International Journal of Management and Data Analytics (IJMADA)

Int. J. Management and Data Analytics, Vol. 1(2), 38-46
ISSN: (online)
Journal Homepage: http://ijmada.com

Big Data Analytics: A Tutorial of Some Clustering Techniques

Said Baadel

Canadian University Dubai, United Arab Emirates.

Received: July 20, 2021, Revised August 27, 2021, Published: September 17, 2021

ARTICLE INFO
Keywords:
Clustering; Data Mining; K-means; K-Medoid; MCOKE; Overlapping Clustering

ABSTRACT

Data Clustering or unsupervised classification is one of the main research areas in Data Mining. Partitioning Clustering involves the partitioning of n objects into k clusters. Many clustering algorithms use hard (crisp) partitioning techniques where each object is assigned to one cluster. The most widely used in hard partitioning algorithm is the K-means and its variations and extensions such as the K-Medoid. Other algorithms use overlapping techniques where an object may belong to one or more clusters. Partitioning algorithms that overlap include the commonly used Fuzzy K-means and its variations. Other more recent algorithms reviewed in this paper are the Overlapping K-Means (OKM), Weighted OKM (WOKM) the Overlapping Partitioning Cluster (OPC) and the Multi-Cluster Overlapping K-means Extension (MCOKE). This tutorial focuses on the above-mentioned partitioning algorithms. We hope this paper can be beneficial to students, educational institutions, and any other curious mind trying to learn and understand the k-means clustering algorithm.

1. INTRODUCTION

Data Mining techniques are used in order to extract meaningful information from Big Data and discover hidden patterns that can be used to predict certain trends. Some of these techniques in Data Mining include Association Analysis (rules that can predict relations between object variables in a large dataset), Classification (classifying an object to belong to one or more predefined classes), Clustering (grouping objects in clusters that share similar characteristics), and Regression (determining correlations between object attributes).

These techniques can be broadly divided into two basic concepts: supervised learning and unsupervised learning. In supervised learning, we have a set of labeled training data that given an input object returns an output that can be of continuous value (Regression) or could predict a class label of the input object associated with it (Classification). Supervised learning algorithms would thus learn from the training data where the outcome is known and used to predict the value of the output for any given valid input object after having seen some training examples. Unsupervised learning such as Clustering is the opposite of the former. There is no prior knowledge of the data labels and therefore no right or wrong answer. Hidden patterns have to be extracted from the data by looking at similarities between them and grouping them together into clusters.

These techniques and algorithms can be used for a variety of applications such as clustering data, information retrieval, pattern recognition, classifications, and associations. Clustering algorithms can be further subdivided into two, Hierarchical Clustering and Partitioning Clustering. According to (Abbas, 2008), hierarchical algorithms are more suitable for small data sets and partitioning algorithms for large data sets. The memberships in clustering can be looked at in two angles; hard or uncertain memberships. Hard memberships may have hard or overlapping partitions based on binary functions model whereas uncertain memberships are based on uncertainty functions using possibilistic or fuzzy logic framework. The section below briefly discusses Hierarchical clustering before exploring Partitioning clustering and some of the extensions around the K-means algorithm.

2. HIERARCHICAL CLUSTERING

This technique constructs a tree of clusters in a hierarchy also known as dendrogram. There are two ways to achieving this i.e., divisive and agglomerative methods. Divisive methods start with one huge macro-cluster containing all data objects and continuously split it into

Contact: Said Baadel, baadel@cud.ac.ae

two smaller groups generating a top-down hierarchy. On the contrary, agglomerative methods start with clusters with one data object (singleton clusters) at the bottom and continuously merges two clusters at a time to generate a bottom-up hierarchy.

A) *Divisive Clustering*

The framework of the divisive clustering is given in the pseudo-Algorithm 1 below.

Algorithm 1: Divisive Clustering

```
Start with the root cluster consisting of all data objects
repeat
    Select a parent cluster in a set of current clusters to split
    Split the parent node into two parts
until singleton leaves are achieved
```

There are two major considerations needed in using the above algorithm that could affect its performance. Firstly, the splitting method and criteria. As seen above, the technique for splitting the node into two parts is known as bisecting. The most widely used algorithm is the Bisecting K-means (Steinbach et. al., 2000) which uses K-means (MacQueen, 1967) on the parent cluster C to determine the best split which maximizes the Ward's distance (Ward, 1963) between two possible child clusters C1 and C2. The larger of the split cluster is selected as the new parent for further splitting and the method is iterated until K clusters have been obtained.

