K-NEAREST CLUSTERING ALGORITHM

Ahmed Fahim[1,2]

[1] Department of Computer Science, Faculty of Sciences and Humanitarian Study,
Prince Sattam Bin Abdulaziz University, Al-Aflaj, Saudi Arabia
a.abualeala@psau.edu.sa
[2] Department of Computer Science, Faculty of computers and information,
Suez University, Suez, Egypt
ahmmedfahim@yahoo.com

Abstract—*data clustering is very important method in data mining and knowledge discovery, it discovers latent grouping of data. It is useful tool for discovering knowledge from massive dataset. This knowledge is important for decision maker. Data clustering divides set of objects into finite number of subsets called clusters; a cluster contains similar objects in some way. This research introduces a clustering algorithm based on a new definition for cluster. A cluster is a connected graph with weighted nodes satisfies some constraints. The proposed method is able to discover clusters with varied shapes, sizes and densities. The experimental results show the efficiency of the proposed method that requires o(n log n); where n is size of input dataset.*

Keywords - data mining; clustering algorithms; k-nearest neighbors; data clustering

I. INTRODUCTION

Data clustering is very important in discovering intrinsic structure of data. Data clustering aims to group large set of objects into finite number of sub sets such that each sub set collects similar objects together. Each sub set is called cluster. Clustering process is unsupervised learning, so any clustering result is depending on the definition of cluster. In partitioning methods like k-means [1], PAM (Partitioning Around Medoids)[2], CLARA (Clustering LARge Applications)[2] and CLARANS (Clustering Large Applications based on RANdomized Search)[3] cluster is a collection of objects with minimum variance from cluster center. In other words, cluster is a collection of objects with minimum mean square error from their cluster center. This definition prefers convex shaped clusters. If data contains clusters of different shapes, then this definition is not suitable and the clustering method will not discover the correct clusters. It may split or merge some clusters. Another definition is that cluster is a connected graph where each object is connected to the closest neighbor. This definition is the base for a hierarchical method called single link [4], the other methods in this category are based on this definition in somewhat different CURE (Clustering Using REpresentatives) [5] and BIRCH (Balanced iterative reducing and clustering using hierarchies) [6]. The third definition is based on density notion. Where cluster is a set of density connected objects; dense object is known as core object which have not less than specified number of objects within it neighborhood of specified radius. This definition is introduced in DBSCAN (Density-based spatial clustering of applications with noise) [7] method which is the pioneer in density based method to discover clusters of varied shapes and sizes.

Here we introduce another definition for a cluster that steamed from the previous definitions. A cluster is a connected graph where each node is assigned a weight; this weight is equal to the sum of distances to its most nearest neighbors, at most four neighbors. Similar nodes are connected together such that each node is similar to a specified node with threshold. This definition is a middle ground between the previous definitions. We have used many benchmark datasets to test the proposed method. The results demonstrate the efficiency of the proposed algorithm in discovering clusters of different shapes, sizes and densities.

This paper is organized as follow; section II presents some related work and the problem of clustering algorithms. Recent advances and improvements of algorithms is presented in section III, section IV demonstrates the proposed method. Section V shows the effectiveness of the proposed method and discusses the results we obtained from it. Finally, section VI concludes the paper.

II. RELATED WORK

There are large numbers of clustering algorithms have been proposed, these algorithms may be classified into the following categories: 1- partitioning methods, 2. Hierarchical methods, 3. Density based methods, 4. Grid based methods. In partitioning methods each cluster is represented by a single point called cluster center as in k-means [1] or median as in PAM [2], CLARA [2], and CLARANS [3]. Since each cluster is represented by a single point these methods are suitable only for convex shaped clusters. They fail to discover intertwined clusters and varied shaped clusters. Also, these methods require number of clusters to be known in advance. For example, k-means works as follow:

```
Input: dataset and k (required number of clusters)
Output: k cluster centers
    1- Randomly or heuristically select k initial centers
    2- Assign each object to the closest cluster (center)
    3- Compute new centers as the mean of objects in each cluster
    4- Repeat steps 2 and 3 until objects cease to change their clusters.
```

The final result of k-means depends on the initial selected centers, and the correct value of k that require some prior knowledge about the input dataset.

Hierarchical methods build a dendrogram; like a tree structure. These methods are classified into two categories; agglomerative and divisive. Agglomerative methods build the dendrogram from bottom up, and divisive methods build it from top down. For example single link [4] works as follow:

```
Input: dataset and level of dissimilarity
Output: set of clusters
1- Initially, each object is a singleton cluster
2- Merge the closest (most similar) pair of clusters
3- Update distance (similarity) matrix
4- Repeat steps 2 and 3 until level of dissimilarity reached or all objects are in single cluster.
```

This method discovers clusters of varied shapes but have a problem with a chain of objects connecting two clusters that leads to merge them in single cluster, in addition to the large time complexity o(n^2), where n is the size of the input dataset. Other methods may take a sample from data like CURE [5] algorithm, or summarize the data as in BIRCH [6] algorithm.

