i)} \quad (13)$$

The silhouette width values lie in the range from –1 to 1. If the silhouette width value is close to 1, it means that the set I is well clustered. If the silhouette width value for an entity is about zero, it means that that the entity could be assigned to another cluster as well. If the silhouette width value is close to –1, it means that the entity is misclassified. The largest overall average silhouette width indicates the best number of clusters.

Consensus approach: relying on all R clusterings rather than on just the best one.

- Monti et al. [9][10][11][13] present the consensus distribution area method where the consensus matrix is calculated first *C(K)* which is an *N*x*N* matrix whose *(i,j)*-th entry is the proportion of those clustering runs in which the entities $i,j \in I$ are in the same cluster. An ideal situation is when the matrix contains 0's and 1's only: this is the case when all the *R* runs

lead to the same clustering. Consensus distribution is based on the assessment of how the entries in a consensus matrix are distributed within the 0-1 range.
First, for each k connectivity matrix $M^{(di)}$ is calculated where $M^{(di)}(i,j)=1$ if i and j belong to the same cluster, and 0, otherwise. Then calculate the consensus matrix

$$C^{(K)}(i,j)=\sum_{di} M^{(di)}(i,j)/R \qquad (14)$$

The cumulative distribution function (CDF) is defined over the range [0, 1] as follows:

$$CDF(x)=\frac{\sum_{i<j} 1\{C(K)(i,j) \le x\}}{N(N-1)/2} \qquad (15)$$

where 1{condition} denotes the indicator function that is equal to 1 when condition is true, and 0 otherwise.
then determine the area

$$A(K)=\sum_{i=2}^{m} (x_i-x_{i-1})\, CDF(x_i) \qquad (16)$$

and then the relative change in CDF area is calculated as the follow:

$$\Delta(K+1)=\begin{cases} A(K) & K=1 \\ \frac{A(K+1)-A(K)}{A(K)} & K \ge 2 \end{cases} \qquad (17)$$

Then K which maximizes $\Delta(K)$ is determined.

The index davdis is based on the entries of the consensus matrix C(k)(i,j) obtained from the consensus distribution algorithm. The mean and the variance of these entries μK and σK^2 for each K can be obtained, then

$$avdis(K)= \mu K * (1- \mu K) - \sigma K^2 \qquad (18)$$

Finally The index is defined as

$$davdis(K)=(avdis(K)-avdis(K+1))/avdis(K+1) \qquad (19)$$

The estimated number of clusters is decided by the maximum value of davdis(K).

A. Indecies for determining the optimal number of clusters

After introducing the previous methods for determining the optimal number of clusters, another question arises for how to assess the quality of the cluster as in real life, to find out an optimal number of clusters is very challenging with an objective to improve the cluster quality. So different internal and external criteria have been formulated to find out the optimal number of clusters in data sets and also to measure the cluster quality. In literature seven indices were included to predict the optimal number of clusters which are denoted as (*OptK*)[17].

- **Partition coefficient (PC)**

The optimal number of clusters (*OptK*) in a data set are decided through the maximum value of this index.

$$PC(c) = \frac{1}{N}\sum_{i=1}^{c}\sum_{k=1}^{N}\mu_{i,k}^2 \qquad (20)$$

where $\mu i,k$ is the membership of data point k in cluster i. It is a monotonic decreasing with c (no. of clusters) and no direct connection to the geometrical properties of data. It also ignores the additional parameters.

- **Classical entropy (CE)**

This index is one of the important index parameters to decide the optimal number of clusters (*OptK*) where the minimum value of CE indicates the optimal number

$$CE(c) = -\frac{1}{N}\sum_{i=1}^{c}\sum_{k=1}^{N}\mu_{i,k}\ln(\mu_{i,k}) \qquad (21)$$

where $\mu i,k$ membership of data point k in cluster i. N is the total number of data points.
It is a monotonic increasing with c(no. of clusters) and also the hardly detectable connection with the geometrical properties of data. It measures the fuzziness of the cluster partition.