Secondly, deciding the appropriate cluster to split i.e. whether the algorithm should choose the cluster with the largest number of objects or select all clusters at each level. A more compromise alternative is selecting the cluster with the largest square error variance.

Whether referred to as a business continuity plan, or a corporate security plan, the general responsibilities of crisis management fall on the Human Resource department to ensure that the organization's employees are working in a safe environment and have adequate training in how to respond should a crisis arise. In the event that a disaster should occur, it will also fall on the HR department to locate employees, ensure they are safe, and arrange for counselling and recovery programs. There may be situations in which a business cannot access their headquarters, and must operate from a new location, or remotely. In this case the HR department may be responsible for facilitating the transfer.

Due to the unpredictable nature of crises, both man-made and natural, it is imperative for organizations to have an effective strategy that focuses on employee safety, as well as business continuity. In order to analyze how different companies manage crisis situations, interviews were conducted with three large multi-national corporations; American Express Corp., Husky Injection Moulding Systems Ltd., and Johnson & Johnson. The interviews consisted of a multi-part survey aimed at examining the three key functions of crisis management; Planning, Reaction, and Learning (Exhibit 1). It is understood that this topic is emotionally charged, and can relate to an organization's security, thus no information regarding organizational security protocols or proprietary information was requested.

B) *Agglomerative Clustering*

The framework for the agglomerative clustering is given in the pseudo-Algorithm 2 below.

Algorithm 2: Agglomerative Clustering

```
Start with clusters with one data object (singleton cluster)
repeat
    compute the dissimilarity between clusters
    Merge two least dissimilar clusters into a new cluster
until one maximal cluster is achieved that contains all data objects in a single cluster
```

The algorithm computes the dissimilarity of all data sets between clusters. The commonly used formula is the Lance-Williams dissimilarity update formula (Lance & Williams, 1967) to compute distance between the clusters by either considering single linkage (nearest neighbor – similarity is that between most similar member), average linkage (group average – considers average pair-wise similarity), and complete linkage (maximal – choosing a cluster pair whose merge has the smallest diameter) (Sneath & Sokal, 1962). The closest pairs that are less dissimilar are merged bottom-up until one maximal cluster remains.

There are a few advantages and disadvantages of hierarchical clustering. Among the common advantages cited include:

i) Easy to understand representation of the hierarchical tree or dendrogram (Berkhin, 2006).

ii) Different similarity/dissimilarity distance functions can easily be used (Xu et al, 2005).

Disadvantages include:

i) Difficult to reassign an object if a misclassification was done previously (Berkhin, 2006).

ii) Very sensitive to outliers (Xu et al, 2005).

Hierarchical clustering normally produces dendrograms which provide data views at different levels making it easy for the end user to visualize the problem during the clustering process. However, one major disadvantage of hierarchical algorithms is that in any level of the hierarchy, once the merge/fusion or split decisions are made they cannot be undone (Fisher, 1995). In order to overcome such a limitation, Fisher proposed an algorithm that would modify the dendrogram iteratively until the optimal solution is found. Algorithms that use this approach produce quadratic high computational complexity (Aggarwal & Reddy, 2014) making them unrealistic for large real life problems. Most hierarchical algorithms tend to have quadratic or higher complexity in the number of data points (Chakraborty & Nagwani, 2011) and thus not very suitable for large data sets.

3. Partitioning clustering

This technique splits the set of data into partitions based on K disjoint initial groupings or clusters and using an objective function iteratively improves the quality of those partitions. K is provided by the user. We first examine the hard (crisp) clustering techniques followed by soft (overlapping) techniques. The most widely used and simplest algorithm for this hard clustering is K-means (MacQueen, 1967; Jaini, 2010; Lloyd, 1982). Some variations of K-means include K-Medoids, K-Mode, Fuzzy K-Means. Others include CLARANS (Raymond & Han, 2002) and PAM (Partitioning Around Medoids) (Kaufman & Rousseeuw, 1990) which are modifications of K-Medoids.

A) K-Means Clustering

Undoubtedly, the K-means is the most widely used partitional clustering algorithm (Jain, 2010). There are many reasons attributed to this such as; a) it is very easy to implement, b) very versatile in that any part of the algorithm can be easily modified, c) it is guaranteed to converge (Selim and Ismail, 1984) at a quadratic rate (Bottou and Bengio, 1995).

