Decision Science Letters 9 (2020) 51–58

Contents lists available at GrowingScience

Decision Science Letters

homepage: www.GrowingScience.com/dsl

Distance based k-means clustering algorithm for determining number of clusters for high dimensional data

Mohamed Cassim Alibuhtto[a*] and Nor Idayu Mahat[b]

[a]*Department of Mathematical Sciences, Faculty of Applied Sciences, South Eastern University of Sri Lanka, Sri Lanka*
[b]*Department of Mathematics and Statistics, School of Quantitative Sciences, Universiti Utara Malaysia, Malaysia*

C H R O N I C L E

Article history:
Received March 23, 2019
Received in revised format: August 12, 2019
Accepted August 12, 2019
Available online August 12, 2019

Keywords:
Clustering
High Dimensional Data
K-means algorithm
Optimal Cluster
Simulation

A B S T R A C T

Clustering is one of the most common unsupervised data mining classification techniques for splitting objects into a set of meaningful groups. However, the traditional k-means algorithm is not applicable to retrieve useful information / clusters, particularly when there is an overwhelming growth of multidimensional data. Therefore, it is necessary to introduce a new strategy to determine the optimal number of clusters. To improve the clustering task on high dimensional data sets, the distance based k-means algorithm is proposed. The proposed algorithm is tested using eighteen sets of normal and non-normal multivariate simulation data under various combinations. Evidence gathered from the simulation reveal that the proposed algorithm is capable of identifying the exact number of clusters.

1. Introduction

The amount of data collected daily is increasing, but only part of the data that can be used to extract information which are valuable. This has led to data mining, a process of extracting interesting and useful information in the form of relations, and pattern (knowledge) from huge amount of data (Ramageri, 2010; Thakur & Mann, 2014). Some common functions in data mining are association, discrimination, classification, clustering, and trend analysis. Clustering is unsupervised learning in the field of data mining, which deals with an enormous amount of data. It aims to assist users to determine and understand the natural structure of data sets and to extract the meaning of huge data sets (Kameshwaran & Malarvizhi, 2014; Kumar & Wasan, 2010; Yadav & Dhingra, 2016). In this light, clustering is the task of dividing objects which are similar to each other within the same cluster, whereas objects from distinct clusters are dissimilar (Jain & Dubes, 2011). Cluster methods are increasingly used in many areas, such as biology, astronomy, geography, pattern recognition, customer segmentation, and web mining (Kodinariya & Makwana, 2013). These applications use clusters to produce a suitable pattern from the data that may assist users and researchers to make wise decisions. In general, the clustering algorithms can be classified into hierarchical (Agglomerative & divisive clustering), partition (*k*-means, *k*-medoids, CLARA, CLARANS), density based, grid-based, and model based clustering methods (Han et al., 2012; Kaufman & Rousseeuw, 1990; Visalakshi & Suguna, 2009).

* Corresponding author.
E-mail address: mcabuhtto@seu.ac.lk (M. C. Alibuhtto)

doi: 10.5267/j.dsl.2019.8.002

The *k*-means algorithm is a very simple and fast commonly used unsupervised non-hierarchical clustering technique. This technique has been proven to obtain good clustering results in many applications. In recent years, many researchers have conducted various studies to determine the correct number of clusters using traditional and modified *k*-means algorithm (Kane & Nagar, 2012; Muca & Kutrolli, 2015), where the centroids are sometimes based on early guessing. However, very few studies have been performed to determine optimal number of clusters using *k*-means algorithm for high dimensional data set. Furthermore, in the common *k*-means clustering algorithm, ordinary steps encounter some drawbacks when the number of iterations of uncertainty can be processed to determine the optimal number of clusters, especially when using unmatched centroids (k). Selecting the appropriate cluster number (*k*) is essential for creating a meaningful and homogeneous cluster when using the *k*-means cluster algorithm for two-dimensional or multidimensional datasets. The selection of *k* is a major task to create meaningful and consistent clusters where subsequently, the k-means clustering algorithm is applied to high dimensional datasets. Mehar et al. (2013) introduced a novel k-means clustering algorithm with internal validation measures (sum of square errors) that can be used to find the suitable number of clusters (*k*). Alibuhtto and Mahat (2019) also proposed a new distance-based *k*-means algorithm to determine the ideal number of clusters for the multivariate numerical data set. It was found that while the proposed algorithm works well, but the study was limited to small sets of multivariate simulation data with only two clusters (such as *k=2* and *k=3*). Hence, this study aims to introduce a new algorithm to determine the number of optimal clusters using the *k*-means clustering algorithm based on the distance of high dimensional numerical data set.