- **Separation index (S)**

It uses the minimum distance concept for partition validity. This index takes into account the geometrical properties of data. The minimum value of separation index is the indicator of the optimal number of clusters (*OptK*) in a dataset.

$$S(c) = \frac{1}{N}\sum_{i=1}^{c}\frac{\sum_{k=1}^{N}\mu_{i,k}^{2}\ \|x_k - v_i\|^2}{N_{min_{i,j,i\neq j}}\|v_j - v_i\|^2} \quad (22)$$

$\mu i,k$ is the membership of data point k in cluster i. $d = \mu i,k\ xk$ - vi denotes the fuzzy deviation from data point xk to cluster i. vj - vi denotes the distance between cluster i and cluster j. N is the total number of data points.

- **Partition index (SC)**

It is the ratio between the sum of compactness and the separation of the cluster. The minimum value of SC denotes the optimal number of clusters (*OptK*).

$$S(c) = \frac{1}{N}\sum_{i=1}^{c}\frac{\sum_{k=1}^{N}\mu_{i,k}^{2}\ \|x_k - v_i\|^2}{\sum_{k=1}^{N}\mu_{i,k}\ \sum_{j=1,i\neq j}^{c}\|v_j - v_i\|^2} \quad (23)$$

where $\mu i,k$ is the membership of data point k in cluster i. d = q$\mu i,k\ xk$ - vi denotes the fuzzy deviation from data point xk to cluster i . vj - vi denotes the distance between cluster i and cluster j. N is the total number of data points. c is the total number of clusters.

- **Xie and Beni index (XB)**

It defines the inter-cluster separation as the minimum square distance between cluster centers, and the intra-cluster compactness as the mean square distance between each data object and its cluster center. The minimum value of this index denotes the optimal number of clusters (*OptK*) in a dataset.

$$XB(c) = \frac{\sum_{i=1}^{N}\sum_{j=1}^{c}\mu_{ij}^{2}\ \|x_i - c_j\|^2}{N.\min_{j,i\neq j}\left\{\|c_i - c_j\|^2\right\}} \quad (24)$$

where ci and cj are the center points of different clusters namely Ci and Cj respectively. N is the total number of data points in data set. c is the total number of clusters.

- **Davies-Bouldin index (DB)**

It is used to find out the optimal number of clusters in a dataset. So for every point, the similarity value to each cluster is calculated. This index combines the dispersion measure and dissimilarity measure of the cluster. This index determines farthest compact clusters. The minimum value indicates the optimal number of clusters (*OptK*).

$$DB(c) = \frac{1}{c}\sum_{i=1}^{c}\max_{j=1,\dots c,i\neq j}\left\{\frac{dia(c_i) + dia\ (c_j)}{\|c_i - c_j\|}\right\} \quad (25)$$

where c is the total number of clusters. The diameter of the cluster is defined:

$$dia(c_i) = \left\{\frac{1}{n_i}\sum_{x\epsilon c_i}\|x - c_i\|^2\right\}^{\frac{1}{2}} \quad (26)$$

Here c is the total number of clusters. ni is the total number of points and ci is the centroid of the cluster Ci. This index gives the minimum intra-cluster distance.

- **Calinski-Harabasz index (CH)**

It is the average inter-cluster and intra-cluster sum of squared distances. It is also used to find out the optimal number of clusters with faster computation than other indices in a data set. This index is used to maximize the dispersion between clusters and

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

minimize dispersion within clusters. The maximum value of this index denotes the optimal number of clusters (*OptK*) in a dataset and indicates compact and well-separated clusters.

$$CH(c) = \frac{trace(B_m)}{trace(W_m)} \times \frac{N-c}{c-1} \quad (27)$$

Where *Bm* is the between-cluster scatter matrix, *Wm* the internal scatter matrix, *N* is the total number of clustered samples and *c* the number of clusters.

$$W_m = \sum_{i=1}^{c} \sum_{x \in c_i} (x - c_i)(x - c_i)^T \; B_m = \sum_{i} n_i (c_i - k)(c_i - k)^T \quad (28)$$