This research primarily focuses on partitioning clustering algorithms and methods that extend K-means based on the squared Euclidean distance that will measure linear separation between the clusters (for simplicity) and will not explore those K-means variations with nonlinear separation that project data onto high dimensional kernel space such as Kernel K-means (Scholkopf et al., 1998) and Kernel Overlapping K-means (BenN'Cir et al., 2010).

The framework for the K-Means clustering is given in the pseudo-Algorithm 3 below.

```
Algorithm 3: K-means Clustering

1: Select K points as initial cluster centers (centroids) randomly or based on heuristics
2: **repeat**
3:         Assign each data points to its closest centroid
4:     Recalculate the centroid of each cluster and check for convergence of objective function
5: **until** convergence criteria is met and no change on each cluster or maximum number of
           iterations are achieved.
```

The algorithm begins with first assigning K points as the initial cluster centers. This can be done randomly or by based on heuristics. Data objects are then assigned to their nearest cluster center by calculating the distance using Euclidean distance measurement. The sum of squares errors (objective function) is then calculated by squaring the Euclidean distances to each cluster centroid and the object is assigned to the cluster with the smallest value. The recalculation of the centroids is taken as the average of the values of the objects that are part of that cluster. These steps are then iterated in a loop until the objects in each cluster do not change or until a maximum number of iterations are reached. The steps 3 and 4 above is aimed at minimizing the objective function defined in the Equation (1) below for the given set of centroids.

$$C = \sum_{k=1}^{K} \sum_{xi \in Ck} ||xi - ck||^2 \tag{1}$$

Where C_k is the kth cluster, x_i is a point in C_k, and c_k is the mean of the kth cluster.

K-means clustering being a greedy-descent nature algorithm, it will converge to a local minimum (Selim & Ismail, 1984). For up to two-dimensional Euclidean space with arbitrary number of k clusters, the K-means complexity is NP-Hard (Manning et al, 2008; Everitt et al, 2001).

The K-means algorithm does require two things a priori from the users. One is to first choose the initial centroids, and two, to estimate the number of K clusters in advance. Choosing the initial clusters largely affects the outcome of the algorithm. This can be done randomly by picking K points as K centers as suggested in (MacQueen, 1967). Forgy (Forgy, 1965) suggests spreading the randomness of the initial location of the clusters. The idea being that this random selection is likely to pick points from the dense regions which may be good centers. However, this does not eliminate the possibility of picking an outlier for a center. This can be minimized by having multiple runs of this method. Other heuristic approaches that have been proposed for cluster initialization includes Ward's distance method (Ward, 1963) that uses the sum of squared errors to evaluate between two cluster distances as suggested in (Milligan, 1981). Kaufman's method (Kaufman, 1990) selects the K centers sequentially by first choosing the most centrally located data object in the data set and subsequently choosing the other centers that have many data objects around it by a heuristic function. K-means++ (Arthur & Vassilvitskii, 2007) selects the first centroid randomly and subsequently chooses the next centroid which is farthest from the currently selected centroid based on a weighted probability score. (Bradley & Fayyad, 1998) method randomly partitions the set into J subsets which are then combined into a superset clustered in by k-means J times initialized each time with a different center. The center set that gives the least SSE are considered the final centers.

As for estimating the number of K clusters in advance, the ISODATA algorithm (Ball, 1965) was one attempt in determining the optimal K where K-means is first ran on the dataset to obtain the clusters which these clusters are then merged if the distance between them is less than a certain given threshold or split if the standard deviation within the cluster is exceeds the same threshold. The Silhouette Coefficient algorithm (Kaufman, 1990) takes into account the inter and intra cluster distances for any given data object. The average *a1* of the distances is calculated for all points intra cluster for a given point x. The average *a2* is then calculated for all other inter clusters that don't contain point x. These two values *a1* and *a2* are then used to estimate the Silhouette Coefficient of point x. The average of all the silhouettes becomes the width for all the dataset points. Other methods include the Gap statistic method (Tibshirani et. al., 2001) that estimate the number of clusters using gap statistic, Calinski-Harabasz index (Calinski & Harabasz, 1974) among others. This goes to show that estimating the number of clusters is non-trivial and is a major challenge for the K-means algorithm. K-means also has a shortcoming in that it only works for numerical data and not categorical data. The K-mode clustering algorithm (Xu & Wunsch, 2005), a variation of the K-means, works for categorical data sets. The K-mode algorithm differs from K-means in that it uses modes instead of means to calculate the centroids and measures dissimilarity between the categorical data instead of the Euclidean distances between the objects.