2. Methodology

2.1 Data Simulation

In this study, the proposed *k*-means algorithm was tested by generating twelve sets of random normal multivariate numerical data for different sizes of the cluster (*k=2,3,5*) with *n* objects (*n*=10000, 20000), *p* number of variables (*p*=10, 20) where the variables are having a multivariate normal distribution with different mean vectors (μ_i), and covariance matrix. These multivariate normal data were generated using *mvrnorm* () function in R package in the combination of *k, n,* and *p* (Say Data1-Data12). Whereas, the proposed algorithm was tested by a generated six non-normal multivariate data sets for different sizes of cluster (*k=2,3,5*) with *n=1000* and *p=10* using *montel* () function in R (Say Data13-Data18).

2.2. K-means Algorithm

The *k*-means algorithm is an iterative algorithm that attempts to divide the data sets into *k* pre-defined non-overlapping sets of clusters. In this case, each data point belongs to one group. It tries to create the inter-cluster data points as similar as possible while at the same time, keeping the clusters as different as possible. It assigns data points to a cluster, so that the sum of the squared distance between the data points and the cluster's centroid is minimum.

The following steps can be used to perform *k*-means algorithm.

1. Randomly produce predefined value of *k* centroids
2. Allocate each object to the closest centroids
3. Recalculate the positions of the *k* centroids, when all objects have been assigned.
4. Repeat steps 2 and 3 until the sum of distances between the data objects and their corresponding centroid is minimized.

2.3. The Proposed Approach

Determining the optimal number of clusters in a data set is the foremost problem in the *k*-means cluster algorithm for high dimensional data set. In this regard, users are required to determine number of clusters to be generated. Therefore, this study proposes the use of *k*-means algorithm based on

Euclidean distance measures to identify the exact number of optimal number of clusters from the data. The proposed structure of the study is shown in Fig. 1.

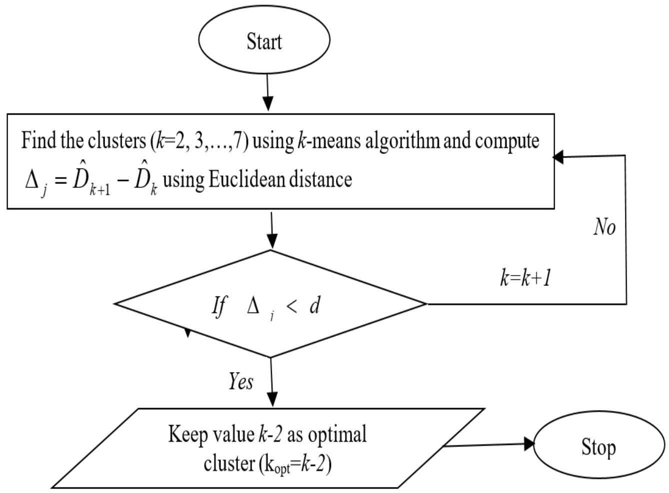

Fig. 1. Structure of proposed k-means clustering algorithm

The constant value (d) in Fig. 1 represents the test value, where that the objects are repeatedly clustered if the value Δ_j is greater than d ($j=k+1,k+2,..$). Whereas, Δ_j is the computed minimum distance between centres of k^{th} clusters ($k=2,3,…,7$). In this proposed algorithm, the Euclidean distance was chosen as a measurement of separation between objects due to its straightforward computation for numerical high dimensional data set. The following steps can be used to achieve the suitable number of clusters.

