I.J. Intelligent Systems and Applications, **2020, 6, 1-20**
Published Online December 2020 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijisa.2020.06.01

Finding the Number of Clusters in Data and Better Initial Centers for K-means Algorithm

Ahmed Fahim[1,2]
[1] Faculty of Sciences and Humanitarian Study,Prince Sattam Bin Abdulaziz University, Al-Aflaj, KSA.
E-mail: a.abualeala@psau.edu.sa

[2] Faculty of computers and information, Suez University, Suez, Egypt.
E-mail: ahmmedfahim@yahoo.com

Received: 02 November 2019; Revised: 15 January 2020; Accepted: 16 March 2020; Published: 08 December 2020

Abstract: The k-means is the most well-known algorithm for data clustering in data mining. Its simplicity and speed of convergence to local minima are the most important advantages of it, in addition to its linear time complexity. The most important open problems in this algorithm are the selection of initial centers and the determination of the exact number of clusters in advance. This paper proposes a solution for these two problems together; by adding a preprocess step to get the expected number of clusters in data and better initial centers. There are many researches to solve each of these problems separately, but there is no research to solve both problems together. The preprocess step requires o(n log n); where n is size of the dataset. This preprocess step aims to get initial portioning of data without determining the number of clusters in advance, then computes the means of initial clusters. After that we apply k-means on original data using the resulting information from the preprocess step to get the final clusters. We use many benchmark datasets to test the proposed method. The experimental results show the efficiency of the proposed method.

Index Terms: Data clustering, k in k-means, initial centers in k-means, clustering algorithms.

1. Introduction

Data clustering is an important method in data mining to discover latent knowledge from data. It is an unsupervised learning method, it aims to group similar data together from the collected data, and the dissimilar data objects will be in different groups. Researchers suggested dozen methods to perform this task, these methods may fall into four categories; (a) partitioning (b) hierarchical (c) density and (d) grid based methods. Every method has its advantages and disadvantages. For example k-means has linear time complexity so it handles large datasets very well, but it is suitable only for convex shaped clusters. The most important open issues in k-means are determination the value of k (number of clusters in dataset) in advance, and the selection of starting centers. Many papers have been proposed to select initial centers, for example you may refer to [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]. The other type of papers is dedicated to determine the number of clusters in data in advance, for example you may refer to [15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28]. All these papers are suggested to solve each of these problems separately. Till now there is no proposed research to solve both of these problems together. Next section reviews some of the proposed researches for these problems.

This research proposes a method to solve both of these problems together simultaneously. The idea comes from merging two clustering algorithms; k-means and DBSCAN "Density-based spatial clustering of applications with noise". In [29] the authors mix between k-means and DBSCAN to improve the execution time of DBSCAN algorithm. Other authors use k-means with DBSCAN to solve the problem of clusters with various densities; you may refer to [30]. In [31] authors merge between k-means and single link method to overcome the time complexity and the chain effect which are the main problems of single link method. The most important features of DBSCAN algorithm are that it does not require number of clusters as input parameter from user and handles clusters of diverse shapes and sizes very well. The proposed method merges between DBSCAN and k-means to handle these two important issues of k-means; they are the expected number of clusters (k) and the selection of initial centers. The suggested method uses DBSCAN to get the expected number of clusters, and then computes the mean of each cluster which will be used as initial centers in k-means algorithm. After that it applies k-means on input dataset to cluster all data objects and produces the final means since DBSCAN discards noise objects and outlier.

This article is arranged as follows; section II reviews some of the earlier works, section III presents the suggested method, section IV shows experimental results, and section V concludes the research.

2. Earlier Works

Most clustering methods depend on some user input parameters; some of them can be estimated easily from data and the others cannot be estimated easily. Here we concentrate on convex shaped, compact, well-separated clusters since we use the k-means algorithm. K-means is very efficient partitioning method but it suffers from two problems; setting the value of k and selecting the initial centers. Many ideas have been proposed to solve each of these problems separately. There is no single idea to solve both problems together in the same time. The proposed idea does this task.

Some papers have been proposed for selecting good initial centers in k-means algorithm [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]. We review some of the recent of them. In [14], authors merge between Genetic algorithm and k-means algorithm to find better initial centers, they outline each data point to the average value of its attributes, then arrange the points based on their average values and partition the arranged points into k subsets of equal size, then compute the mean of points in each subset. This method does not guarantee good initial centers, since the points of inverted attribute values will be in the same subset; they have the same average.

In [13], authors introduce tri-level k-means algorithm that applies k-means on data in three continues levels; in the first level it partitions data into k1 clusters, where k1 is less than k. In the second level it applies k-means on large clusters according to their sizes and Standard deviation of points in each cluster to get smaller clusters, such that the total number of resulting clusters from this level is k. In the last level it applies k-means on all data points using the means of clusters resulting from level two. This method introduces a good idea to get better initial centers but the problem of setting the value of k is still exist.

In [11], authors propose a method to select the initial centers. They calculate the range (R) of selected attributes of the dataset as in (1) then divide the range by k to get the length of interval as in (2). The center of the first interval will be equal to the minimum value of attribute plus half of length of interval as in (3). The center of the other intervals is calculated by adding the length of interval to the center of previous interval as in (4)

$$R = \max(attrib) - \min(attrib) \tag{1}$$

$$L = \frac{R}{k} \tag{2}$$

$$c_1 = \min(attrib) + \text{int}(\frac{L}{2}) \tag{3}$$

$$c_i = c_{i-1} + L \ , \ i = 2,3,...,k \tag{4}$$

This method is suitable for single attribute dataset, but for multiple attributes dataset the problem is still exist. Using the probability theory there will be k^d possible centers, where d is the number of attributes in dataset (dimensionality) and k is the required number of clusters.

In [12], authors use the highest and the lowest weighted mean to find initial centroid. This method is suitable only for datasets that contain only two clusters, they do not tell about more than two clusters.

Some researchers propose methods for finding the suitable value for k in k-means [15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28], we review some of them. In [27], the authors introduce a method to detect the value of k, but their method assumes all clusters of the same radius and computes all pairwise distances for all data object that require o(n^2).

Cluster validity indices are considered a useful tool in finding out number of clusters. Validity indices can work only on pre-computed partitions of the dataset. Finding out a number of possible partitions and then validating them - using a validity measure- is a very time consuming process [24].

In [28], authors propose a visual technique to discover tendency for clustering. This method is called VAT "Visual Assessment of Tendency for clustering", this method starts with pairwise dissimilarity matrix and reorders this matrix such that accumulating the smaller dissimilarity values around the diagonal in contagious regions, and plots this matrix as an image. Each darker block represents cluster, from this image they can visually deduce the possible number of clusters in dataset. This method is time consuming. This method has been enhanced in [20] by introducing automatic technique for finding out the dark blocks in the VAT image.

Till now, no research try to solve these two important problems in k-means together, this is our main contribution in this paper. Good selection of initial centers leads to get better clusters and decrease number of iterations, set the right value for k allow finding the correct clusters. Solving these two problem simultaneously guarantee converging to global minima for the objective function (squared error function) used in k-means algorithm which is shown in (5).

$$SEF = \sum_{j=1}^{n} \sum_{i}^{k} \min(\| o_j - c_i \|^2) \quad (5)$$

where o_j is the current object, c_i is the mean of cluster i.

3. Density Partitioning Clustering Algorithm (DPCA)

This section shows the framework of the proposed method. It merges two well-known clustering methods together; they are DBSCAN and k-means algorithm, so this method will be called Density Partitioning Clustering Algorithm (DPCA), firstly the main steps of both algorithms are presented.