Density based methods consider a clustering problem as separating dense regions from sparser ones; a cluster is a dense region isolated form other clusters by low dense regions. DBSCAN [7] method is the first method introduced this idea, it depends on counting the objects in fixed neighborhood distance from the current object, if neighborhood has objects larger than or equals to threshold then this object is dense (core) object and the cluster can be expanded from this objects otherwise the object is considered noise temporarily. So objects in dataset may be core, border or noise. Border object is not core but lies in neighborhood of core object. DBSCAN works as follow:

```
Input: dataset, neighborhood radius (Eps), and minimum object in neighborhood (MinPts)
Output: set of clusters
    1- All objects are unclassified, clus_id=0
    2- For i= 1 to dataset size
    3- If object(i) is unclassified then goto 4 else goto 5
    4- Expand cluster(dataset, object (i), Eps, MinPts)
            a. neighbors= regionquery (object( i), Eps)
            b. If |neighbors| ≥ MinPts then goto c else goto m
            c. Increase clus_id by 1
            d. Assign object(i) and its unclassified or noise neighbors to current clus_id
            e. While |neighbors| < > 0
            f. current_object = neighbors.first
            g. object_neighbors = regionquery(current_object, Eps)
            h. if |object_neighbors| ≥ MinPts then goto i else goto k
            i. append all unclassified object_neighbors to neighbors
            j. assign current_object and all unclassified or noise object_neighbors to current clus_id
            k. delete current_object from neighbors
            l. end while
            m. if object(i) is unclassified assign it to noise temporally
            n. end expand cluster
    5- Next i
    6- End
```

This method is efficient in discovering varied shape and size clusters. But have trouble handling varied density clusters. Grid based methods don't work directly with objects; it works with object in the same grid cell as a single unit and is based on some statistical information to merge cells to form clusters. STING algorithm is an example for a method in this category. Reader may be referred to [8].

III. RECENT ADVANCES IN CLUSTERING METHODS

Since clustering algorithms play very important role in data mining, and are involved in many applications from different branches in science like image segmentation, search engine, clustering of genes, graphical information system, drug activity prediction, wireless sensor network's based application and so many other. The most popular algorithms that receive a great attention from the researchers in this field are k-means and DBSCAN. So here, this section concentrates on them. Since k-means is simple and has linear time complexity lot of researchers proposed many ideas to improve it and solve its drawback (value of k, initial centroids, improve its running time). Reference [9] proposed an idea allows the algorithm to discover the actual number of clusters in dataset, i.e. k-means will not require k as an input, and it will detect the value of k based on some calculation. But the time complexity of this version is $o(n^2)$. So this version is not suitable for large datasets. Reference [1] introduces an idea to improve the running time of the algorithm. This idea depends on reducing the calculation while redistributing objects over the new cluster centers based on some information from previous iteration of the algorithm. Reference [10] introduces an idea to improve the quality of resulting clusters and reduce the number of iteration. The idea is to partition the dataset into m block and apply the k-means on each block separately, then apply the k-means on the resulting means from all blocks, the final result is used as the initial centers for the original dataset. By this method the k-means starts with fine initial k cluster centers which lead to improve the quality of final result. Reference [11] introduces parallel implementation of k-means presented in [1]. Reference [12] studies the properties of k-means on several benchmark datasets; this is evidence the popularity and efficiency of the k-means in wide range of applications that use it. In spite of its limitation handling overlapped clusters, varied shaped clusters, and when there is large variance in cluster sizes it is used in large number of application.

The second efficient, popular and is used in various application is DBSCAN. It has received a lot of attention from researcher all over the world. DBSCAN is able to handle clusters of different sizes and shapes but have a trouble handling varied density clusters, so large number of researches proposes ideas to overcome this problem. We list some of them. Reference [13] proposes a simple idea allows Eps to be varying from cluster to another and controls the density of any core object within the cluster. Reference [14] introduces a two phase algorithm; firstly, it computes a density of each object as the average distance to the k-nearest neighbors, and applies the k-means algorithm to divide data into k different levels of density. In second phase, it applies a modified version of DBSCAN algorithm on each level of density. Reference [15] introduces new paradigm to handle varied density clusters. It assigns each object a local density value that is equals to the sum of distance to k1-nearest neighbor. Then it classifies objects into attractors and attracted objects based on counting the objects in k-nearest neighbors that are denser than the object. The clusters are formed from denser objects to sparser ones. This algorithm requires four input parameter, and this is large number. Reference [16] proposes to use mutual k-nearest neighborhood to define k-deviation density of point and uses threshold called density factor (instead of Eps) to find direct density reachable neighbors for core points to expand cluster. GMDBSCAN [17] is a grid technique and is based on spatial index (sp-tree). It allows Minpts to be varied from cell to another based on density of each grid cell. And use the same value for Eps in each grid cell and applies DBSCAN on data in each grid cell. This method uses many parameters that affect the resulting clusters. The Problem of GMDBSCAN is a time consuming to perform well on large datasets.