1. Set the minimal number of $k = 2$
2. Perform k-means clustering and compute Euclidean distance between centroids of each clusters
3. Increase the number of clusters as $k+1$, perform again k-means clustering and compute the distance between clusters.
4. Compare two consecutive distances at k and $k+1$
5. If the difference is acceptable, then the best optimal cluster is k-2. Otherwise, repeat Step 3.

2.4. Identify the test value (d)

The constant value (d) was determined using the scatter plot [difference between cluster centroids (Δ_j) vs cluster number (k)] through the points close to the peak point in different conditions. The value d was computed by obtaining the average of three points close to the peak point (succeeding and preceding points). For instance,

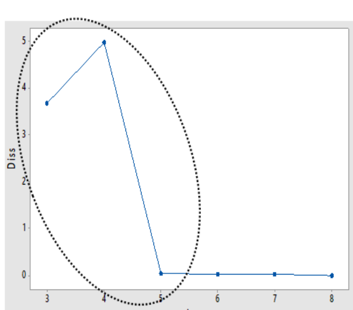

Fig. 2. Scatter plot for Δ_j vs k

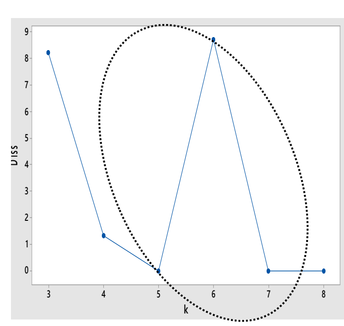

Fig. 3. Scatter plot for Δ_j vs k

In Fig. 2, the peak value can be seen when $k=4$. Not much fluctuations were observed afterwards. Therefore, the constant value d_1 was computed (taking average of 3 neighboring points close to the peak point) using formula 1. Likewise, as shown in Fig. 3, after the first point, the peak point is at $k=6$. Hence, the d_2 was calculated by using formula 2.

$$d_1 = \frac{(\Delta_3 + \Delta_4 + \Delta_5)}{3}, \tag{1}$$

$$d_2 = \frac{(\Delta_5 + \Delta_6 + \Delta_7)}{3}. \tag{2}$$

2.5. Cluster Validity Indices

Cluster validation measure is important for evaluating the quality of clusters (Maulik & Bandyopadhyay, 2002). Different quality measures have been used to assess the quality of the discovered clusters. In this study, Dunn and Calinksi-Harbaz indices were used to assess the cluster results, and they are briefly described in section *2.51* and *2.5.2*.

2.5.1. Dunn Index (DI)

This index is described as the ratio between the minimal intra cluster distances to maximal inter cluster distance. The Dunn index is as follows:

$$DI = \min_{1 \le i \le k}\left[\min_{i+1 \le j \le k}\left[\frac{dist(c_i, c_j)}{\max\limits_{1 \le l \le k} diam(c_l)}\right]\right], \tag{3}$$

where $dist(c_i, c_j) = \min\limits_{x_i \in c_i \, and \, x_j \in c_j} d(x_i, x_j)$ is the distance between clusters c_i and c_j ; $d(x_i, x_j)$ is the distance between data objects x_i and x_j ; $diam(c_l)$ is diameter of cluster c_l, as the maximum distance between two objects in the cluster. The maximum value of the Dunn index identifies that k is the optimal number of clusters.

2.5.2 Calinski-Harabasz Index (CH)

This index is commonly used to evaluate the cluster validity and is defined as the ratio of the between-cluster sum of squares (BCSS) and within-cluster sum of squares (WCSS) (Calinski & Harabasz, 1974). This index can be calculated by the following formula:

$$CH = \frac{(n-k)BCSS}{(k-1)WCSS}, \tag{4}$$

where n is the number of objects and k is the number of clusters. The maximum value of CH indicates that k is the optimal number of clusters.