Another drawback for the K-means algorithm is that it is very sensitive to outliers. Since all data objects must be assigned to a cluster, an outlier can easily affect the mean of the data objects in that given centroid. K-Medoids algorithm addresses this problem by choosing the actual data points as the cluster prototypes. The K-Medoid algorithm is further improved by the Partitioning Around Medoids (PAM) algorithm (Kaufman & Rousseeuw, 1990) and the Clustering LARge Applications (CLARA) algorithm (Raymond & Han, 2002). Although beyond the scope of this research, it is worth mentioning a brief description of the K-Medoid algorithm in the section below.

B) K-Medoid

The framework for the K-Medoid Clustering is given in the pseudo-Algorithm 4 below.

Algorithm 4: K-Medoid Clustering

```
Select K points as initial cluster centers (centroids) randomly or based on heuristics
repeat
    Assign each data points to its closest centroid
    Select a random non-representative object x and compute cost of swapping x with a representative object y.
    If the cost < 0 then swap x with m to form the new set of K representative objects
until convergence criteria is achieved.
```

The algorithm uses the actual data objects as prototypes and randomly assigns an object x to replace an object y which is represented in the cluster prototypes. Once this is done, the membership of all data points that belonged to the representative y are checked and if they are closer to x then y is swapped with x. The cost of swapping is computed as the absolute error criterion for K-Medoids and is recalculated for every assignment of x and y as it obtains the final representative points for each cluster. This fact makes K-medoid computational complexity higher than that of K-means and thus not very suitable for big data sets.

4. Overlapping Clustering

Many clustering algorithms are hard clustering techniques where an object is assigned to a single cluster. Fuzzy clustering techniques allow objects to belong to multiple clusters with different degrees by assigning membership degrees to the objects and allowing the object to belong to the cluster that has the highest degree. Data points with very small membership degrees can in this case help us distinguish noise points. The sum of all membership degrees add up to unity.

There are several overlapping clustering algorithms which are graph-based clustering algorithms, which are out of scope of this research. Overlapping graph-based methods use greedy heuristics and may be applicable in community detection in complex networks (Fellows et al., 2011). However, it is worth mentioning that these algorithms have major limitations that do not make them practical for real-life problems as out lined by Perez-Suarez et. al (Perez-Suarez et. al, 2013) . Some of the mentioned limitations indicated are:

i) They produce a large number of clusters in that analyzing these clusters could be as difficult as analyzing the whole collection.
ii) There is a very high overlapping in the clusters which would essentially hinder getting useful information about the structure of the data.
iii) They have a very high computational complexity thus making them unrealistic to apply them to real-life problems.

The primarily focus of this research is on partitioning clustering algorithms and methods that extend K-means based on the squared Euclidean distance which are fast and have low computational complexity compared to hierarchical making them suitable for large data sets.

The algorithms discussed optimize partitions for fixed *k* clusters (i.e. k is defined a priori) and in most cases are suitable if there is some domain information regarding the *k*. Otherwise it is not trivial to determine what suitable *k* is for any given data set. Alternatively, different runs can be made with different values of *k* and the results be compared to see which produced the best partition and can be then be used as the benchmark for future runs. (Bandyopadhyay et. al, 2002; Hruschka et. al, 2004) proposes methods to optimize the number of *k* clusters without having to define it beforehand.
One of the commonly used soft-clustering techniques is the Fuzzy K-means commonly referred to as Fuzzy C-means (FCM) (Bezdek, 1981). The algorithm works similar to the K-means where the algorithm minimizes the objective function (sum of squares error) until the centroid converges. Other algorithms that are a variation of FCM to deal with non-numerical data sets include Fuzzy K-mode, and Fuzzy K-medoid. Some extensions of FCM include Possibilistic C-means (Krishnapuram & Keller, 1996).

The K-Medoid algorithm outlined earlier is more robust to outliers compared to the K-means. (Cleuzious, 2009) proposed an algorithm OKMED that extends the method PAM and Kernel Overlapping K-Means or KOKMϕ (BenN'Cir et al., 2010) both detecting overlapping clustering around medoids. Other works that we will discuss are the Overlapping K-means (OKM) (Cleuzious, 2008), The Weighted-OKM (WOKM) (Cleuzious, 2009), Overlapping Partitioning Cluster (OPC) (Chen & Hu, 2006), the Multi-Cluster Overlapping K-means Extension (MCOKE) algorithm (Baadel et. al, 2015; Baadel et. al, 2016), and Improved Multi-cluster Overlapping K-Means Extension Algorithm (IMCOKE) (Danganan, et al., 2021).