Ci are the set of points in the cluster. *ci* is the center of the cluster *Ci* ,*ni* is the number of points of Cluster *Ci*. *k* is the center of the input data set

B. Methods for Initial Centroid Selection of k-means Algorithm with Linear Time Compelexity

linear methods are often random and/or order-sensitive, which renders their results unrepeatable[18]. In this section, we briefly review some of the commonly used initial centroid selection methods with linear time complexity (N) . From the determinism perspective, these methods can be divided into deterministic (Katsavounidis, PCA-Part and Var-Part) and non-deterministic methods , while from the order sensitivity perspective, these methods can be divided into order-sensitive(MacQueen's first method, Ball and Hall's method, Späth's method) and non-order sensitive (Forgy's method)

- **Forgy's method** assigns each point to one of the K clusters uniformly at random. The centers are then given by the centroids of these initial clusters. This method has no theoretical basis, as such random clusters have no internal homogeneity[19].
- **Jancey's method** assigns to each center a synthetic point randomly generated within the data space. Unless the data set fills the space, some of these centers may be quite distant from any of the points, which has a great deficiency point that it might lead to the formation of empty clusters [19]

- **MacQueen** proposed two different methods. The first one, which is the default option in the Quick Cluster procedure of IBM SPSS Statistics, takes the **first** K points in X as the centers. An obvious drawback of this method is its sensitivity to data ordering. The **second** method chooses the centers randomly from the data points. The rationale behind this method is that random selection is likely to pick points from dense regions, i.e., points that are good candidates to be centers. However, there is no mechanism to avoid choosing outliers or points that are too close to each other, thus the standard way of initializing k-means is the Multiple runs of this method [20]

- **Ball and Hall's method** takes the centroid of X, as the first center. Then it traverses the points in arbitrary order and takes a point as a center if it is at least T units apart from the previously selected centers until K centers are obtained. The purpose of the distance threshold T is to ensure that the seed points are well separated. However, it is difficult to decide on an appropriate value for T. In addition, the method is sensitive to data ordering [19][21].

- **Späth's method** is similar to Forgy's method with the exception that the points are assigned to the clusters in a cyclical fashion, i.e., the j-th ($j \in \{1, 2, \ldots, N\}$) point is assigned to the ($j - 1 \pmod K + 1$)-th cluster. Unlike Forgy's method, this method is sensitive to data ordering.

- **Maximin method** chooses the first center c1 arbitrarily and the i-th ($i \in \{2, 3, \ldots, K\}$) center ci is chosen to be the point that has the greatest minimum-distance to the previously selected centers, i.e., c1, c2, . . . , ci−1. This method was originally developed as a 2-approximation to the K-center clustering problem. It should be noted that, motivated by a vector quantization application, Katsavounidis et al.'s variant takes the point with the greatest Euclidean norm as the first center [21][19][20].

- **Al-Daoud's density-based method** uniformly partitions the data space into M disjoint hypercubes. It then randomly selects KN_m/N points from hypercube m ($m \in \{1, 2, \ldots, M\}$) to obtain a total of K centers (N_m is the number of points in hypercube m). Two main disadvantages are associated with this method. The first one, it is difficult to decide on an appropriate value for M. Second, the method has a storage complexity of $O(2^{BD})$, where B is the number of bits allocated to each attribute[19].

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

- **Bradley and Fayyad's method** starts by randomly partitioning the data set into J subsets. These subsets are then clustered using k-means initialized by MacQueen's second method producing J sets of intermediate centers each with K points. These center sets are combined into a superset, which is then clustered by k-means J times, each time initialized with a different center set. Members of the center set that give the least SSE are taken as the final centers [22][21].

- **Pizzuti et al.** improved upon Al-Daoud's density-based method using a multiresolution grid approach. Their method starts with 2D hypercubes and iteratively splits these as the number of points they receive increases. Once the splitting phase is completed, the centers are chosen from the densest hypercubes [19].