3. Results and Discussions

The proposed algorithm was tested using twelve sets of normal multivariate simulated data (Data1-Data12) with two, three, and five clusters to determine the exact number of clusters. Fig. 4 to Fig. 6 present the scatter plot of differences between cluster centroids (Δ_j) against cluster number (k) for data sets with k=2, 3 and 5. The test value (d) was calculated from Fig. 4 to Fig. 6, as described in section *2.4*. The validity index (DI and CH), the difference between consecutive clusters centroids (Δ_j), test value (d) for each data set (Data1-Data4) are presented in Table 1. The maximum value of DI and CH was obtained when k=2, which confirms that the number of clusters of data sets is 2. In addition, the

Δ_j is less than at *k=4*. According to section *2.4* and Fig. 1, the optimal number of cluster for each data set (Data1-Data4) is 2. Similarly, Table 2, and Table 3 report the maximum values of DI and CH obtained for k=3 and k=5. Also, the Δ_j is less than at k=5 and 7 for data sets (Data5-Data8) with three clusters and data set (Data9-Data12) with five clusters respectively. These results indicate that the optimal number of clusters for each data set is 3 and 5, respectively. Therefore, the proposed algorithm is more appropriate for finding the correct number of clusters for high dimensional normal data.

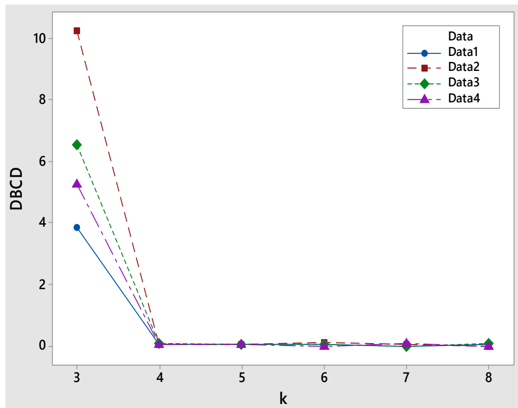

Fig. 4. Scatter plot for distance between cluster centroids (DBCD) vs k for Data1-Data4

Table 1
Clustering results for Data1-Data4 with 2 clusters

Data Set	*n*	*p*	*k*	Clusters of sizes	DI	CH	Δ_j	d
Data1	10000	10	2	10000,10000	0.784	124429.40	-	1.325
			3	10000,5026,4974	0.066	65104.39	3.869	
			4	3306,10000,3290,3404	0.060	44809.81	0.055	
			5	4892,3442,5108,3278,3280	0.053	35347.62	0.051	
Data2			2	10000,10000	2.628	642046.60	-	3.449
			3	4827,5173,10000	0.072	333190.30	10.259	
			4	3142,10000,3417,3441	0.069	227474.30	0.044	
			5	3265,3342,5040,3393,4960	0.062	177709.80	0.044	
Data3	25000	20	2	25000,25000	1.124	301828.70	-	2.231
			3	25000,12629,12371	0.143	155192.60	6.566	
			4	8520,25000,8214,8266	0.127	105574.90	0.084	
			5	8864,12357,12643,7568,8568	0.130	8040750	0.043	
Data4			2	25000,25000	0.922	203144.10	-	1.803
			3	12572,25000,12428	0.131	104749.70	5.269	
			4	8279,8382,8339,25000	0.128	71299.03	0.073	
			5	6226,25000,6158,6160,6456	0.128	54334.43	0.068	

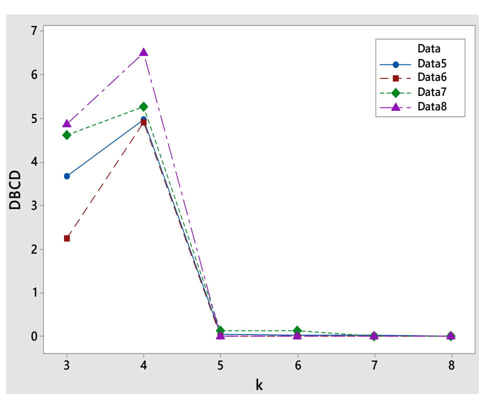

Fig. 5. *Scatter* plot for distance between cluster centroids (DBCD) vs k for Data5-Data8