A) Fuzzy K-Means

The algorithm works similar to that of K-means and the solution will correspond to the local minimum of the objective function. The sum of squared errors (SSE) objective function is defined in the Equation (2) below:

$$C = \sum_{k=1}^{K} \sum_{xi \in Ck} w_{xik}^{\beta} \left|\left|xi - ck\right|\right|^2 \tag{2}$$

Where C_k is the *k*th cluster, x_i is a point in C_k, c_k is the mean of the *k*th cluster and *w* is the membership weight of point x_i belonging to cluster C_k. β controls the fuzziness of the memberships such that when it approaches one it acts like k-means algorithm assigning crisp memberships.

The algorithm minimizes this SSE iteratively and updates the membership weightage and clusters until convergence criteria are met or improvement over the previous iteration does not meet a certain threshold. By assigning the memberships a weightage degree between 0 and 1, the objects are able to belong to more than one cluster with a certain weight hence generating soft partitions or clusters. The overall weight however must add to unity i.e. 1. Objects are eventually assigned to clusters that have the highest degree of membership. If the highest degree of membership is not unique, then an object is assigned to an arbitrary cluster that achieves the maximum. By adding a constraint where the data object must belong to a cluster with the highest membership degree, a "1" is imposed on every object in the matrix thus degenerating it to crisp-partitioning.

B) Overlapping K-Means (OKM) and Weighted OKM (WOKM)

The framework of the Overlapping K-means is given in the pseudo-Algorithm 5 below.

```
Algorithm 5 : Overlapping K-Means

Input: a set of data vectors, max number of iterations, threshold on the objective
Output: final coverage of the points
1: Draw K points as initial cluster prototypes
2: **repeat**
3:     Assign the data points to its closest prototype
4:     Re-compute the cluster prototypes and re-compute the assignments based on the new coverage
5: **until** convergence criteria is met and no change on each cluster or maximum number of iterations are achieved or a threshold on the decreasing number of the objective function
```

The algorithm proposed by (Cleuzious et. al, 2008) initializes a random cluster prototype with random centroids as an image of the data. Optional threshold value can be entered by the user during the initialization step. The aim is to minimize the objective function given in the Equation (3) below.

$$J(\{\pi_c\}_{c=1}^{k}) = \sum_{i=1}^{N} ||x_i - \emptyset(x_i)||^2 \tag{3}$$

Where π_c represents the c^{th} cluster with $x_i \in \mathbb{R}$.

After calculating the SSE of the data objects to their centers using the Euclidean square distance, it assigns these objects to their nearest centroids. The algorithm then computes the SSE of the prototype and compares these objects with the prototype center assignments to determine the mean of the two vectors to become the threshold to assign the objects to multiple clusters. Once the initial assignment of objects to their centroids is done, the mean between each cluster (threshold) is used to determine if the object should belong to the next nearest cluster as well. OKM uses heuristic to determine the set of possible assignments by sorting the clusters from nearest to furthest and assigning the object to the nearest cluster. If the mean m_x of the clusters already associated with the object plus the mean m_y of the next nearest cluster is lower than the threshold (mean of all the clusters associated with the object), then these two clusters are associated and the object will belong to that cluster as well. This assignment procedure is iterated until the stopping criteria or the maximum number of iterations is met resulting into a new coverage of the data objects in multiple clusters.

The WOKM is an extension of the OKM and Weighted K-means (Huang et al., 2005) that introduces a weighting vector λ_c of a subset of attributes relative to a given cluster c that may be assigned to that cluster and a vector $\gamma_i$of weights relative to the representative $\phi(x_i)$ with the aim of minimizing the objective function given by Equation (4) below

$$J(\{\pi_c\}_{c=1}^{k}) = \sum_{x_i \in X} \sum_{v=1}^{P} \gamma_{i,v}^{\beta} | x_{i,v} - \phi_v(x_i)|^2 \tag{4}$$

The objective function is optimized by first assigning each data object to the nearest cluster while minimizing the error, and secondly by updating both the cluster representatives and the set of cluster weights. In this algorithm the distance feature is also weighted by the feature weights contrary to the standard K-means which ignores the weights of any particular feature and considers all of the features to be equally important.