DBSCAN steps
Input: Dataset, Eps, MinPts=4
Output: set of clusters

1. Foreach unclassified object in dataset
2. Find neighbors of object in Eps radius
3. If size of neighbor of object ≥ Minpts goto 4 else goto 5
4. Start new cluster for this object

    a. Add all unclassified neighbors to seedlist and assign them to current cluster
    b. While seedlist isn't empty
    c. Object = top object in seedlist
    d. Find neighbors of object in Eps radius
    e. If size of neighbor of object ≥ Minpts goto f else goto g
    f. append all unclassified neighbors to seedlist and assign them to current cluster
    g. Remove object from seedlist
    h. End while

5. Next object in dataset
6. End

All objects satisfying condition in step 3 are called core objects otherwise object may be border or noise object. Reader may be referred to [32]. DBSCAN discovers clusters in dataset without specifying their number in advance, so this step solves the problem of setting the value of k in k-means algorithm. The proposed method computes the mean of each cluster resulting from DBSCAN algorithm, this computation leads to get k means, and these means will be used as initial centers for k-means algorithm. Note that there are some objects do not assigned to any cluster. K-means assigns them to closest cluster so means need to be updated at least once. The difference between the final means and initial means is very small since noise and outlier objects form small percentage of dataset's size.

k-means steps
Input: Dataset, k
Output: k means (one mean for each cluster)

1. Randomly select k initial centers
2. Assign each object to closest cluster
3. Compute new means for clusters
4. Repeat steps 2 and 3 until means unchanged

K-means is very simple, efficiently handle large dataset, and has linear time complexity.

The proposed framework applies DBSCAN on dataset; the result of this phase is set of clusters. Next step, the method finds the mean of all objects in each cluster and counts the means (clusters). In second phase, the framework applies k-means on dataset using the computed means as initial centers and the count of clusters as value for k. The framework is depicted in Fig.1.

Selecting the value of Eps from the k-dist plot in DBSCAN is not a problem since the final output will be the means of convex shaped clusters, as we know k-means algorithm handle clusters of similar size. The most important feature of DBSCAN is that it can handle clusters of varied shapes, and sizes well, so it is very easy for it to discover convex shaped clusters of similar sizes. It is easy task to count the resulting clusters from DBSCAN. As soon as getting clusters from DBSCAN it is very easy to compute the mean for each cluster, these means will be used as initial centers

in k-means algorithm, so we guarantee that no two initial centers belong to the same cluster, since DBSCAN forms clusters in dense regions so each initial center for k-means will be a representative for dense region (cluster). Optimal selection of initial means leads to very small number of iteration in k-means; which is less than 6 experimentally. Since DBSCAN discards noise and outlier objects from the final result, so k-means will assign each object of them (noise and outlier) to the closest cluster (mean). We have tested the proposed method on many benchmark datasets and the result reveals the efficiency of it.

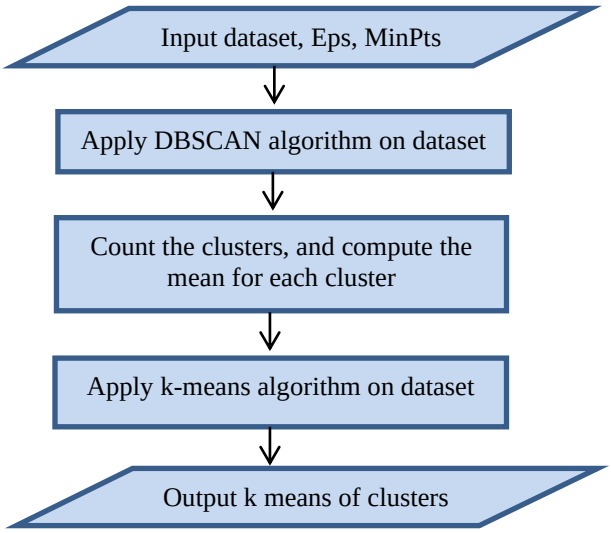

Fig.1. The proposed framework (DPCA)

4. Experimental Results

This section shows some of the results we get from applying the proposed method on some benchmark datasets. First a brief description of these datasets is presented. These datasets can be downloaded from http://cs.uef.fi/sipu/datasets/

A sets (A1, A2, A3): These sets contain spherical shaped clusters of the same size, and the number of clusters k =20, 35 and 50 respectively, each cluster size is 150 objects; the overlap of clusters is 20%. The sets are subsets of each other: A1 $\subset$ A2 $\subset$ A3.

S sets (S1, S2, S3, S4): These sets contain Gaussian clusters with varying overlap that range from 9% to 44%. Most of clusters are spherical shaped, but a few of them have been truncated to resemble non-spherical Gaussian clusters. The set S4 has the strongest overlap. These datasets have the same size which is 5000 objects and contain the same number of clusters which is 15 clusters.

Unbalance: This dataset has 8 clusters in two well separated groups. The first three clusters are of high density and each one contains 2000 objects. The other five clusters are sparse and each one contains 100 objects. The two groups are well-separated so that using large value for Eps results in correct clusters.

R15 dataset has 15 convex shaped Gaussian clusters, dataset size is 600 objects.

D31 dataset has 31 convex shaped Gaussian clusters, dataset size is 3100 objects.

Figures 2 to 11 show the k-dist plot used to select Eps for DBSCAN while MinPts is fixed to 4 in Figures labeled with a. Figures labeled with b depict the clusters resulting from DBSCAN after computing the mean for each cluster. Figures labeled with c show the actual dataset and the initial centers that will be used by k-means. Comparing between Figures b and c you find outlier and noise objects are discarded by DBSCAN algorithm, but k-means assigns all objects in datasets to the closest cluster. Figures labeled with d graphically show the final means for clusters and the objects belonging to each cluster by using different color for each cluster. The black stars represent the means of clusters in Figures b, c, and d.

Fig.2.c shows that clusters are of similar sizes, shapes, and there are overlaps among some clusters. Clusters have similar densities and there are some sparse objects on borders of clusters; these objects are discarded by DBSCAN as shown in Fig.2.b. The proposed method discovers the right clusters as shown in Fig.2.d.

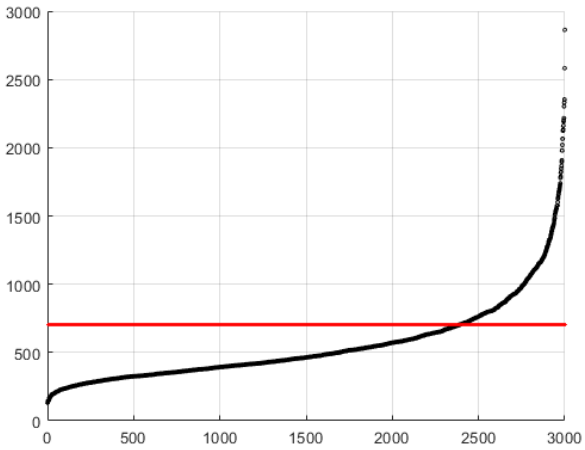

Fig.2.a. 4-dist plot for A1

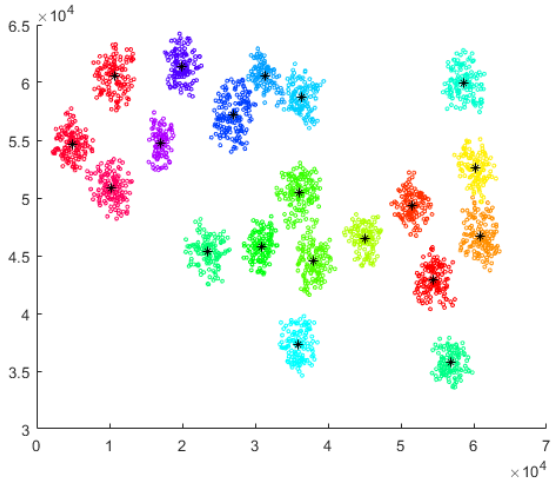

Fig.2.b. Resulting k, and means from DBSCAN

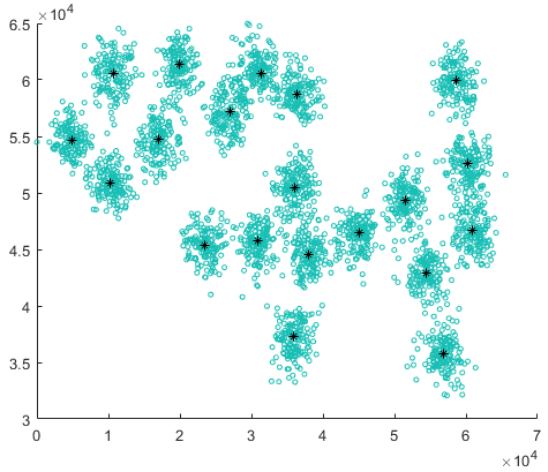

Fig.2.c. Input to k-means (A1, k, means)