- The **k-means++ method** interpolates between MacQueen's second method and the maximin method. It chooses the first center randomly and the i-th ($i \in \{2, 3, \ldots, K\}$) center is chosen to be $x' \in X$ with a probability of $j=1\ md(xj)2$, where md(x) denotes the minimum-distance from a point x to the previously selected centers. This method yields an $\Theta(\log K)$ approximation to the MSSC problem. The greedy k-means++ method probabilistically selects log(K) centers in each round and then greedily selects the center that most reduces the SSE. This modification aims to avoid the unlikely event of choosing two centers that are close to each other.

- The **PCA-Part method** uses a divisive hierarchical approach based on PCA (Principal Component Analysis) that start from an initial cluster that contains the entire data set, then the method iteratively selects the cluster with the greatest SSE and divides it into two subclusters using a hyperplane that passes through the cluster centroid and is orthogonal to the principal eigenvector of the cluster covariance matrix. This procedure is repeated until K clusters are obtained. The centers are then given by the centroids of these clusters [19][20][22].

- The **Var-Part method** is an approximation to PCA-Part, where the covariance matrix of the cluster to be split is assumed to be diagonal. In this case, the splitting hyperplane is orthogonal to the coordinate axis with the greatest variance. Lu et al.'s method uses a two-phase pyramidal approach. The attributes of each point are first encoded as integers using 2Q-level quantization, where Q is a resolution parameter. These integer points are considered to be at level 0 of the pyramid. In the bottom-up phase, starting from level 0, neighboring data points at level k ($k \in \{0, 1, \ldots\}$) are averaged to obtain weighted points at level k + 1 until at least 20K points are obtained. Data points at the highest level are refined using k-means initialized with the K points with the largest weights. In the top-down phase, starting from the highest level, centers at level k + 1 are projected onto level k and then used to initialize the k-th level clustering. The top-down phase terminates when level 0 is reached. The centers at this level are then inverse quantized to obtain the final centers. The performance of this method degrades with increasing dimensionality[20][21][22].

- **Onoda et al.'s method** first calculates K Independent Components (ICs) of X and then chooses the i-th ($i \in \{1, 2, \ldots, K\}$) center as the point that has the least cosine distance from the i-th IC [19].

IV. Recommendations for Practitioners

- The Silhouette is a metric used for validation while clustering. Thus, it can be used in combination with the another Method.
- Silhouette Method are not alternatives to other method for finding the optimal number of clusters. Rather they are tools to be used together for a more confident decision.
- In general, methods Forgy's method, MacQueen's second method, maximin, should not be used. These methods are easy to understand and implement, but they are often ineffective and unreliable. Furthermore, despite their low overhead, these methods do not offer significant time savings since they often result in slower k-means convergence.
- In time-critical applications that involve large data sets or applications that demand determinism, methods Var-Part, or PCA-Part should be used. These methods need to be executed only once and they lead to very fast k-means converge. The efficiency difference between the two is noticeable only on high dimensional data sets. This is because method Var-Part calculates the direction of split by determining the coordinate axis with the greatest variance (in O(D) time), whereas method P achieves this by calculating the principal eigenvector of the covariance matrix (in O(D2) time using the power method. Note that despite its higher computational complexity, method PCA-Part can, in some cases, be more efficient than method Var-Part. This is because the former converges significantly faster than the latter. The main disadvantage of these methods is that they are more complicated to implement due to their hierarchical formulation.
- The deterministic methods have good Initial SSE performance because these methods are approximate (divisive hierarchical) clustering methods by themselves and thus they give reasonable results even without k-means refinement

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

- MacQueen proposed two different methods. The first one, which is the default option in the Quick Cluster procedure of IBM SPSS Statistics, takes the first K points in X as the centers. An obvious drawback of this method is its sensitivity to data ordering, The second method chooses the centers randomly from the data points. The rationale behind this method is that random selection is likely to pick points from dense regions, i.e., points that are good candidates to be centers. However, there is no mechanism to avoid choosing outliers or points that are too close to each other, thus Multiple runs of this method is the standard way of initializing k-means.
- we focus on deterministic methods for two main reasons. First, these methods are generally computationally more efficient as they need to be executed only once. In contrast, random methods are inherently unreliable in that the quality of their results is unpredictable and thus it is common practice to perform multiple runs of such methods and take the output of therun5that produces the best objective function value. Second, several studies demonstrated that despite the fact that they are executed only once, some deterministic methods are highly competitive with well-known and effective random methods such as Bradley and Fayyad's method and k-means++