GMDBSCAN-UR [18] is an adaptation for GMDBSCAN algorithm. In each grid cell, it selects number of well scattered points capture the shape and extent of the dataset as representative object. This method overcomes the problem of time consuming of GMDBSCAN. It allows one parameter of DBSCAN (Minpts or Eps) to be varied from cell to another, its performance is better than that of GMDBSCAN. But it doesn't produce accurate results with varied density clusters. VDBSCAN [19] partitions the k-dist plot based on seeing sharp change on the curve. For each partition, it selects a suitable value for Eps and applies DBSCAN. It performs well when dataset contains clusters of different uniform density. But when gradient in density exist it produces inaccurate clusters. It may split dense cluster or merge sparser ones unless they are well separated. DBSCAN-DLP [20] divides the input dataset into different density level based on some statistical characteristic of density variation, then Estimates Eps value for each density level and adopts DBSCAN clustering on each density level with corresponding Eps. This algorithm is suitable for clusters of uniform density; the variance in density of objects in the same cluster should be very small and less than threshold. Reference [21] introduces a mathematical idea to select several values for Eps form the k-dist plot and applies the DBSCAN algorithm on the data moving from smallest to largest Eps with ignoring the previously clustered objects. It uses spline cubic interpolation to find inflection points on the curve where the curve changes its concavity. This method leads to split some clusters. DSets-DBSCAN [22] runs DSets clustering first, where the input parameters of DBSCAN are determined from the original cluster extracted by DSets. Most previous algorithms estimate the value of Eps based on some local density criteria and apply the DBSCAN to discover clusters from dataset with multi-levels of density.

IV. The proposed method

This section describes the details of the proposed method where cluster is a connected graph with a threshold. The proposed method retrieves the k-nearest neighbors for each object in data, then sums the distances to the most nearest neighbors; at most four neighbors, after that it starts to construct clusters starting from any object in data, this object will be the reference object in current cluster, the weight of the reference object is compared with its k-nearest neighbors weight. The objects with similar weight well be assigned to the current cluster and the cluster will be grown from them based on the similarity between them and the reference object of the current cluster. The proposed method is based on the following definitions:-

Def. 1: Knn(k-nearest neighbors) of object p is a set of objects q_j, j = 1,2,…, k such that dis(p,q_j) is less than or equals to dis(p,q_k). Mathematically, it can be written as in (1) where dis(p,q) is the Euclidean distance.

$$knn(p) = \{q_j \mid dis(p,q_j) \le dis(p,q_k), j = 1,2,...,k\} \quad (1)$$

Def. 2: Weight of object p is the sum of distances to its most nearest neighbors that have large effect on it as in (2).

$$w(p) = \sum_{i=1}^{mostnear} dis(p,q_i) \quad (2)$$

Since any object is effected by small number of objects in its neighbors so mostnear will be assigned values 3 or 4. The object has smaller weight has larger density and the vice versa. Since the clustering process will start from any object in dataset, the algorithm requires low bound and high bound for weigh of objects allowed to be in the same cluster. The algorithm compounds these bounds in one threshold by dividing the weight of current object by the weight of starting object (reference object) in the current cluster and the vice versa, and the algorithm will use the ration that will be less than or equals to one. This can be represented by the following function weight ratio as in (3).

$$WeightRatio(p,q) = \begin{cases} \dfrac{w(p)}{w(q)} & if\ (w(p) \le \mathrm{w(q)}) \\ \dfrac{w(q)}{w(p)} & if\ (w(p) > \mathrm{w(q)}) \end{cases} \quad (3)$$

Def. 3: cluster is a set of connected objects satisfying the condition WeightRatio ≥ SimRatio.

Def. 4: noise is a very small cluster (less than 0.006 of input dataset size).

As the ratio in (3) closes to one the objects have high similarity in weight, as it closes to zero there will be large variance in weight (density) of objects within the same cluster. So the threshold should be range from 0 to 1. Experimentally, threshold is ranging from 0.25 to 0.75 since the objects on cluster edges (border objects) have large weight compared to the weight of core objects in cluster. If the reference object has the largest weight (lowest density) this means the object is border object and the clustering process will move toward smaller weight (larger density) objects within the threshold specified by the user. On the other hand, if the reference object has the smallest weight then the clustering process will move toward larger weight objects within the threshold. This threshold will be referred to as similarity ratio (SimRatio). In each step, the algorithm checks the k-nearest objects from the current object and assigns the unclassified object to the current cluster regardless their counts or the neighborhood radius of current object as in DBSCAN method. So the value of k in k-nearest neighbors should be small, its maximum value equals to three times mostnear. The proposed algorithm works as follow:

Input: dataset, mostnear, k (for k-nearest neighbors)

