Data Clustering with Cluster Size Constraints Using a Modified *k*-means Algorithm

Nuwan Ganganath†, Chi-Tsun Cheng, and Chi K. Tse
Department of Electronic and Information Engineering
The Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong
†Email: nuwan.marasinghearachchige@connect.polyu.hk

***Abstract*—Data clustering is a frequently used technique in finance, computer science, and engineering. In most of the applications, cluster sizes are either constrained to particular values or available as prior knowledge. Unfortunately, traditional clustering methods cannot impose constrains on cluster sizes. In this paper, we propose some vital modifications to the standard *k*-means algorithm such that it can incorporate size constraints for each cluster separately. The modified *k*-means algorithm can be used to obtain clusters in preferred sizes. A potential application would be obtaining clusters with equal cluster size. Moreover, the modified algorithm makes use of prior knowledge of the given data set for selectively initializing the cluster centroids which helps escaping from local minima. Simulation results on multidimensional data demonstrate that the *k*-means algorithm with the proposed modifications can fulfill cluster size constraints and lead to more accurate and robust results.**

Keywords*—*k-means, data mining, data clustering, size constraints, constrained clustering

I. Introduction

Data clustering can be identified as a learning method which groups a set of data in such a way that the data in the same group show higher similarity in certain properties when considering with data in other groups [1]. These groups are commonly referred to as clusters in data mining [2]. Clustering plays an important role in many applications such as big data clustering [3], document clustering [4], image segmentation [5], and sensor clustering in wireless sensor networks [6], [7]. Over the previous several decades, numerous data clustering algorithms have been proposed by the researchers. One may refer to [2] for a comprehensive review on data clustering algorithms.

k-means is one of the well known clustering method due to its simplicity. It is based on the idea of *centroids* which are used to define clusters in this work. It partitions a given set of data into k clusters using the distance from each data point to k different centroids (or means). The term "*k*-means" was first used by by James MacQueen in [8]. However, the basic idea behind this algorithm goes back to Polish mathematician Hugo Steinhaus [9]. Even though the *k*-means is efficient, it may converge to local minima producing counterintuitive results, mainly due to the randomness in its initialization. Also, it has very loose control on cluster sizes.

In many real world applications, sizes of the clusters are available either as prior knowledge or as constraints. The results of existing clustering methods may be further enhanced by using additional information harvested from the data set [10], whereas these additional information must be incorporated when they come as constraints. In this paper, we modify a standard *k*-means algorithm such that it can constrain each individual cluster to a predefined size. The modified *k*-means algorithm can partition a given set of data into clusters with same size or different sizes according to predefined requirements or prior knowledge about the data set. Furthermore, the proposed algorithm has relatively lesser chance of getting trapped in local minima as the centroids are selectively initialized with data points from each cluster using the prior knowledge. Therefore, the modified *k*-means clustering algorithm can produce more reasonable results compared to the standard *k*-means algorithm.

The rest of the paper is organized as follows. Section II presents the problem formulation. Section III briefly reviews the standard *k*-means algorithm. Modifications to the *k*-means algorithm are introduced in Section IV such that it can incorporate the size constraints of each cluster. Simulation results are presented and performances of the proposed algorithm are analyzed in Section V. Concluding remarks are given in Section VI.

II. Problem Formulation

Let $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$ be a given data set of n objects where $x_i \in \mathbb{R}^m$. In a data clustering problem without any cluster size constraints, the objective of a clustering algorithm is to find k $(1 \leq k \leq n)$ clusters, $\mathbf{c} = \{c_1, c_2, \ldots, c_k\}$, such that the similarity among objects in each cluster is maximized. Similarity measuring criteria can be different from one clustering algorithm to another. Here, the total size of the data set $|\mathbf{c}| = \sum_{\forall j} |c_j|$, where $|c_j|$ denotes the size of a cluster c_j and $1 \leq j \leq k$. Thus, $|\mathbf{c}| = |\mathbf{x}|$. In the data clustering with cluster size constraints, the maximum cluster size ζ_j is available for each cluster c_j. Therefore, a size constrained data clustering algorithm has to satisfy an extra constraint $|c_j| \leq \zeta_j$, such that $\sum_{j=1}^{k} \zeta_j \geq |\mathbf{x}|$.

III. The *k*-means Algorithm

Similar to many other clustering algorithms, the *k*-means algorithm utilizes an iterative procedure. In each iteration, it tries to minimize the within-cluster sum of squares, i.e.

$$\sum_{j=1}^{k} \sum_{\forall x_i \in c_j} \|x_i - \mu_j\|^2, \tag{1}$$

in order to maximize the similarity among objects in each cluster. Here, $\mu_j \in \mathbb{R}^m$ is the centroid of a cluster c_j and $\| \cdot \|^2$ is the squared Euclidean norm. The standard k-means algorithm can be described in three steps as given below.