V. CONCLUSION & FUTURE WORK

The initialization of centroid in k-mean Algorithm affect cluster formation. If we do not select clustering good initialization point, we can get poor local optimal solution. In this paper, we have presented comparison of different k-mean Centroid Initialization Algorithm.

With the increasing need to analyze large amounts of data to get useful insights, it is essential to develop complex parallel machine learning algorithms that can scale with data and number of parallel processes. So our future work is the use of parallel processing for accelerating a newly modified k-means algorithm since, for every k, the N executions of the k-means algorithm are independent and can be performed in parallel.

REFERENCES

[1] A. Sinha and P. K. Jana, "A novel K-means based clustering algorithm for big data," *2016 Int. Conf. Adv. Comput. Commun. Informatics, ICACCI 2016*, pp. 1875–1879, 2016.

[2] L. Rokach and O. Maimon, "Chapter 15 Minkowski : Distance Measures for Numeric Attributes."

[3] L. Jegatha Deborah, R. Baskaran, and A. Kannan, "A Survey on Internal Validity Measure for Cluster Validation," *Int. J. Comput. Sci. Eng. Surv.*, vol. 1, no. 2, pp. 85–102, 2010.

[4] L. Ma, L. Gu, B. Li, Y. Ma, and J. Wang, "An Improved K-means Algorithm based on Mapreduce and Grid," vol. 8, no. 1, pp. 189–200, 2015.

[5] P. Arora, D. Virmani, and H. Jindal, "Proceedings of International Conference on Communication and Networks," vol. 508, 2017.

[6] A. M. Baswade and P. S. Nalwade, "Selection of Initial Centroids for k-Means Algorithm," *Ijcsmc*, vol. 2, no. 7, pp. 161–164, 2013.

[7] K. Teknomo, "K-Means Clustering Tutorial," *Medicine (Baltimore).*, pp. 1–12, 2006.

[8] T. V. Sai Krishna, A. Yesu Babu, and R. Kiran Kumar, "Determination of Optimal Clusters for a Non-hierarchical Clustering Paradigm K-Means Algorithm," pp. 301–316, 2017.

[9] M. M.-T. Chiang and B. Mirkin, "Experiments for the Number of Clusters in K-Means," *Prog. Artif. Intell.*, pp. 395–405, 2007.

[10] M. M. Chiang and B. Mirkin, "Number of clusters in K-Means clustering : an experimental study," pp. 1–20.

[11] M. M. T. Chiang and B. Mirkin, "Intelligent choice of the number of clusters in k-means clustering: An experimental study with different cluster spreads," *J. Classif.*, vol. 27, no. 1, pp. 3–40, 2010.

[12] B. Mirkin, "Choosing the number of clusters," *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 1, no. 3, pp. 252–260, 2011.

[13] M. M. Chiang and B. Mirkin, "Determining the number of clusters in the Straight K-means : Experimental comparison of eight options," *Proc. 13th Port. Conf. Prog. Artif. Intell.*, pp. 395--405, 2007.

[14] T. V. S. Krishna, A. Y. Babu, and R. K. Kumar, "Determination of Optimal Clusters for a Non-hierarchical Clustering Paradigm K -Means Algorithm," 2018.

[15] and T. H. Tibshirani, R., G. Walther, "Estimating the number of clusters in a data set via the gap statistic," *J. R. Stat. Soc. Ser. B (Statistical Methodol.*, vol. 63, no. 2, pp. 411-423., 2001.