Output: set of clusters

```
1- All objects are unclassified and clus_id=0.
2- For all objects in dataset retrieves the k-nearest neighbors and computes w(p) as in (2).
3- For i=1 to dataset size
4-  If (p(i) is unclassified then goto 5 else goto 6
5- Expandcluster(dataset, p(i), SimRatio)
       a. clus_id = clus_id + 1
       b. Assign p(i) to cluster clus_id
       c. For j=1 to k
```

```
        d.  If q(j) is unclassified and WeightRatio (p(i), q(j)) $\geq$ SimRatio then
                i.   assign q(j) to cluster clus_id
                ii.  Seedlist.append(q(j))
        e.  Next j
        f.  While Seedlist.size < > 0
        g.  Currobj = Seedlist.first()
        h.  For x=1 to k
        i.  If q(x) is unclassified and WeightRatio (p(i), q(x)) $\geq$ SimRatio then
                i.   assign q(x) to cluster clus_id
                ii.  Seedlist.append(q(x))
        j.  Next x
        k.  Seedlist.delete(Currobj)
        l.  End while
        m.  End Expandcluster
    6-  Next i
    7-  End
```

The most important parameter in this algorithm is similarity ratio (SimRatio). The algorithm produces good quality clusters and requires time complexity of o(n log n) using index structure like R*-tree finding k-nearest neighbors. Next, the experimental results are presented.

V. EXPERIMENTAL RESULTS

We have used many challenging benchmark datasets to test the proposed algorithm. The first dataset (shown in Fig. 1 left-hand) contains 473 objects that are forms clusters with varied shapes, sizes and densities. It contains 6 noise points. The algorithm produces 10 clusters as shown in Fig. 1 right-hand. The challenging part in this dataset is the right region that contains a low density cluster has some high density clusters and one of them has two density clusters. The algorithm discards the noise objects efficiently.

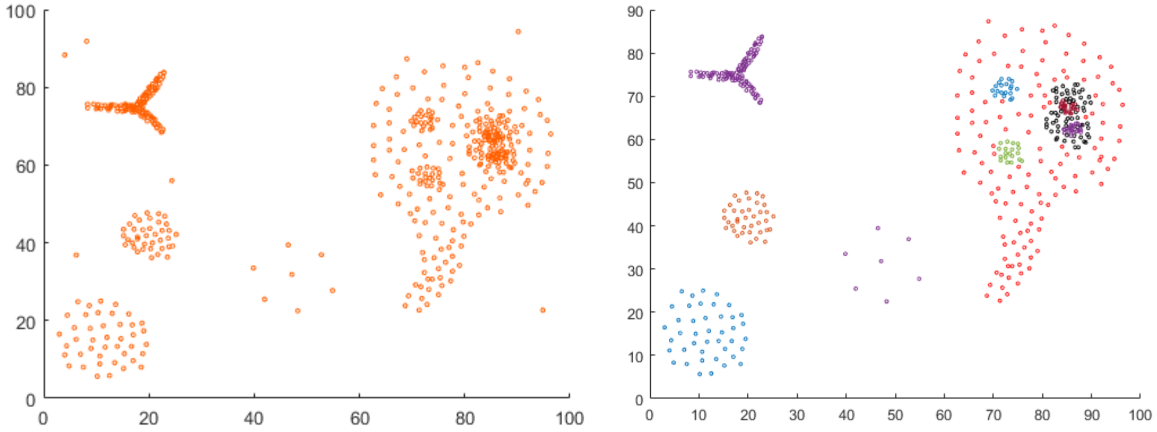

Figure 1. Dataset contains 473 objects on left, and the result of the proposed algorithm is on right.

The second dataset is shown in Fig. 2 which contains 399 objects. This dataset is very challenging since it contains intertwined clusters that are very close to each other. The proposed algorithm produces 7 clusters where the upper left cluster contains two inner dense clusters as shown in Fig. 2 on right-hand. Also, the right cluster has inner high dense cluster with no separation between them and the algorithm discovers them successfully.