C) Overlapping Partitioning Cluster (OPC)

The framework of the Overlapping Partitioning Cluster is given in the pseudo-Algorithm 6 below.

```
Algorithm 6 : Overlapping Partitioning Cluster

Input: k number of clusters, similarity level
Output: final coverage of the points
1: Create 2 tables, distance table (based on the data objects) and similarity table (based on the distance table and the minimum similarity threshold entered by user)
2: **repeat**
3:     Assign the data points to its closest centroid that satisfies the similarity threshold
4:     If the new objective value is greater than maximal one, maximal is new objective
5: **until** convergence criteria is met
```

(Chen & Hu, 2006) proposed the algorithm which accepts the k number of clusters and the s similarity level or threshold as inputs. It first does some preprocessing work which creates two separate tables; a distance table which has the distances between all object pairs and a similarity table that stores the similarity of the objects based on the threshold entered by the user. If the distance between the two objects is greater than the top 5% percentile of all object pairs then the similarity level is 0 otherwise a 1 is assigned that indicates it should be included. Random initial centroids are selected based on heuristics and objects are assigned to the nearest centroid. The objective function works in two folds, minimizes the intra-distance between the object and the centroid while maximizing the inter-distance between the centroids. The objects are assigned meeting the objective function and the cluster centroids are adjusted iteratively with the new objects until the objective function converges. The OPC is based on the K-medoid algorithm and inherits similar characteristics and performance as its parent algorithm.

D) Multi-Cluster Overlapping K-Means Extension (MCOKE)

The framework for the MCOKE algorithm is given in the pseudo-Algorithm 7 below.

```
Algorithm 7: MCOKE: Multi-Cluster Overlapping K-means Extension

Input: Number of clusters $K$, a set of data vectors
Output: Membership matrix
1:    Select K points as initial cluster centers (centroids)
2: **repeat**
3:        Assign each data points to its closest centroid using the Euclidean distance
4:        Re-compute and update the centroids
5: **until** convergence criteria is met and no change on each cluster or maximum number of iterations are achieved
6.    Return assignment vector, final centroid vector, and the maximum Euclidean distance (maxdist) allowed in the initial assignments
7.    Draw an initial membership matrix table
8.    Compare each data point to the final centroid vector distance with maxdist and update membership table if distance is shorter
```

The MCOKE algorithm introduced by (Baadel et. al, 2015) consists of two procedures. The first part is the standard K-means clustering that iterates through the data objects in order to attain a distinct partitioning of the data points given a priori number of k clusters by minimizing the distance between the objects and the cluster centroids. The second part creates a membership table that compares the matrix generated after the initial K-means run to *maxdist* (the maximum distance of an object to a centroid that an object was allowed to belong to any cluster). This maxdist is used as the threshold to allow objects to belong to multiple clusters. Overlapping objects are not assigned degree of memberships but rather a 1 if it belongs and a 0 otherwise.

The first part of the algorithm is that of K-means and the solution will correspond to the local minimum of the objective function. The sum of squared errors (SSE) objective function is defined in the Equation (3.1) above for the K-means. K-means clustering being a greedy-descent nature algorithm, it will converge to a local minimum. After an initial run of the first step (that includes steps 1 to 5 above), the algorithm will return 3 things. Firstly, a vector of all the data objects with their assignment to each cluster. Secondly, a vector containing the final list of the cluster centroids. This vector of all centroids will be used in the second part of the algorithm to determine if the objects should belong to them. Thirdly, the *maxdist* as determined by the Euclidean distance of the objects to the centroids is made the global threshold to compare similarity of the objects to other clusters.

The second part of the algorithm draws an initial membership matrix table with hard clustering result of the data objects and softens these partitioning by iterating through the membership matrix and comparing the objects to the final centroids vector using the threshold *maxdist* and reassigning them to the clusters if the distance of the object to those centroids is less than *maxdist*.

E) Enhanced Multi-cluster Overlapping K-Means Extension (ehMCOKE)

eHMCOKE is an improvement of Improved MCOKE (Danganan et al., 2018) that was built to identify overlapping objects while avoiding some of the shortcomings identified in MCOKE. The algorithm, proposed by Danganan et al. (2021) incorporates median absolute deviation (MAD) to detect outliers in the dataset. To improve the performance of eHMCOKE, the authors deployed a three-step process in the algorithm.

Step 1 used MAD to identify and prune any outliers in the dataset. The second step utilized MCOKE algorithm to cluster the data. Finally, the last step uses the *maxdist* technique identified in MCOKE with additional parameters that were deployed in step 1 to assign the datapoints into overlapping clusters.