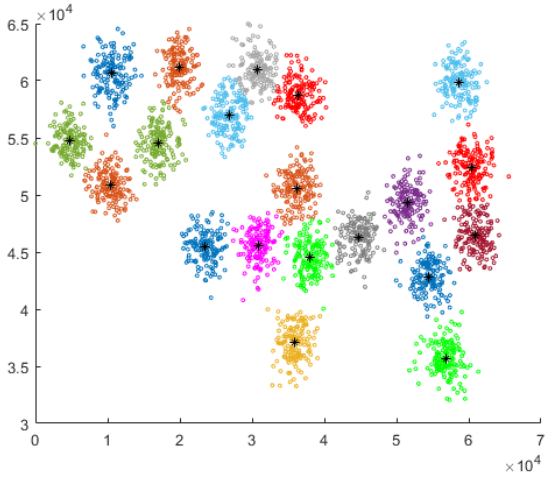

Fig.2.d. Resulting from k-means

Fig.2. The result of applying the DPCA on dataset A1

Fig.3.c shows that clusters are of similar sizes, shapes, and there are overlaps between clusters. Clusters have similar densities and there are some sparse objects on borders of clusters; there are large number of objects have been discarded by DBSCAN as shown in Fig.3.b, Since we use small value for Eps as shown in Fig.3.a. The proposed method discovers the right clusters as shown in Fig.3.d. Comparing between Fig.3.c and Fig.3.d there is very small change between the initial centers and the final ones.

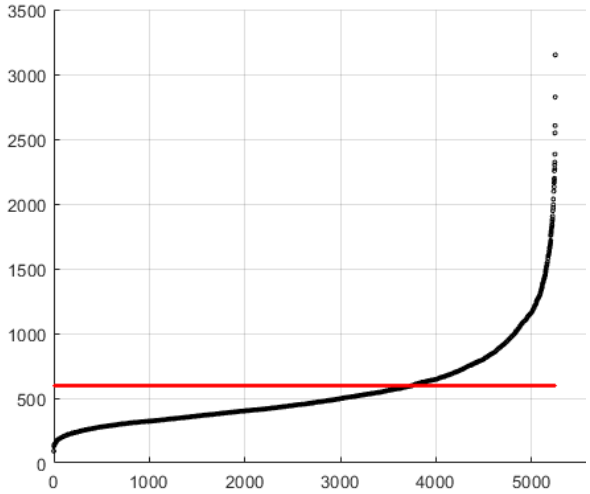

Fig.3.a. 4-dist plot for A2

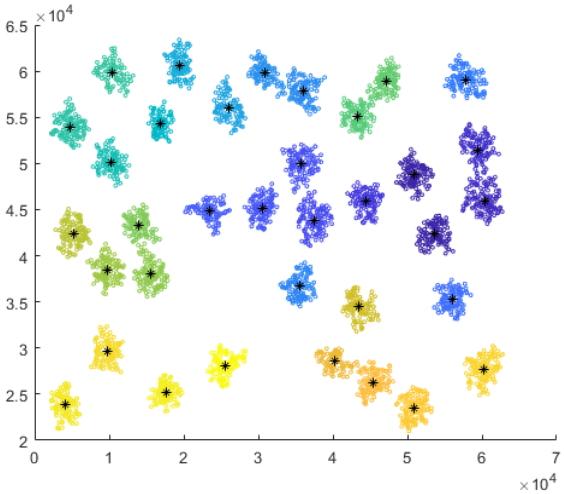

Fig.3.b. Resulting k, and means from DBSCAN

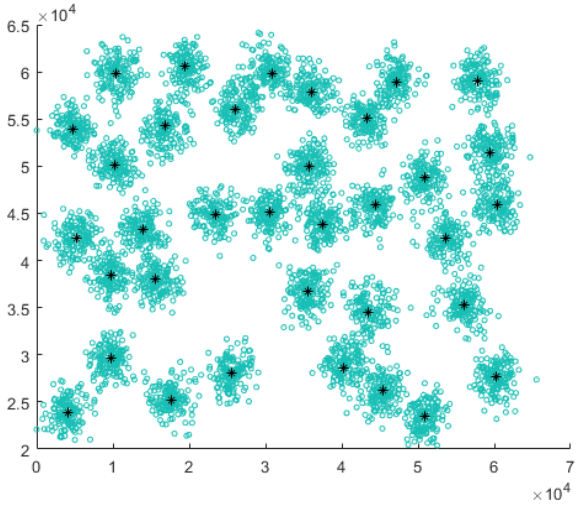

Fig.3.c. Input to k-means (A2, k, means)

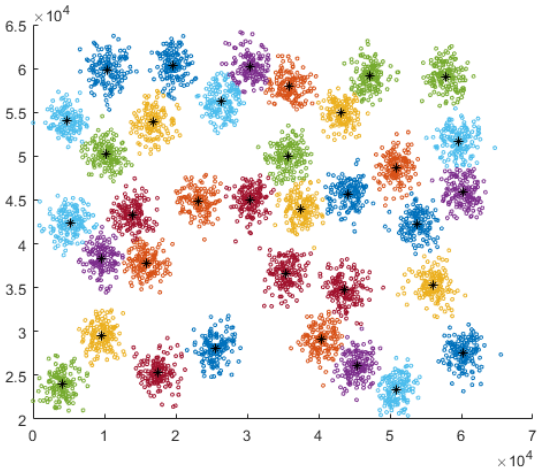

Fig.3.d. Resulting from k-means

Fig.3. The result of applying the DPCA on dataset A2

Fig.4.c shows that the size of dataset is increased and the data space becomes more crowded. Clusters have similar densities and there are less sparse objects on borders of clusters; there are large number of objects have been discarded by DBSCAN as shown in Fig.4.b, Since we use small value for Eps as shown in Fig.4.a. The proposed method discovers the right clusters as shown in Fig.4.d. Comparing between Fig.4.c and Fig.4.d there is very small change between the initial centers and the final ones.