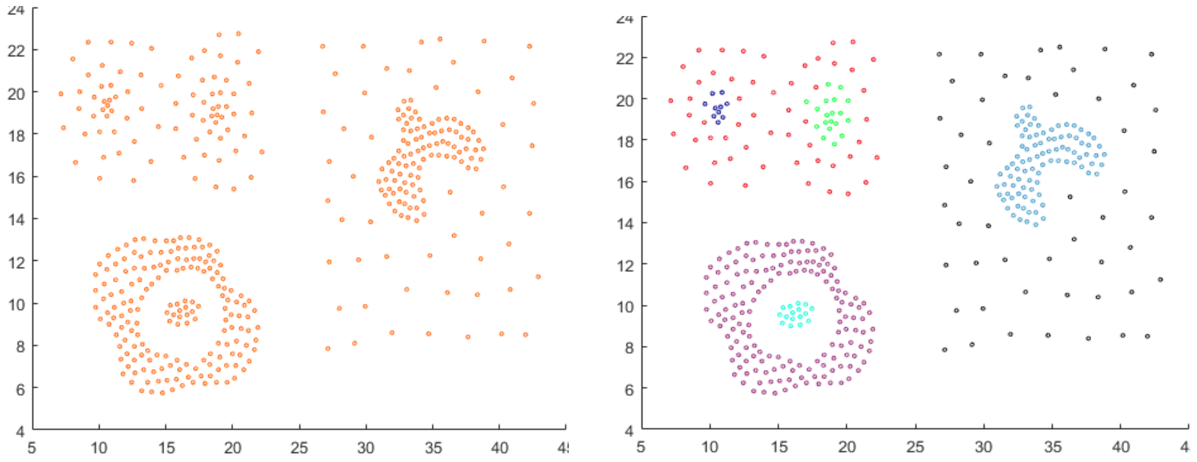

Figure 2. Dataset2 contains 399 objects and the resulting clusters by the proposed algorithm.

The third dataset contains 788 objects is shown in Fig. 3. The challenging in this dataset is the chains that connect between two pair of clusters, but the algorithm cuts the chains and returns the correct clusters as shown in Fig. 3 on right-hand where it produces 7 clusters, all the clusters are very close in density.

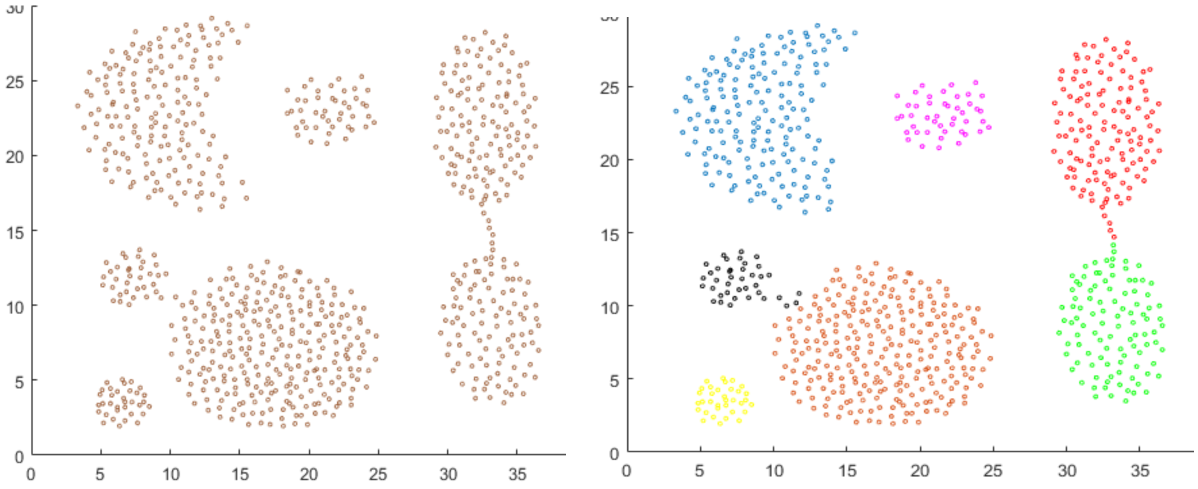

Figure 3. Dataset3 contains 788 objects and the results of the proposed algorithm.

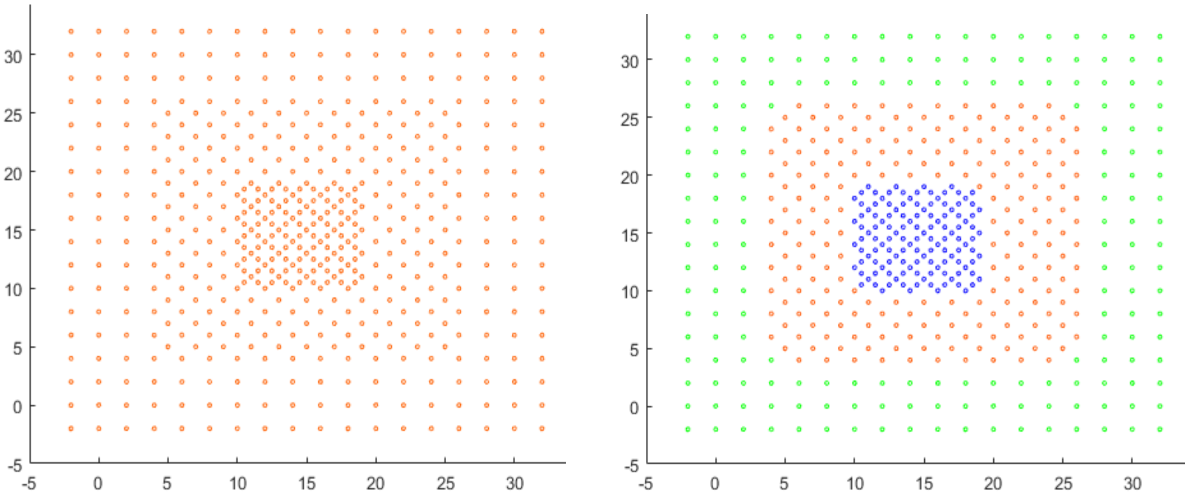

Figure 4. Dataset4 contains 526 objects and the resulting clusters from it by the proposed algorithm.