Table 2
Clustering results for Data5-Data8 with 3 clusters

Data Set	*n*	*p*	*k*	Clusters of sizes	DI	CH	Δ_j	d
Data5	10000	10	2	20000,10000	0.799	94657.10	-	2.896
			3	10000,10000,10000	1.053	395378.80	3.672	
			4	10000,10000,5016,4984	0.073	270195.70	4.973	
			5	3314,10000,3278,3408,10000	0.037	206047.30	0.042	
Data6			2	10000,20000	0.406	61669.30	-	2.387
			3	10000,10000,10000	0.820	226310.90	2.245	
			4	10000.5184,10000,4816	0.068	155564.90	4.909	
			5	10000,3441,10000,3288,3271	0.056	119189.10	0.006	
Data7	25000	20	2	50000,25000	0.625	231326.50	-	3.339
			3	25000,25000,25000	0.886	530252.30	4.611	
			4	49993,8543,8086,8378	0.061	43290.43	5.269	
			5	6182,50000,6223,6296,6299	0.068	58575.99	0.138	
Data8			2	25000,50000	0.610	217738.80	-	3.786
			3	25000,25000,25000	1.035	616868.30	4.868	
			4	12403,25000,25000,12597	0.129	418971.20	6.491	
			5	12607,12537,12463,25000,12393	0.117	320078.10	0.000	

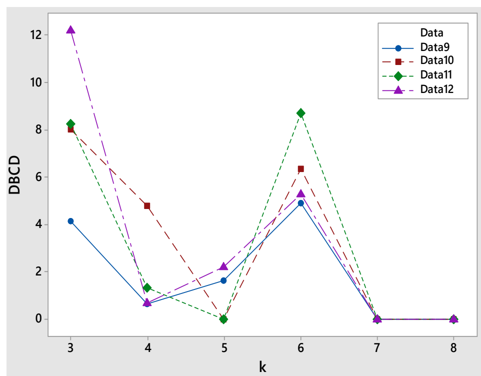

Fig. 6. Scatter plot for distance between cluster centroids (DBCD) vs k for Data9-Data12

Table 3
Clustering results for Data9-Data12 with 5 clusters

Data Set	*n*	*p*	*k*	Clusters of sizes	DI	CH	Δ_j	d
Data9	10000	10	4	20000,10000,10000,10000	0.421	190191.10	0.644	2.192
			5	10000,10000,10000,10000,10000	0.811	506145.40	1.645	
			6	5091,4809,5191,4909,10000,20000	0.024	112564.70	4.922	
			7	20000,5064,3426,3314,4954,10000,3260	0.025	96429.63	0.009	
Data10			4	10000,10000,10000,20000	0.699	346493.60	4.783	2.120
			5	10000,10000,10000,10000,10000	1.167	1332698.00	0.000	
			6	3261,10000,10000,3310,20000,3429	0.018	141983.00	6.360	
			7	20000,1968,2033,2018,20000,1964,2017	0.017	47530.06	0.000	
Data11	25000	20	4	25000,25000,50000,25000	0.528	356637.50	1.337	2.908
			5	25000,25000,25000,25000,25000	1.448	1555790.00	0.000	
			6	50000,12363,12376,25000,12624,12637	0.046	214903.10	8.719	
			7	25000,50000,8595,12616,12384,7937,8468	0.045	167883.60	0.004	
Data12			4	50000,25000,25000,25000	0.646	997992.80	0.679	2.496
			5	25000,25000,25000,25000,25000	1.086	2490765.00	2.196	
			6	12596,50000,12405,12404,12595,25000	0.058	602549.40	5.285	
			7	7798,50000,8378,8824,12404,25000,12596	0.051	386859.20	0.008	

The proposed *k*-means algorithm was also tested for generated non-normal multivariate data set with three different clusters *k=2, 3 and 5*. The values of the constant *d* for each data set were computed according to the graph as shown in Fig. 7 to Fig. 9. The results of the proposed algorithm and validation indices for non-normal datasets (Data13 – Data18) are presented in Table 4.