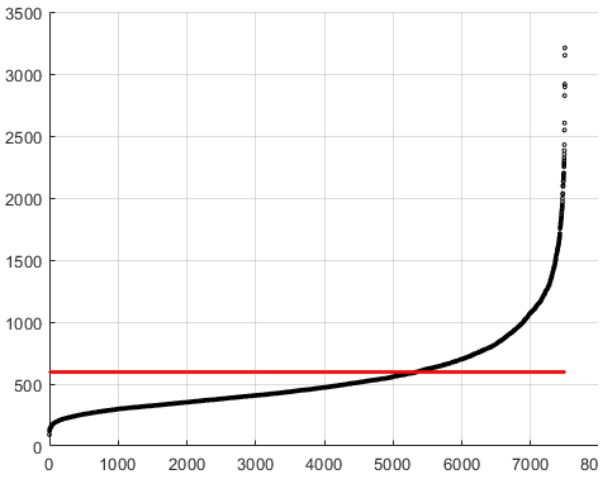

Fig.4.a. 4-dist plot for A3

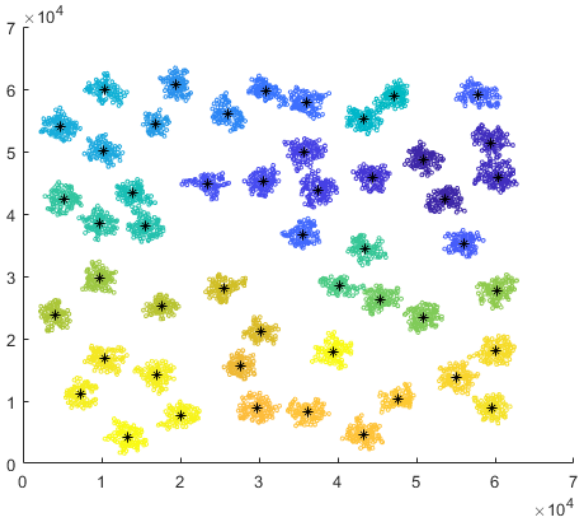

Fig.4.b. Resulting k, and means from DBSCAN

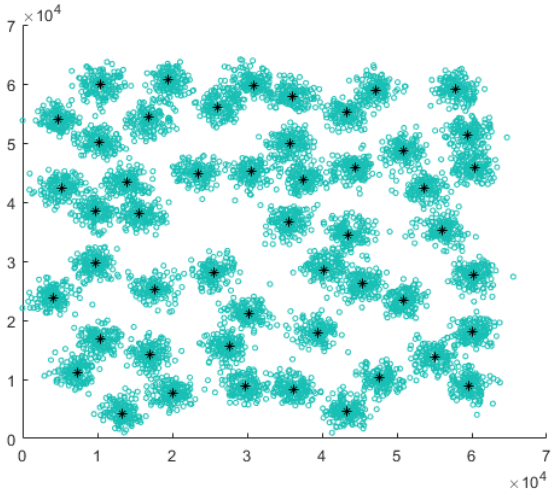

Fig.4.c. Input to k-means (A3, k, means)

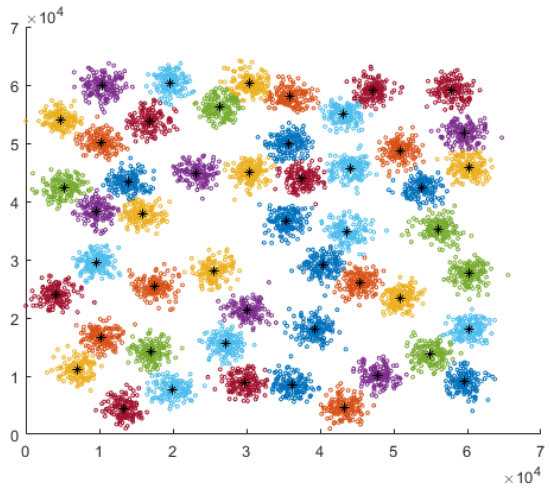

Fig.4.d. Resulting from k-means

Fig.4. The result of applying the DPCA on dataset A3

Fig.5.c shows that dataset contains some elongated clusters and most clusters are of spherical shaped. Clusters have similar densities and are well separated; there are large number of objects have been discarded by DBSCAN as shown in Fig.5.b, Since we use small value for Eps as shown in Fig.5.a. The proposed method discovers the right clusters as shown in Fig.5.d. Comparing between Fig.5.c and Fig.5.d you find that the initial centers and the final ones are almost the same.

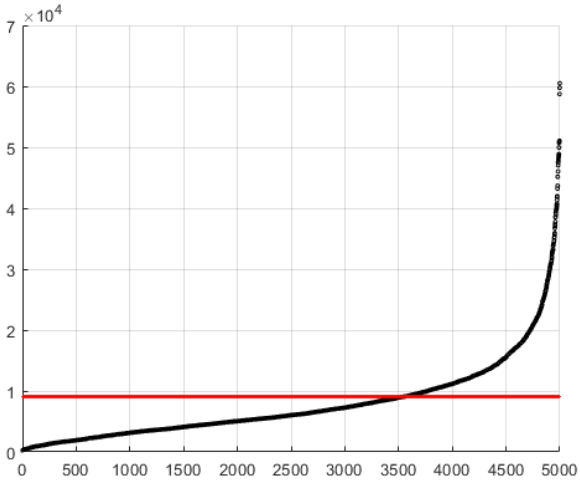

Fig.5.a. 4-dist plot for S1

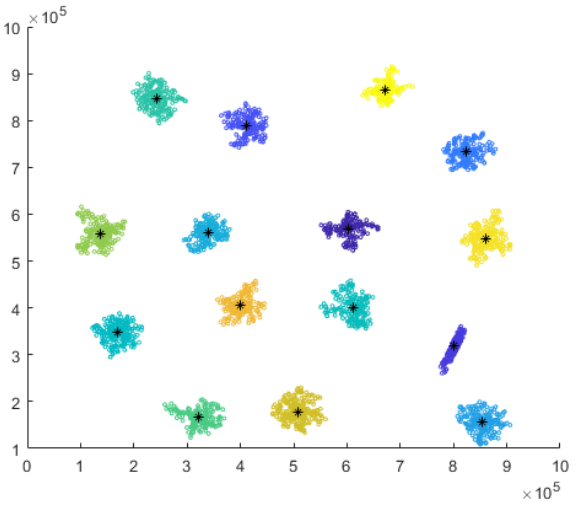

Fig.5.b. Resulting k, and means from DBSCAN

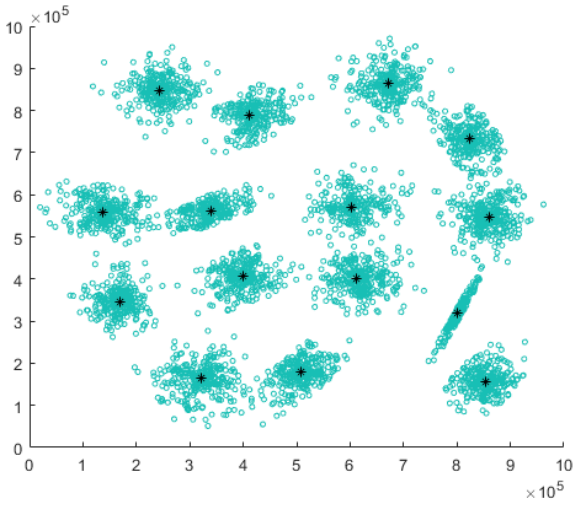

Fig.5.c. Input to k-means (S1, k, means)

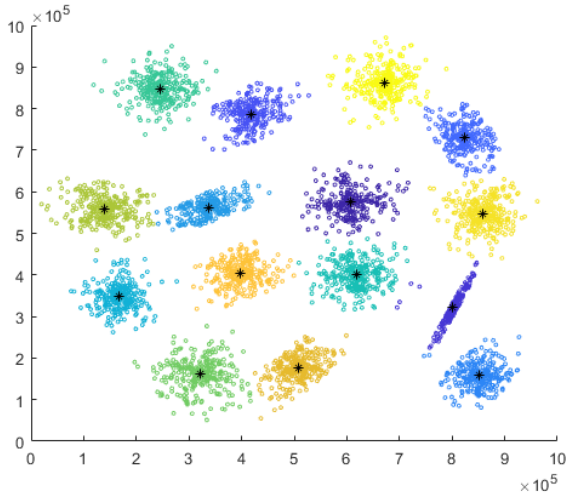

Fig.5.d. Resulting from k-means

Fig.5. The result of applying the DPCA on dataset S1

Fig.6.c shows that dataset contains some elongated clusters and most of clusters are spherical. Clusters have similar densities and are not well separated; there are large number of objects have been discarded by DBSCAN as shown in Fig.6.b, Since we use small value for Eps as shown in Fig.6.a. The proposed method discovers the right clusters as shown in Fig.6.d. Comparing between Fig.6.c and Fig.6.d the initial centers and the final ones are almost the same.