The fourth dataset has 526 objects as shown in Fig. 4. The challenging in this dataset is the uniform gradient of density. The outer cluster represents the lowest density and has two inner intertwined clusters, again the algorithm discover the clusters correctly as shown in Fig. 4 on right-hand.

The fifth dataset contains 315 objects as shown in Fig. 5. The challenging in this dataset is intertwined clusters of varied shapes especially the cluster takes figure like 2 or z character, but the proposed algorithm discovers the right clusters as shown in Fig. 5 on right-hand.

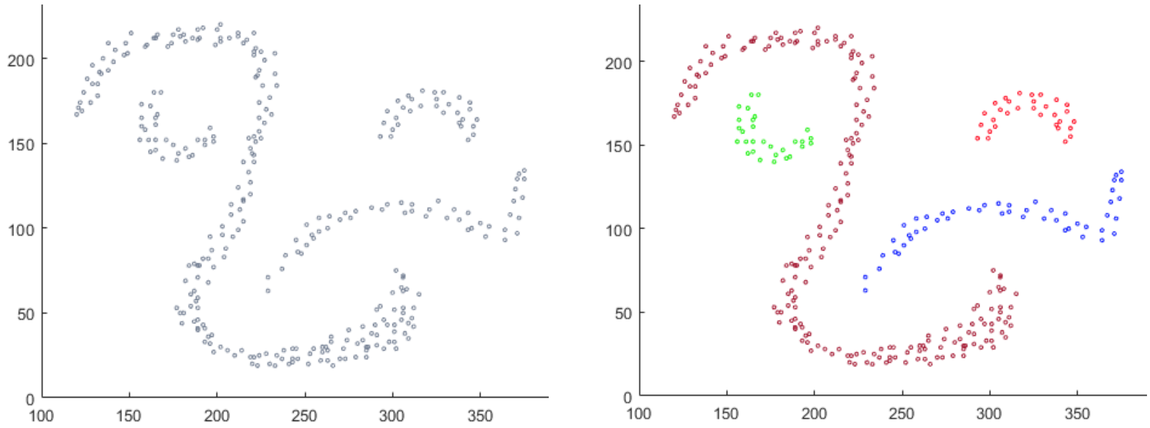

Figure 5. Dataset5 contains 315 objects and the results from the proposed method.

The sixth dataset contains 582 objects form four clusters of different shapes, sizes and density. The proposed algorithm produces the correct clusters and discards noise objects efficiently as shown in Fig. 6.

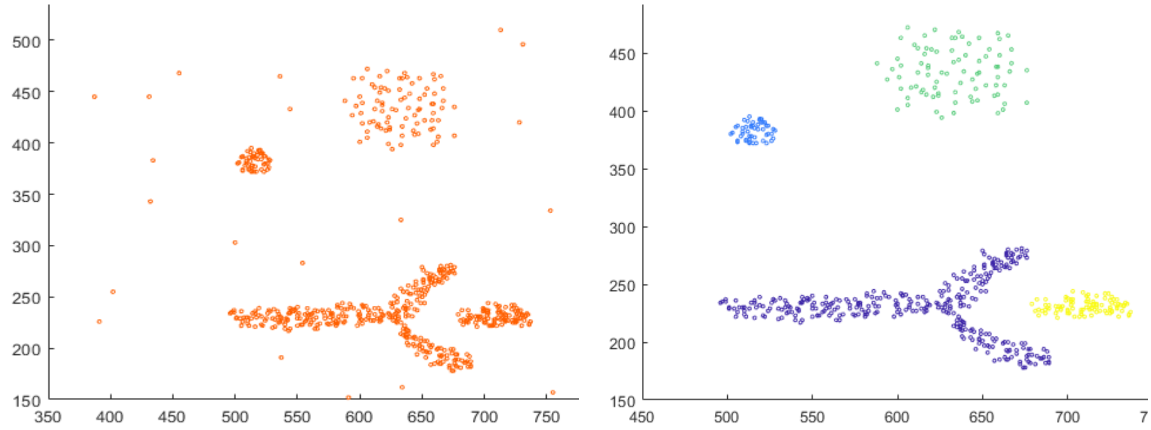

Figure 6. Dataset6 has 582 objects, some of them are noise and the resulting clusters by the proposed algorithm.

The seventh dataset contains 3147 objects as shown in Fig. 7. The proposed algorithm discovers the four correct clusters from the data.