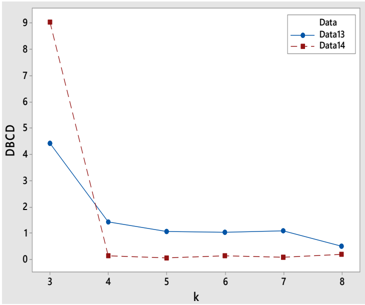

Fig. 7. Scatter plot for distance between cluster centroids (DBCD) vs k for Data13-Data14

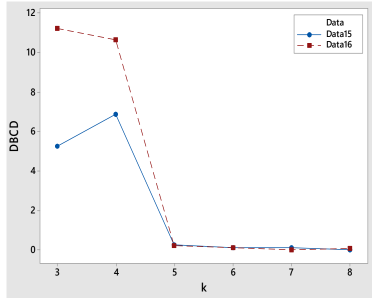

Fig. 8. Scatter plot for distance between cluster centroids (DBCD) vs k for Data15-Data16

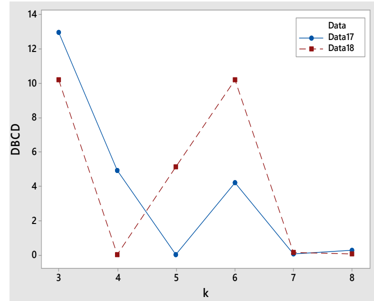

Fig. 9. Scatter plot for distance between cluster centroids (DBCD) vs k for Data17-Data18

Table 4
Clustering results for Data13-Data18 with 2, 3 and 5 clusters

Data Set	*n*	*Clu*	*p*	*k*	Clusters of sizes	DI	CH	Δ_j	d
Data13	10000	2	10	2	9970,10030	0.091	20703.76	-	2.315
				3	3075,6850,10075	0.033	10635.34	4.447	
				4	9868,1982,6191,1959	0.041	8035.90	1.429	
				5	5014,1760,1620,10015,1591	0.039	6237.28	1.069	
Data14	20000			2	20000,20000	0.292	56954.23	-	3.093
				3	20000,12481,7519	0.057	24656.43	9.042	
				4	20000,4132,12229,3639	0.062	13340.22	0.160	
				5	3528,4845,4353,20000,7274	0.064	10261.14	0.076	
Data15	10000	3	20	2	19950,10050	0.074	46639.67	-	4.118
				3	9823,10121,10056	0.130	78440.32	5.243	
				4	3136,6892,9955,10017	0.032	51061.52	6.860	
				5	6404,6388,10001,3683,3524	0.034	38532.40	0.251	
Data16	20000			2	20000,40000	0.512	152845.30	-	7.357
				3	20000,20000,20000	0.611	209643.10	11.202	
				4	13166,6834,20000,20000	0.059	134352.70	10.642	
				5	20000,5764,10751,3485,20000	0.072	102443.30	0.227	
Data17	10000		10	4	10000,10011,19418,10571	0.051	136576.40	0.014	1.444
				5	10052,10005,10015,9910,10018	0.092	157120.30	4.230	
				6	10000,9908,7219,2822,10015,10036	0.039	124235.60	0.088	
				7	2949,9908,7913,10054,2134,7042,10000	0.042	102431.50	0.263	
Data18	20000		20	4	20000,20000,40000,20000	0.212	129872.40	5.144	5.163
				5	20000,20000,20000,20000,20000	0.295	331244.70	10.210	
				6	3062,3150,3232,40000,39998,10558	0.041	61543.32	0.135	
				7	14320,20000,5680,11385,40000,5035,3580	0.050	104325.90	0.088	

According to the Table 4, the maximum values of the DI and CH obtained when *k=2* for Data13 and Data14, *k*=3 for Data15 and Data16, and *k*=5 for Data17 and Data18. This result confirmed that the number of clusters of non- normal multivariate datasets is 2, 3, and 5 respectively. Furthermore, the minimum distances between cluster centroids (Δ_j) of datasets Data13 and Data14 is less than *d* for *k=2*, whereas Data15 and Data16 for *k*=3, and Data17 and Data18 for *k*=5 (section *2.3* & Fig. 1). This result indicate that the optimal number of clusters of non-normal multivariate data set is two, three and five. Hence, the proposed new distanced based k-means algorithm is the best technique to find the exact number of clusters for high dimensional data sets.