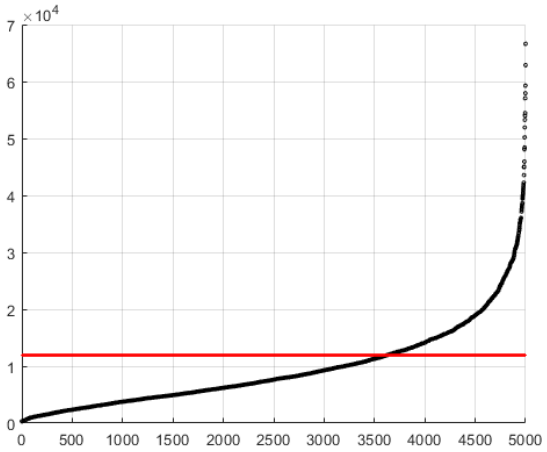

Fig.6.a. 4-dist plot for S2

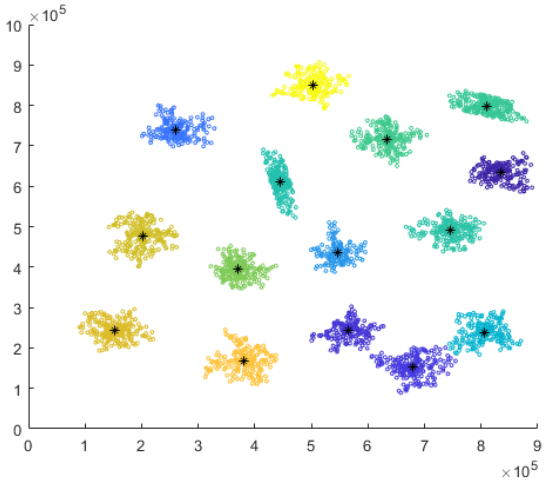

Fig.6.b. Resulting k, and means from DBSCAN

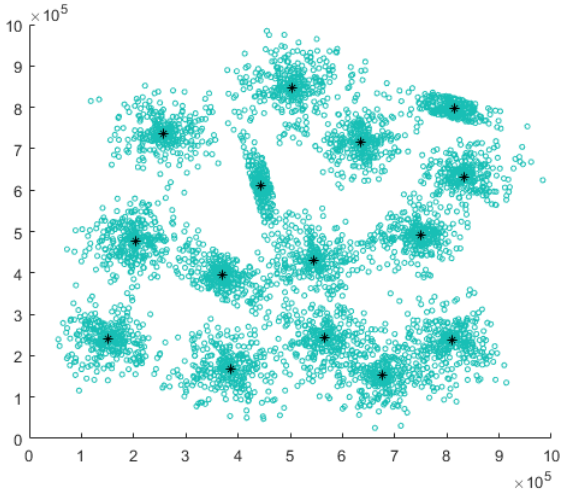

Fig.6.c. Input to k-means (S2, k, means)

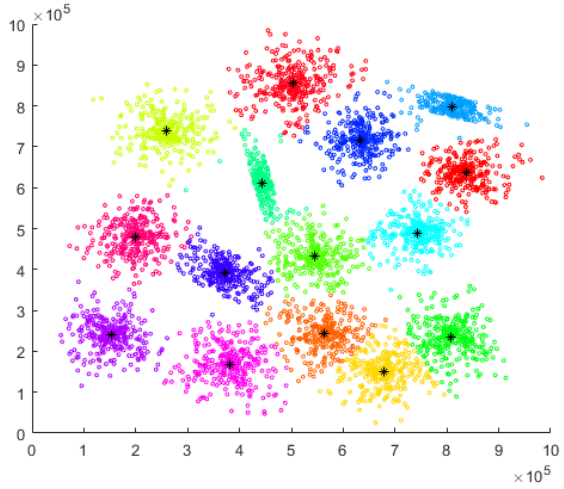

Fig.6.d. Resulting from k-means

Fig.6. The result of applying the DPCA on dataset S2

Fig.7.c shows that dataset contains one elongated cluster and the other clusters are spherical. This dataset is challenging for applying original k-means only; there are no separations between clusters. There is large overlap between clusters; there are large number of objects have been discarded by DBSCAN as shown in Fig.7.b, Since we use very small value for Eps as shown in Fig.7.a. The proposed method discovers the right clusters as shown in Fig.7.d. Comparing between Fig.7.c and Fig.7.d there is very small change between the initial centers and the final ones.

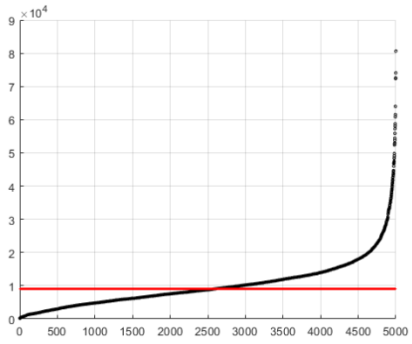

Fig.7.a. 4-dist plot for S3

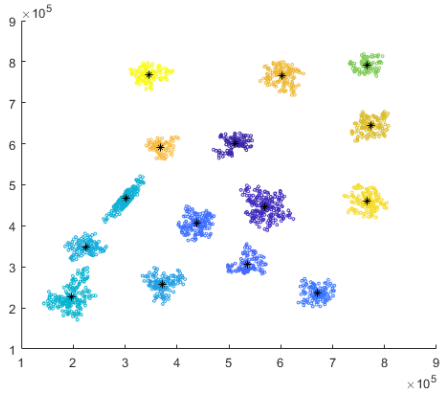

Fig.7.b. Resulting k, and means from DBSCAN

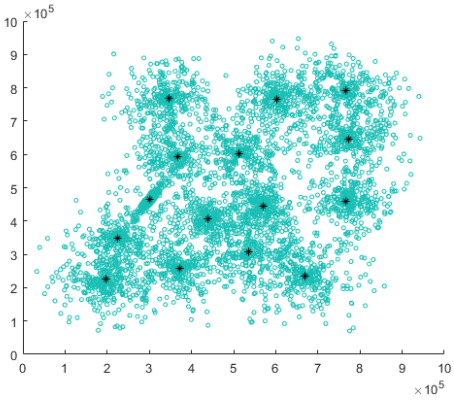

Fig.7.c. Input to k-means (S3, k, means)

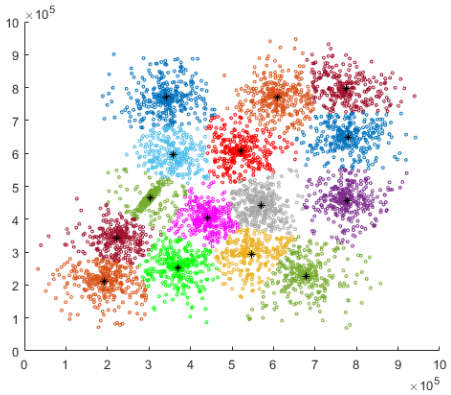

Fig.7.d. Resulting from k-means

Fig.7. The result of applying the DPCA on dataset S3

Fig.8.c shows that dataset contains three elongated clusters and the others are spherical. This dataset is the most challenging for applying original k-means only; there are no separations among clusters. All clusters are overlapped with each other; about half of objects have been discarded by DBSCAN as shown in Fig.8.b, since we use very small value for Eps as shown in Fig.8.a. The proposed method discovers the right clusters as shown in Fig.8.d. Comparing between Fig.8.c and Fig.8.d there is very small change between the initial centers and the final ones.