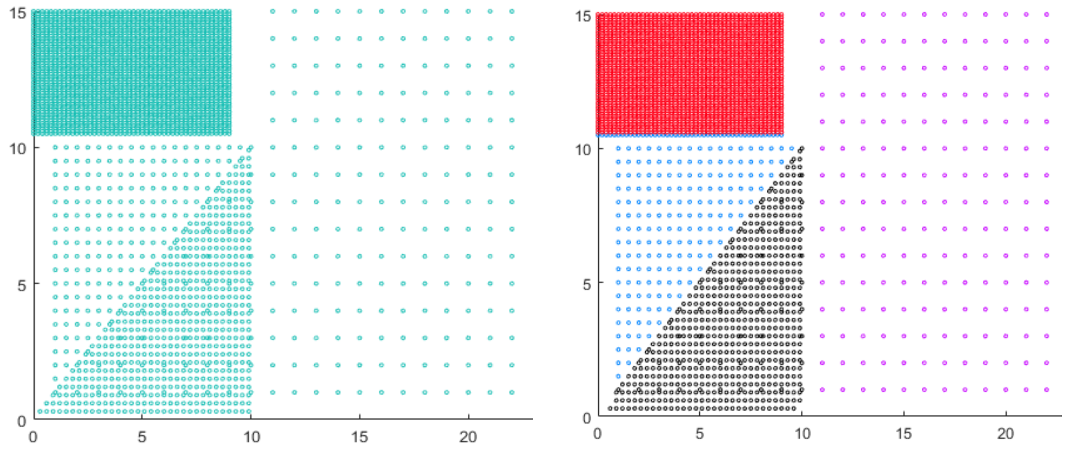

Figure 7. Dataset7 contains 3147 objects, and the resulting clusters from the proposed algorithm.

The eighth dataset contains 8000 objects with noise as shown in Fig. 8. The challenging here is the chain objects among clusters. The proposed algorithm discovers the original six clusters from data and discards the noise point efficiently.

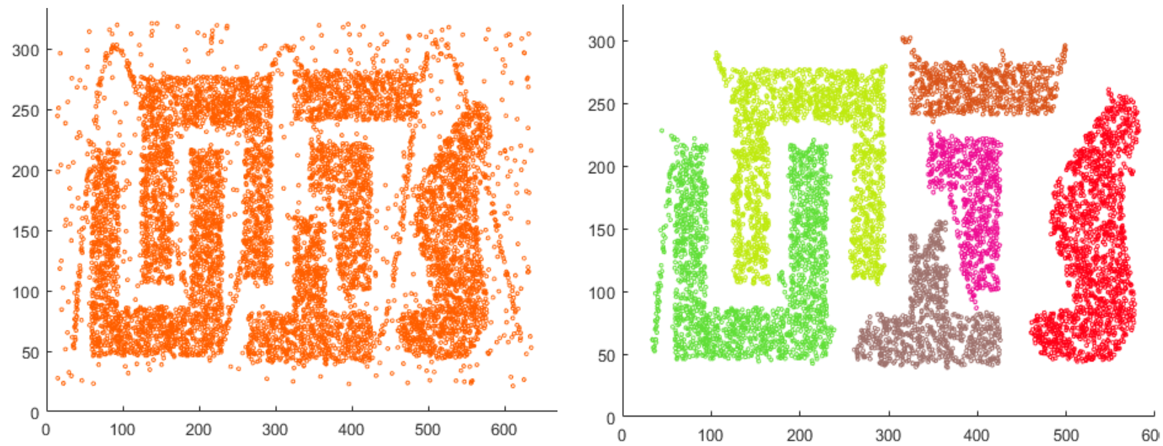

Figure 8. Dataset8 contains 8000 objects with noise, the algorithm discovers the right clusters.

VI. Conclusion

In this paper, we have proposed a new clustering algorithm which introduces new definition for cluster. A cluster is a connected graph where nodes in graph have similar weights compared to the weight of the reference node. The k-nearest neighbor adapts the distance to neighbors where it tends to be large in sparse regions and small in dense regions. The algorithm considers the very small clusters as noise. The experimental results are evidence that the proposed algorithm has the ability to discover clusters with varied shapes, sizes and densities. All these properties together for clusters motivate researcher to propose a lot of adaptation on DBSCAN algorithm to solve these problems together especially clusters with varied densities. We can say that the algorithm requires only two input parameters; they are k for k-nearest neighbors and SimRatio for weight of objects within their clusters. The other parameter which called mostnear is fixed to 3. The value of k is ranging from 4 to 7; finally the value of SimRatio is ranging from 0.25 to 0.75.

We have used many challenging benchmark datasets to test the algorithm. It overcomes all challenges and discovers the correct clusters efficiently.

Acknowledgment

This project was supported by the Deanship of Scientific Research at Prince Sattam Bin Abdulaziz University under the research project no. 2017/01/7120.

References

[1] A. M. Fahim, A. M. Salem, F. A. Torkey, and M. A. Ramadan, “An efficient enhanced k-means clustering algorithm,” Journal of Zhejiang University-Science A, vol 7(10), pp. 1626-1633, 2006.