4. Conclusion

This study has proposed a distance-based *k*-means clustering algorithm to determine the suitable number of clusters for high dimensional data set. The proposed algorithm hs examined eighteen sets of normal and non-normal high dimensional simulation data and results revealed that the proposed algorithm was more accurate for finding the correct number of optimal clusters without using any validation indices. In addition, this paper is useful for finding the exact number of clusters for big data, because the validation index is insufficient to assess the quality of clusters for big data. However, the proposed algorithm can be improved to be used on categorical and mixed data.

Acknowledgements

This research paper is a part of first author's PhD studies under the supervision of the second author.

References

Alibuhtto, M.C., & Mahat, N.I. (2019). New approach for finding number of clusters using distance based *k*-means algorithm, *International Journal of Engineering, Science and Mathematics, 8*(4), 111-122. https://ijesm.co.in/uploads/68/6159_pdf.pdf

Calinski, T., & Harabasz, J.(1974). A dendrite method for cluster analysis, *Communications in Statistics*, *3*(1),1–27. https://www.researchgate.net/publication/233096619_A_Dendrite_Method_for_Cluster_Analysis

Dunn, J.C. (1974). Well separated clusters and optimal fuzzy partitions, *Journal of Cybernetics,* 4, 95-104.

Han, J., Kamber, M., & Pei, J. (2012). *Data mining: Concepts and Techniques,* San Francisco, CA, Litd: Morgan Kaufmann (Vol. 5).

Jain, A.K., & Dubes, R.C. (2011). *Algorithms for Clustering Data*. Pretice Hall, Englewood Cliffs, New Jersey.

Kameshwaran, K., & Malarvizhi, K. (2014). Survey on clustering techniques in data mining, *International Journal of Computer Science and Information Technologies*, *5*(2), 2272–2276. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.661.4512&rep=rep1&type=pdfKane, A., & Nagar, J. (2012). Determining the number of clusters for a *k*-means clustering algorithm. *Indian Journal of Computer Science and Engineering (IJCSE)*, *3*(5), 670–672.

Kaufman, L., & Rousseeuw, P. J. (1990). *Finding groups in data: An Introduction to Cluster Analysis.* Wiley Series in Probability and Statistics. Eepe.Ethz.Ch.

Kodinariya, T. M., & Makwana, P. R. (2013). Review on determining number of cluster in k-means clustering, *International Journal of Advance Research in Computer Science and Management Studies*, *1*(6), 90–95.

Kumar, P., & Wasan, S. K. (2010). Comparative analysis of k-mean based algorithms, *International Journal of Computer Science and Network Security, 10*(4), 314–318.

Maulik, U., & Bandyopadhyay, S. (2002). Performance evaluation of some clustering algorithms and validity indices, *IEEE Transactions on Pattern Analysis and Machine Intelligence, 24*(12), 1650-1654.

Mehar, A. M., Matawie, K., & Maeder, A. (2013). Determining an optimal value of k in k-means clustering, In *Proceedings of the International Conference on Bioinformatics and Biomedicine: IEEE BIBM*, 51–55.

Muca, M., & Kutrolli, G. (2015). A proposed algorithm for determining the optimal number of clusters. *European Scientific Journal, 11*(36), 112–120. https://eujournal.org/index.php/esj/article/download/6756/6482Ramageri, B.M. (2010). Data Mining Techniques and Applications, *Indian Journal of Computer Science and Engineering*, *1*(4), 301-305.

Thakur, B., & Mann, M. (2014). Data mining for big data: A review, *International Journal of Advanced Research in Computer Science and Software Engineering*, *4*(5), 469-473.

Visalakshi, N. K., & Suguna, J. (2009). *K*-means clustering using max-min distance measure, *Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS)*, 1–6.

Yadav, A., & Dhingra, S. (2016). A review on *k*-means clustering technique, *International Journal of Latest Research in Science and Technology, 5*(4), 13–16.