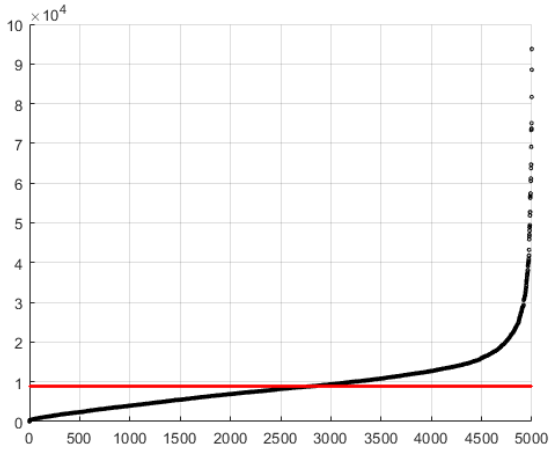

Fig.8.a. 4-dist plot for S4

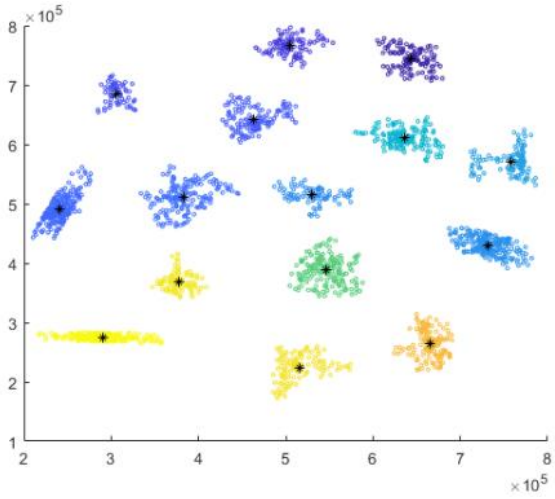

Fig.8.b. Resulting k, and means from DBSCAN

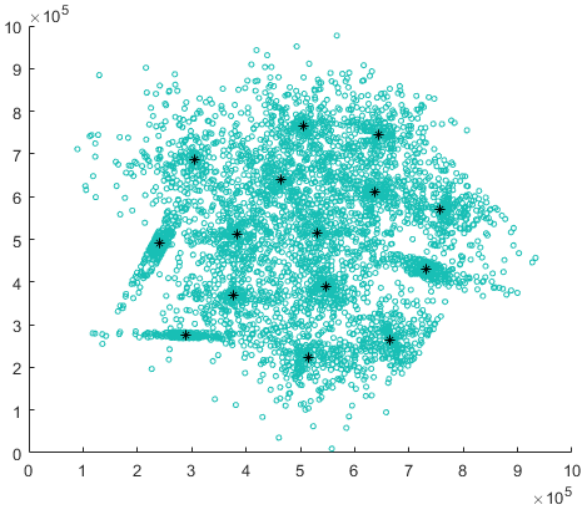

Fig.8.c. Input to k-means (S4, k, means)

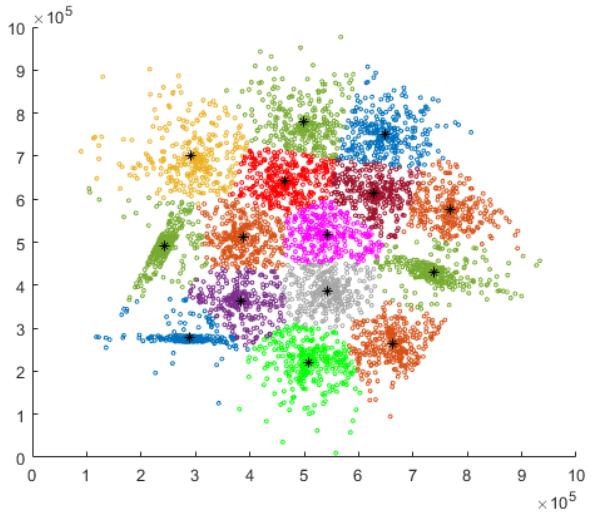

Fig.8.d. Resulting from k-means

Fig.8. The result of applying the DPCA on dataset S4

Fig.9.c shows that dataset contains 8 clusters in two levels of density, all clusters are spherical. The clusters are well separated. So DBSCAN can discover these clusters using large value for Eps - as shown in Fig.9.a - to discover the five low density clusters as shown in Fig.9.b. The proposed method discovers the right clusters as shown in Fig.9.d. Comparing between Fig.9.c and Fig.9.d the initial centers and the final ones are almost the same.

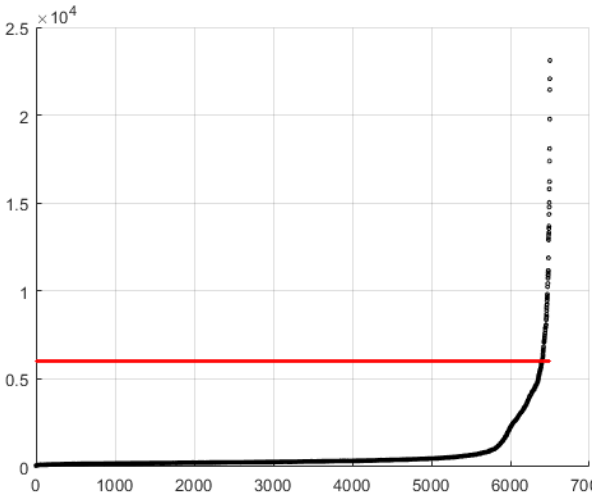

Fig.9.a. 4-dist plot for unblanced

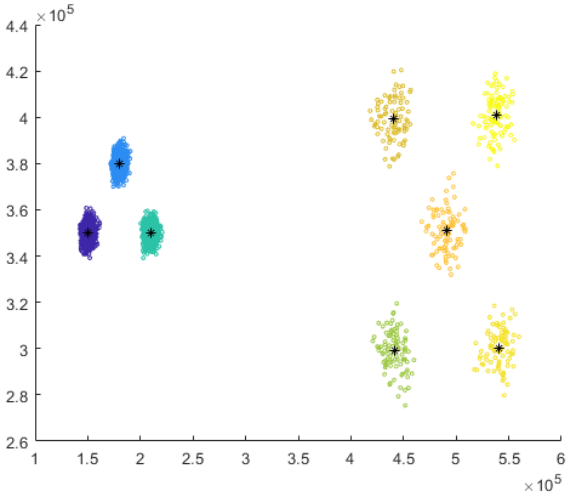

Fig.9.b. Resulting k, and means from DBSCAN

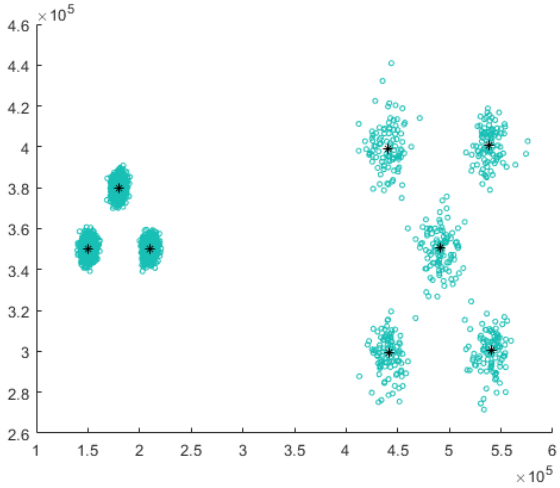

Fig.9.c. Input to k-means (unblanced, k, means)

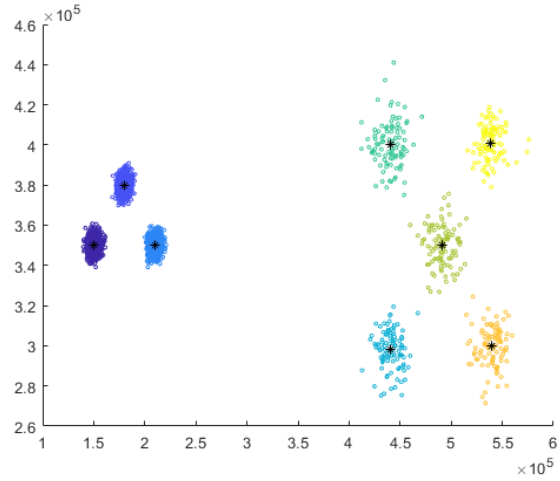

Fig.9.d. Resulting from k-means

Fig.9. The result of applying the DPCA on dataset unblanced

Fig.10.c shows that dataset contains separated spherical shaped clusters of the same densities. There are few number of objects have been discarded by DBSCAN as shown in Fig.10.b, Since we use appropriate value for Eps as shown in Fig.10.a. The proposed method discovers the right clusters as shown in Fig.10.d. Comparing between Fig.10.c and Fig.10.d the initial centers and the final ones are almost the same.