5. CONCLUSION

In this paper we mentioned the different forms of clustering algorithms. Most of the algorithms discussed are hard clustering where an object belongs to one cluster i.e. single membership. The Fuzzy K-means and its variations are the most commonly used soft-clustering algorithms where an object belongs to one or more clusters i.e. multiple memberships. Object memberships in Fuzzy techniques are based on variation of degrees on their belonging to each cluster and must add to unity. The

OKM, WOKM, and OPC algorithms break away from the fuzzy concept but require a threshold be set for the similarity function in determining the belonging of objects. This may not be easily done by novice users. MCOKE and eHMCOKE algorithms differs from other overlapping algorithms in that they do not require a similarity threshold to be defined a priori which may be difficult to set depending on the data samples. It instead uses the maximum distance (*maxdist*) allowed in K-means based on the SSE on Euclidean distance to assign objects to a given cluster as the global threshold. However, while the *maxdist* can be significantly affected in the presence of outliers rendering it not very effective (as in MCOKE), eHMCOKE algorithm prunes the outliers prior to applying clustering.

Finally, the algorithms require users to enter the number of *k* clusters and assign the objects based on the defined number of *k* clusters. There is a need for new algorithms to be able to assign and add new clusters on the fly on top of the *k* depending on the data set, i.e., data sets that are updated frequently or contain outliers deemed as noise without pruning them.

REFERENCES

[1] Abbas, O.A. (2008). Comparisons between Data Clustering Algorithms. The International Arab Journal of Information Technology, Vol 5. No. 3.

[2] Aggarwal, C.C, Reddy, C.K. (2014). Data Clustering: Algorithms and Applications. CRC Press, pages 15-19.

[3] Arthur, D. and Vassilvitskii, S. (2007). K-means++: The advantages of careful seeding. In proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pages 1027-1035.

[4] Baadel, S., Thabtah, F., and Lu, J. (2015). Multi-Cluster Overlapping K-means Extension Algorithm. In proceedings of the XIII International Conference on Machine Learning and Computing, ICMLC'2015.

[5] Baadel, S., Thabtah, F., and Lu, J. (2016). Overlapping Clustering: A Review. SAI Computing Conference, London, UK. IEEE. DOI: 10.1109/SAI.2016.7555988

[6] Ball, G.H. (1965). Isodata, a novel method of data analysis and pattern classification. Technical report, DTIC Document.

[7] Bandyopadhyay, S., Maulik, U. (2002). "Genetic Clustering for Automatic Evolution of Clusters and Application to Image Classification", Pattern Recognition, Vol. 35, pp. 1197-1208.

[8] BenN'Cir, C., Essoussi, N., Bertrand, P. (2010). Kernel Overlapping K-Means for Clustering in Feature Space. In International Conference on Knowledge Discovery and Information Retrieval (KDIR), pages 250-256

[9] Berkhin, P. (2006). A survey of Clustering Data Mining Techniques. In Grouping Multidimentional Data, Springer, Berlin Heidelberg, pages 25-71.

[10] Bezdek, J.C. (1981). Pattern Recognition with Fuzzy Objective Function Algorithms. Kluwer Academic Publishers.

[11] Bradley, P.S., Fayyad, U. (1998). Refining Initial Points for K-means Clustering. 15th International Conference on Machine Learning, 1998, pp. 91-99

[12] Bottou, L, Bengio, Y., (1995). Convergence Properties of the K-Means Algorithms. Advances in Neural Information Processing Systems 7, MIT Press, pp. 585–592.

[13] Calinski, T. and Harabasz, J., (1974). A dendrite method for cluster analysis. Communications in Statistics, vol 3 pp. 1–27.

[14] Chakraborty, S., Nagwani, N. K. (2011). Analysis and Study of Incremental K-Means Clustering Algorithm. High Performance Architecture and Grid Computing. Springer Berlin Heidelberg, pages 338-341

[15] Chen, Y. Hu, H. (2006). An overlapping Cluster algorithm to provide non-exhaustive clustering. European Journal of Operational Research. Elsevier, pages 762-780.

[16] Cleuzious, G. (2008). An extended version of the k-means method for overlapping clustering. IEEE International Conference on Pattern Recognition.

[17] Cleuzious, G. (2009). Two Variants of the OKM for Overlapping Clustering. Advances in Knowledge Discovery and Management, pages 149-166.