[2] L. Kaufman and P. J. Rousseeuw, “Partitioning around medoids (program pam), Finding groups in data: an introduction to cluster analysis,” John Wiley & Sons, 1990.

[3] R. T. Ng, and J. Han, “Efficient and effective clustering methods for spatial data mining,” in Proc. of the 20th Int. Conference on Very Large Databases, Santiago, Chile, pp. 145–155, 1994.

[4] R. Sibson, “SLINK: an optimally efficient algorithm for the single-link cluster method,” The Computer Journal, vol.16(1), pp. 30-34, 1973.

[5] S. Guha, R. Rastogi and K. Shim, “Cure: An efficient clustering algorithm for large databases,” in SIGMOD 1998, Proceedings ACM SIGMOD Int. Conference on Management of Data, Seattle,Washington, USA (L. M. Haas and A. Tiwary, eds.), pp. 73-84,1998.

[6] T. Zhang, R. Ramakrishnan and M. Livny, “BIRCH: an efficient data clustering method for very large databases,” SIGMOD '96 Proc. of the 1996 ACM SIGMOD int. conference on Management of data, ACM New York, NY, USA, pp. 103-114, 1996.

[7] M. Ester, H. P. Krigel, J. Sander, and X. Xu, “A Density-based algorithm for discovering clusters in large spatial databases with noise,” in Proc. of the International Conference on Knowledge Discovery and DataMining, pp. 226–231, 1996.

[8] W. Wang, J. Yang, and R. Muntz, “STING: A statistical information grid approach to spatial data mining,” In VLDB Vol. 97, pp. 186-195, August 1997.

[9] Md. Z. Hossain, Md.N. Akhtar, R.B. Ahmad and M. Rahman, “A dynamic K-means clustering for data mining,” Indonesian Journal of Electrical Engineering and Computer Science, vol. 13(2), pp. 521-526, February 2019.

[10] A. M. Fahim, A. M. Salem, F. A. Torkey, M. A. Ramadan, and G. Saake,”An Efficient K-Means with Good Initial Starting Points,” Georgian Electronic Scientific Journal: Computer Science and Telecommunications, No.2(19), pp. 47-57, 2009.

[11] A. M. Fahim, “An Efficient Parallel K-Means On Multi-Core Processors” International Journal of Science, Engineering and Technology Research (IJSETR), Vol. 4(12), pp. 4234-4241, December 2015.

[12] P. Fränti and S. Sieranoja, “K-means properties on six clustering benchmark datasets,” Applied Intelligence, Vol. 48(12), pp 4743-4759, December 2018.

[13] A. Fahim, "Homogeneous Densities Clustering Algorithm", International Journal of Information Technology and Computer Science(IJITCS), Vol.10(10), pp.1-10, 2018.

[14] M. Debnath, P. K. Tripathi and R. Elmasri. “K-DBSCAN: Identifying Spatial Clusters With Differing Density Levels”, International Workshop on Data Mining with Industrial Applications, pp. 51-60, 2015.

[15] A. Fahim, "A Clustering Algorithm based on Local Density of Points", International Journal of Modern Education and Computer Science (IJMECS), Vol.9(12), pp. 9-16, 2017.

[16] C. Jungan, C. Jinyin, Y. Dongyong, and Li Jun, “A *k*-Deviation Density Based Clustering Algorithm”, Mathematical Problems in Engineering, vol. 2018 , pp.1-16, 2018.

[17] C. Xiaoyun, M. Yufang, Z. Yan, and W. Ping, “GMDBSCAN: Multi-density DBSCAN cluster based on grid,” in Proc. of the IEEE International Conference on e-Business Engineering, ICEBE’08, China, pp. 780–783, October 2008.

[18] M. A. Alhanjouri and R. D. Ahmed, “New Density-Based Clustering Technique: GMDBSCAN-UR,” International Journal of Advanced Research in Computer Science, vol. 3(1), pp. 1-9, 2012.

[19] L. Peng, Z. Dong, and W. Naijun, “VDBSCAN: Varied Density Based Spatial Clustering of Applications with Noise,” in Proc. of the ICSSSM’07: 2007 Int. Conference on Service Systems and Service Management, China, 2007.

[20] Z. Xiong, R. Chen, Y. Zhang, and X. Zhang, “Multi-density DBSCAN algorithm based on density levels partitionin” Journal of Information and Computational Science, vol. 9(10), pp. 2739–2749, 2012.

[21] S. Louhichi, M. Gzara and H. Ben Abdallah, “A density based algorithm for discovering clusters with varied density,” 2014 World Congress on Computer Applications and Information Systems (WCCAIS), Hammamet, pp. 1-6, 2014.

[22] J. Hou, H. Gao, and X. Li, “DSets-DBSCAN: a parameter-free clustering algorithm,” IEEE Transactions on Image Processing, vol. 25(7), pp. 3182–3193, 2016.