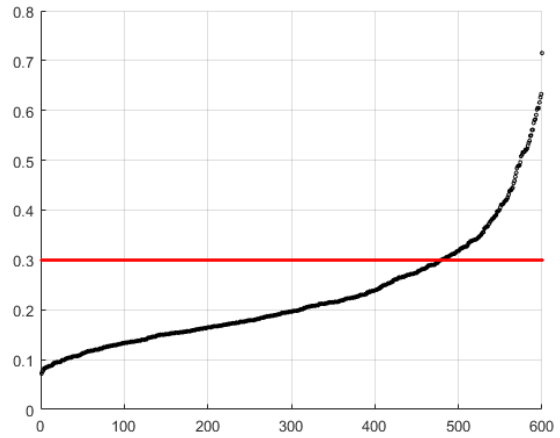

Fig.10.a. 4-dist plot for R15

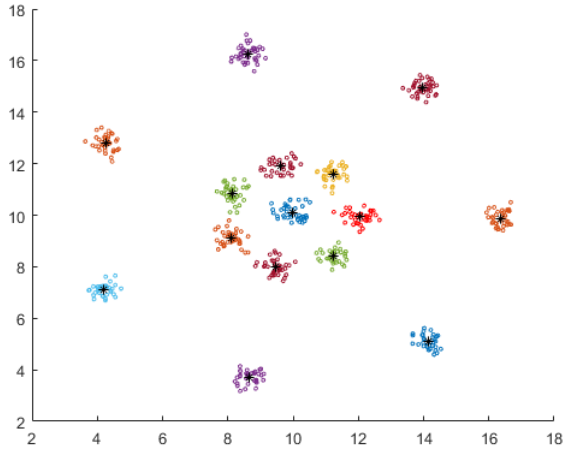

Fig.10.b. Resulting k, and means from DBSCAN

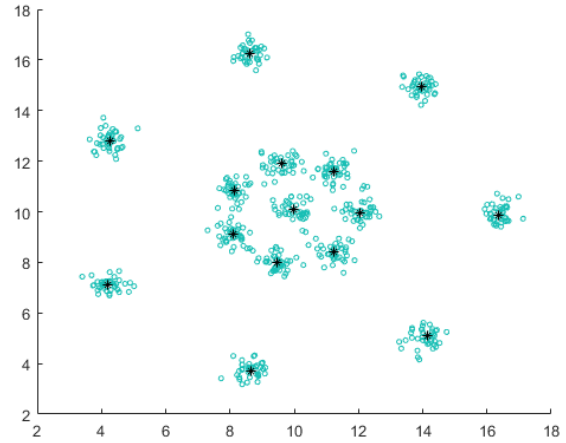

Fig.10.c. Input to k-means (R15, k, means)

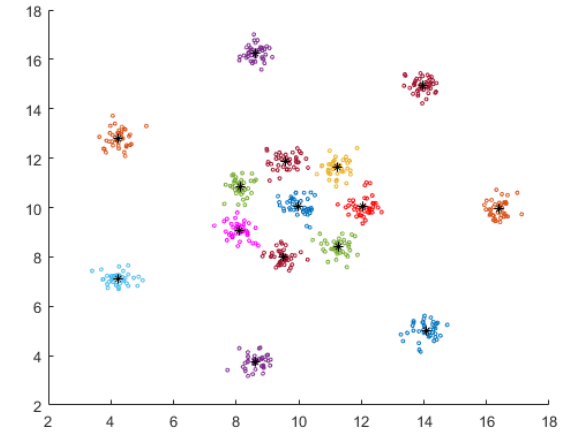

Fig.10.d. Resulting from k-means

Fig.10. The result of applying the DPCA on dataset R15

Fig.11.c shows that dataset contains spherical shaped clusters that are overlapped. Clusters have similar densities. Most sparse objects have been discarded by DBSCAN as shown in Fig.11.b, since we use good value for Eps as shown in Fig.11.a. The proposed method discovers the right clusters as shown in Fig.11.d. Comparing between Fig.11.c and Fig.11.d the initial centers and the final ones are almost the same.

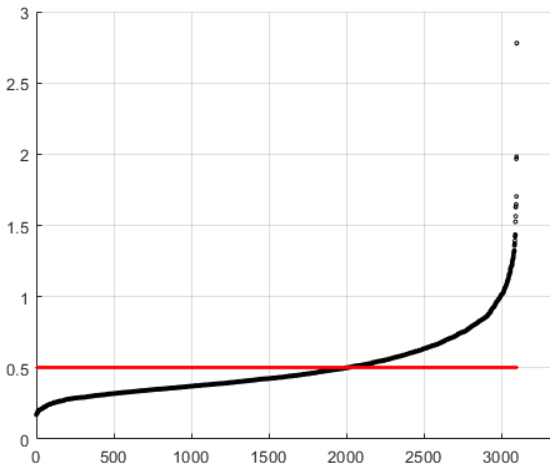

Fig.11.a. 8-dist plot for D31

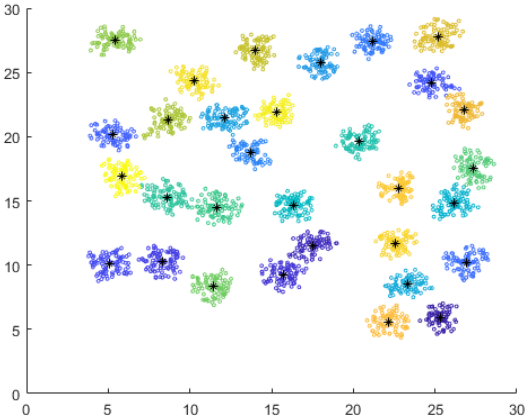

Fig.11.b. Resulting k, and means from DBSCAN

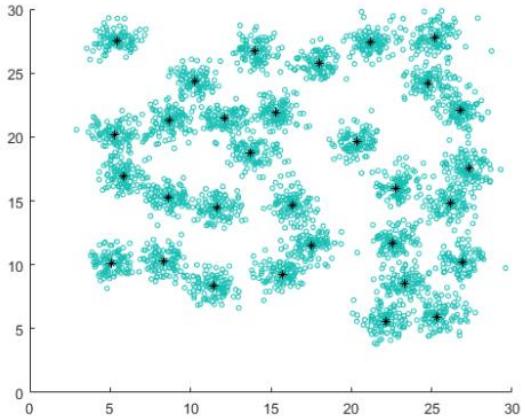

Fig.11.c. Input to k-means (D31, k, means)

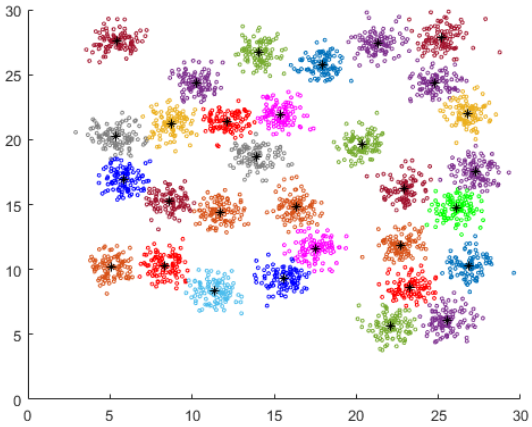

Fig.11.d. Resulting from k-means

Fig.11. The result of applying the DPCA on dataset D31

The following Table 1 shows the actual size of each dataset, and the input value for DBSCAN parameter; you note that MinPts = 4 for all datasets except the last one MinPts = 8. The value of Eps we get it from the k-dist plot. K in table refers to the discovered number of clusters by DBSCAN and this number is used as a value for k in k-means, and the mean value of objects in each cluster is used as initial center for a cluster in k-means, the last column shows that the number of iteration in k-means which is very small due to the good selection for the initial centers for clusters. These results are great evidence the efficiency of the proposed method for solving the two main problem of k-means algorithm.

Table 1. Characteristic of Datasets with the Resulting Clusters

Dataset name	size	minpts	eps	k	iteration
Unbk8	6500	4	5000	8	2
S1	5000	4	9000	15	2
S2	5000	4	9000	15	2
S3	5000	4	9000	15	2
S4	5000	4	9000	15	2
A1	3000	4	700	20	5
A2	5250	4	600	35	3
A3	7500	4	600	50	3
R15	600	4	0.3	15	3
D31	3100	8	0.5	31	4

5. Conclusion

K-means is very efficient clustering method for handling large scale dataset; but it is suitable for datasets that contain clusters of similar sizes and convex shapes. This algorithm suffers from two main problems affecting the quality of its final result; they are the determination of the number of clusters in advance and the selection of initial centers. This paper introduces a very simple idea to solve these two problems together. This paper benefits from the properties of DBSCAN where it does not require the number of cluster in advance and its ability to handle clusters of varied shapes and sizes. While k-means prefers convex shaped clusters of similar sizes, so it is good idea to use the resulting clusters from DBSCAN and count them, compute the mean value of objects in each cluster, and then use this information in k-means, so the quality of clusters is enhanced, number of iterations is decreased.