[18] Danganan, A. E., Sison, A. M. and Medina, R. P. (2018). An Improved Overlapping Clustering Algorithm to Detect Outlier. Indones. J. Electr. Eng. Informatics Vol. 6 (4) pp. 401-409. DOI: 10.11591/ijeei.v6i4.499.

[19] Danganan, A. E., De Los Rayes, E. (2021). eHMCOKE: an Enhanced Overlapping Clustering Algorithm for Data Analysis. Bulletin of Electrical Engineering and Informatics. Vol. 10 (4). Pp. 2212-2222. DOI: DOI: 10.11591/eei.v10i4.2547

[20] Everitt, B.S., Landau, S., Leese, M. (1981). Cluster Analysis, Arnold Publishers.

[21] Fellows, M. R., Guo, J., Komusiewicz, C., Niedermeier, R., and Uhlmann, J. (2011). Graph-based data clustering with overlaps. Discrete Optimization, 8(1):2–17.

[22] Fisher, D. (1995). Optimization and simplification of hierarchical clustering. In Proceedings of the 1st International Conference on Knowledge Discovery and Data Mining (KDD), pages 118-123.

[23] Forgy, E.W. (1965). Cluster analysis of multivariate data: efficiency versus interpretability of classifications. Biometrics, 21: pages 768-769.

[24] Hruschka, E.R., de Castro, L.N., Campello, R.J.G.B. (2004). Evolutionary Algorithms for Clustering Gene-Expression Data, In Proc. 4th IEEE Int. Conference on Data Mining, pp. 403-406.

[25] Huang, J. Z., Ng, M. K., Rong, H., Li, Z. (2005). Automated Variable Weighting in K-means type Clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(5), pages 657-688.

[26] Jaini, A. (2010). Data Clustering: 50 years beyond k-means. Pattern Recognition Letters, 31(8): pages 651-666.

[27] Kaufman, L. and Rousseeuw, P. (1990). Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley & Sons.

[28] Krishnapuram, R. and Keller, J.M. (1996). The possibilistic C-means algorithm: Insights and recommendations. IEEE Transactions on Fuzzy Systems, 4(3): pages 385-393.

[29] Lance, G.N. and Williams, W.T. (1967). A general theory of classificatory sorting strategies II. Clustering Systems. The computer Journal, pages 271-277.

[30] Lloyd, S. (1982). Least squares quantization in pcm. IEEE Transactions on Information Theory, 28(2): pages 129-137.

[31] MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, volume 1, pages 281-297, Berkeley, USA.

[32] Manning, C.D., Raghavan, P. and Schutze, H. (2008). Introduction to Information Retrieval, volume 1. Cambridge University Press, Cambridge.

[33] Pal, K., Keller, J.M, and Bezdek, J.C. (2005). A possibilistic fuzzy C-means Clustering Algorithm. IEEE transactions of Fuzzy Systems, 13(4): pages 517-530.

[34] Pérez-Suárez, A. et. al. (2013) OClustR: A new graph-based algorithm for overlapping clustering. Journal on Advances in Artificial Neural Networks and Machine Learning, vol. 121: pages 234-247.

[35] Raymond, T., and Han, Jiawei. (2002). CLARANS: A method for clustering objects for spatial data mining. IEEE Transactions on Knowledge and Data Engineering, 14(5): pages 1003-1016.

[36] Selim, S. Z. and Ismail, M.A. (1984). K-means-type algorithms: A generalized convergence theorem and characterization of local optimality. IEEE Transactions on Pattern Analysis and Machine Intelligence, 6(1): pages 81-87.

[37] Scholkopf, B., Smola, A., Muller, K. R. (1998). Nonlinear Component Analysis as a Kernel Eigen Value Problem. Neural Computation, 10(5), pages 1299-1319.

[38] Sneath, P.H. and Sokal, R. (1962). Numerical Taxonomy. Nature, pages 855-860.

[39] Steinbach, M., Karypis, G., and Kumar, V.(2000). A Comparison of Document Clustering Techniques. In KDD Workshop on Text Mining, volume 400, pages 515-526. Boston, USA.

[40] Tibshirani, R., Walther, G., and Hastie, T. (2001). Estimating the number of data clusters via the gap statistic. Journal of the Royal Statistical Society B, volume 63, pages 411–423

[41] Ward, J. (1963). Hierarchical grouping to optimize an objective function. Journal of the American Statistical Association, 58(301), pages 236-244.

[42] Xu, R. and Wunsch, D. (2005). Survey of clustering algorithms. IEEE Transactions on Neural Networks, 16(3): pages 645-678