[16] T. M. Kodinariya and P. R. Makwana, "Review on determining number of Cluster in K-Means Clustering," *Int. J. Adv. Res. Comput. Sci. Manag. Stud.*, vol. 1, no. 6, pp. 2321–7782, 2013.

[17] K. Chowdhury, D. Chaudhuri, A. K. Pal, and A. Samal, "Seed selection algorithm through K-means on optimal number of clusters," *Multimed. Tools Appl.*, 2019.

[18] S. Zahra, M. A. Ghazanfar, A. Khalid, M. A. Azam, U. Naeem, and A. Prugel-Bennett, "Novel centroid selection approaches for KMeans-clustering based recommender systems," *Inf. Sci. (Ny).*, vol. 320, pp. 156–189, 2015.

[19] B. B. C. D. Chaudhuri, "A novel multiseed nonhierarchical data clustering technique-ITOSMAC-27-5-1997- p 871-876.pdf." p. 6.

[20] J.F.Lu .B.TangZ. M.Tang. Y.Yang, "Hierarchical initialization approach for K-Means clustering," *Pattern Recognit. Lett.*, vol. 29, no. 6, pp. 787–795.

[21] F. Cao, J. Liang, and G. Jiang, "An initialization method for the K-Means algorithm using neighborhood model," *Comput. Math. with Appl.*, vol. 58, no. 3, pp. 474–483, 2009.

[22] D. Reddy and P. K. Jana, "Initialization for K-means Clustering using Voronoi Diagram," *Procedia Technol.*, vol. 4, pp. 395–400, 2012.

[23] P. A. V. Celebi, M. Emre, Hassan A. Kingravi, "A comparative study of efficient initialization methods for the k-means clustering algorithm. Expert," *Expert Syst. Appl.*, vol. 2, no. 1, pp. 1–13, 2012.

[24] A. Al Malki, M. M. Rizk, M. A. El-Shorbagy, and A. A. Mousa, "Hybrid Genetic Algorithm with K-Means for Clustering Problems," *Open J. Optim.*, vol. 05, no. 02, pp. 71–83, 2016.

[25] P. Purohit, "A New Efficient Approach towards k-means Clustering Algorithm," *Int. J. Comput. Appl.*, vol. 65, no. 11, p. 8887, 2013.

[26] T. Jinlan, Z. H. U. Lin, and L. I. U. L. M, "Improvement and Parallelism of k- Means Clustering Algorithm," *Tsinghua Sci. Technol. June*, vol. 10, no. 3, pp. 277–281.

[27] A. M. Fahim, A. M. Salem, F. A. Torkey, and M. A. Ramadan, "An efficient enhanced k-means clustering algorithm," *J. Zhejiang Univ. A*, vol. 7, no. 10, pp. 1626–1633, 2006.

[28] K. A. A. Nazeer and M. P. Sebastian, "Improving the Accuracy and Efficiency of the k-means Clustering Algorithm," vol. I, no. May, pp. 1–5, 2009.

[29] B. B. Bhusare and S. M. Bansode, "Centroids Initialization for K-Means Clustering using Improved Pillar Algorithm," vol. 3, no. 4, pp. 1317–1322, 2014.

[30] L. Bai, J. Liang, C. Dang, and F. Cao, "A cluster centers initialization method for clustering categorical data," *Expert Syst. Appl.*, vol. 39, no. 9, pp. 8022–8029, 2012.

[31] S. K. Pal and P. K. Pramanik, "Fuzzy measures in determining seed points in clustering," *Pattern Recognition Letters*, vol. 4, no. 3. pp. 159–164, 1986.

[32] S. Alam, G. Dobbie, Y. S. Koh, P. Riddle, and S. Ur Rehman, "Research on particle swarm optimization based clustering: A systematic review of literature and techniques," *Swarm Evol. Comput.*, vol. 17, pp. 1–13, 2014.

[33] R. Jothi, S. K. Mohanty, and A. Ojha, "DK-means: a deterministic K-means clustering algorithm for gene expression analysis," *Pattern Anal. Appl.*, vol. 22, no. 2, pp. 649–667, 2019.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500