We have tested the proposed method on various datasets, and we get very promised results that prove the superiority of it. In future work, we will study to parallelize this method to improve the scalability of the method.

Acknowledgements

The author would like to thank the reviewers for their comments that improved the quality of the manuscript. This publication was supported by the Deanship of Scientific Research at Prince Sattam bin Abdulaziz University, Alkharj, Saudi Arabia.

References

[1] P. S. Bradley, and U. M. Fayyad, "Refining Initial Points for K-Means Clustering." In ICML 98, pp. 91-99, July 1998.

[2] A. M. Fahim, A. M. Salem, F. A. Torkey, M. A. Ramadan, and G. Saake, "An efficient k-means with good initial starting points." Computer Sciences and Telecommunications, 2009, vol. 2(19), pp. 47-57.

[3] C. Zhang, and S. Xia, "K-means clustering algorithm with improved initial center." In 2009 Second International Workshop on Knowledge Discovery and Data Mining, IEEE, 2009, pp. 790-792.

[4] M. Yedla, S. R. Pathakota, and T. M. Srinivasa, "Enhancing K-means clustering algorithm with improved initial center." International Journal of computer science and information technologies, 2010, vol. 1(2), pp. 121-125.

[5] M. Erisoglu, N. Calis, and S. Sakallioglu, "A new algorithm for initial cluster centers in k-means algorithm." Pattern Recognition Letters, 2011, vol. 32(14), pp. 1701-1705.

[6] C. S. Li, "Cluster center initialization method for k-means algorithm over data sets with two clusters." Procedia Engineering, 2011, vol. 24, pp. 324-328.

[7] A. Alrabea, A. V. Senthilkumar, H. Al-Shalabi, and A. Bader, "Enhancing k-means algorithm with initial cluster centers derived from data partitioning along the data axis with PCA." Journal of Advances in Computer Networks, 2013, vol. 1(2), pp.137-142.

[8] G. Sathiya, and P. Kavitha, "An efficient enhanced K-means approach with improved initial cluster centers." Middle-East Journal of Scientific Research, 2014, vol. 20(1), pp. 485-491.

[9] R. Mawati, I. M. Sumertajaya, and F. M. Afendi, "Modified Centroid Selection Method of K-Means Clustering." IOSR Journal of Mathematics, 2014, vol. 10(2), pp. 49-53.

[10] R. Suryawanshi, and S. Puthran, "A Novel Approach for Data Clustering using Improved Kmeans Algorithm." International Journal of Computer Applications, 2016, vol. 142(12), pp. 13-18.

[11] PL. Chithra, and Jeyapriya.U, "Premeditated initial points for K-Means Clustering." International Journal of Computer Science and Information Security IJCSIS, 2017, vol. 15(9), pp. 278- 281.

[12] A. C. Fabregas, B. D. Gerardo, B. T. Tanguilig, "Enhanced initial centroids for k-means algorithm." International Journal of Information Technology and Computer Science, 2017, vol. 9(1), pp. 26-33.

[13] S. S. Yu, S. W. Chu, C. M. Wang, Y. K. Chan, and T. C. Chang, "Two improved k-means algorithms." Applied Soft Computing, 2018,vol. 68, pp. 747-755.

[14] S. Awawdeh, A. Edinat, and A. Sleit, "An Enhanced K-means Clustering Algorithm for Multi-attributes Data." International Journal of Computer Science and Information Security (IJCSIS), 2019, vol. 17(2), pp. 1-6.

[15] M. Yan "Determining the number of clusters using the weighted gap statistic." Biometrics, 2007, vol. 63, pp. 1031–1037.

[16] C. A. Sugar, and G. M. James, "Finding the number of clusters in a data set: an information-theoretic approach." Journal of the American Statistical Association, 2003, 98:463, pp. 750-763, DOI:10.1198/016214503000000666.

[17] D. Kim, Y. Park, D. Park, "A novel validity index for determination of the optimal number of clusters." IEICE Tans Inf Syst, 2001, E84, pp. 281–285.

[18] G. Bel Mufti, P. Bertrand, and L. El Moubarki, "Determining the number of groups from measures of cluster stability," In: Proceedings of International Symposium on Applied Stochastic Models and Data Analysis. Brest: France, 2005, pp. 404–412.

[19] S. S. Chae, J. L. Dubien, and W. D. Warde, "A method of predicting the number of clusters using Rand's statistic." Computational Statistics & Data Analysis 50 (2006), pp. 3531 -3546.

[20] M. K. Pakhira," Finding Number of Clusters before Finding Clusters." Procedia Technology, 4, pp. 27 – 37, 2012.

[21] A. Kane, "Determining the number of clusters for a k-means clustering algorithm." Indian Journal of Computer Science and Engineering (IJCSE), 2012, vol. 3 (5), pp. 670 – 672.

[22] D. T. Pham, S. S. Dimov, and C. D. Nguyen, "Selection of K in K-means clustering." Mechanical Engineering Science, 2005, 219, pp. 103 – 119.

[23] A. Fujita, D. Y. Takahashi, and A. G. Patriota, "A non-parametric method to estimate the number of clusters." Computational Statistics and Data Analysis, 2014, vol. 73, pp. 27–39.

[24] C. Subbalakshmi, G. R. Krishna, S. K. M. Rao, and P. V. Rao, "A Method to Find Optimum Number of Clusters Based on Fuzzy Silhouette on Dynamic Data Set." Procedia Computer Science, 2015, 46, pp. 346 – 353.

[25] D. Steinley, M. J. Brusco, "Choosing the Number of Clusters in K-Means Clustering." Psychological Methods, 2011, 16(3), pp. 285–297.

[26] A. M. Mehar, K. Matawie, and A. Maeder, "Determining an Optimal Value of K in K-means Clustering." IEEE international conference on Bioinformatics and Biomedicine, 2013, pp. 51-55.

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

[28] J. C. Bezdek and R. J. Hathaway, "VAT: A tool for visual assessment of (cluster) tendency," in Proc. Intl. Joint Conf. on Neural Networks. Honohulu, HI, 2002, pp. 2225-2230.

[29] A. Fahim, A. M. Salem, F. Torkey, M. Ramadan, and G. Saake, "Scalable varied density clustering algorithm for large datasets." Journal of Software Engineering and Applications, 2010, vol. 3(06), pp. 593-602.

[30] M. Debnath, P. K. Tripathi, and R. Elmasri, "K-DBSCAN: Identifying spatial clusters with differing density levels," in Proceedings of the 2015 International Workshop on Data Mining with Industrial Applications, DMIA 2015, Paraguay, September, 2015, pp. 51–60.

[31] A. M. Fahim, A. M. Salem, F. A. Torkey, M. A. Ramadan, "Hierarchical Clustering Based on K-Means as Local Sample (HCKM)." Egyptian Computer Science Journal, 2007, vol. 29(1), pp. 26-35.

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

Authors' Profiles

Ahmed Fahim was born in Menoufia, Egypt, 1976. He received his B.S., M.S., and Ph.D. in computer science from Faculty of Sciences, Menoufia University, Egypt, in 1998, 2004, and 2010 respectively. He is working at Faculty of computers and information, Suez University, Suez, Egypt. Now he is working at prince Sattam Bin Abdulaziz University, KSA. He is interested in data mining and knowledge discovery, and has published some research papers in different international journals and conferences.

How to cite this paper: Ahmed Fahim, "Finding the Number of Clusters in Data and Better Initial Centers for K-means Algorithm", International Journal of Intelligent Systems and Applications(IJISA), Vol.12, No.6, pp.1-20, 2020. DOI: 10.5815/ijisa.2020.06.01