Initialization step:

In the initialization step, the k-means algorithm initializes μ_j of cluster c_j, such that

$$\mu_j^{(1)} = \{x_p : x_p \in \mathbf{x}, \mu_i^{(1)} \neq x_p, 1 \leq i \leq k, i \neq j\}. \quad (2)$$

Different variations of the k-means algorithm may use different methods. However, the most common method is to use distinct random data points from **x** to initialize each centroid. After the initialization of the centroids, the k-means algorithm proceeds by alternating between an assignment step and an update step.

Assignment step:

In this step, each data point is assigned to the cluster whose centroid yields the least within-cluster sum of squares.

$$c_j^{(t)} = \{x_p : \|x_p - \mu_j^{(t)}\|^2 \leq \|x_p - \mu_i^{(t)}\|^2, \forall i, 1 \leq i \leq k\}. \quad (3)$$

Note that x_p is assigned to only one cluster $c_j^{(t)}$ in time step t, i.e. $x_p \notin c_i^{(t)}$ where $i \neq j$. In another time step, x_p may be assigned to another cluster that minimizes the within-cluster sum of squares. Here, a data point cannot belong to more than one cluster in a single time step.

Update step:

In the update step, the k-means algorithm calculates the centroids for the next iteration according to the data assigned to each cluster in the assignment step of the current time step.

$$\mu_j^{(t+1)} = \frac{1}{|c_j^{(t)}|} \sum_{\forall x_i \in c_j^{(t)}} x_i. \quad (4)$$

For a given threshold $\xi \geq 0$, if $|\mu_j^{(t+1)} - \mu_j^{(t)}| \leq \xi$, $\forall j$, $1 \leq j \leq k$, the algorithm terminates its iterative process as it has already converged to a minimum. Otherwise, it iterates back to the assignment step and sets $t = t + 1$.

IV. A Modified k-means Algorithm

A modified k-means algorithm is proposed here for data clustering with cluster size constraints. We change the initialization and assignment steps of the standard k-means algorithm which is described in Section III to fulfill the size constraints of each cluster.

Initialization step:

In the initialization step, instead of initializing centroids with random data points from the given data set, we use the prior knowledge about the data set to assign initial centroids such that

$$\mu_j^{(1)} = \{x_p : x_p \in c_j, \forall j, 1 \leq j \leq k\}. \quad (5)$$

Here, we assume that prior knowledge of at least few data points per each cluster is available to the user which is practical in typical real world scenarios. The modified k-means algorithm requires the knowledge of only a single data point from each cluster. Such a selective initialization can greatly reduce the possibility of converging to local minima producing counterintuitive results.

TABLE I. Simulation Parameters.

Simulation	Dimension (m)	Number of clusters (k)	Number of data points (n)	ξ
I	2	5	2000	0
II	3	4	1100	0

Assignment step:

In the assignment step, each data point is assigned to the cluster whose centroid yields the least within-cluster sum of squares only if $|c_j^{(t)}| < \zeta_j$ where $\sum_{j=1}^{k} \zeta_j \geq |\mathbf{x}|$. Therefore,

$$c_j^{(t)} = \{x_p : \|x_p - \mu_j^{(t)}\|^2 \leq \|x_p - \mu_i^{(t)}\|^2, |c_j^{(t)}| < \zeta_j, 1 \leq i \leq k\}. \quad (6)$$

While implementing, this can be easily achieved by sorting the values of $\|x_p - \mu_i^{(t)}\|^2$ in ascending order for all i $(1 \leq i \leq k)$ and iterating through the sorted array till it finds a cluster which satisfies the size constraints $|c_j^{(t)}| < \zeta_j$. For an example, one may use merge sort algorithm whose worst case performance is $O(k \log(k))$ [11], for sorting $\|x_p - \mu_i^{(t)}\|^2$ values in ascending order. Note that as $k \ll n$ in many practical applications, thus it does not have considerable impact on the runtime of the algorithm. Similar to the standard algorithm, each data point is assigned to only one cluster in a single time step.

In this modified version of the algorithm, we keep the update step and the termination criteria unchanged from the standard procedure.

V. Simulations and Performance Analysis

We evaluate and analyze the performances of the modified k-means algorithm against the standard k-means algorithm using computer simulations. Simulation settings and simulation results are presented in this section.

A. Simulation Settings

Two situations were set up using the two different data sets and simulation parameters are shown in TABLE I. In the first simulation, a two dimensional data set with 2000 data points was selected for clustering which consists of five clusters. In the second simulation, a three dimensional data set with 1100 data points was selected for clustering which consists of four clusters. Both the standard k-means algorithm described in Section III and modified k-means algorithm described in Section IV were implemented using Matlab. In the termination conditions of both algorithms, ξ was set to 0 which is the minimum possible value it can take.

B. Simulation Results

The standard k-means algorithm which is described in Section III, the modified k-means algorithm with cluster size constrains but with the random initialization, and the modified k-means algorithm with cluster size constrains and the selective initialization which is described in Section IV are tested with

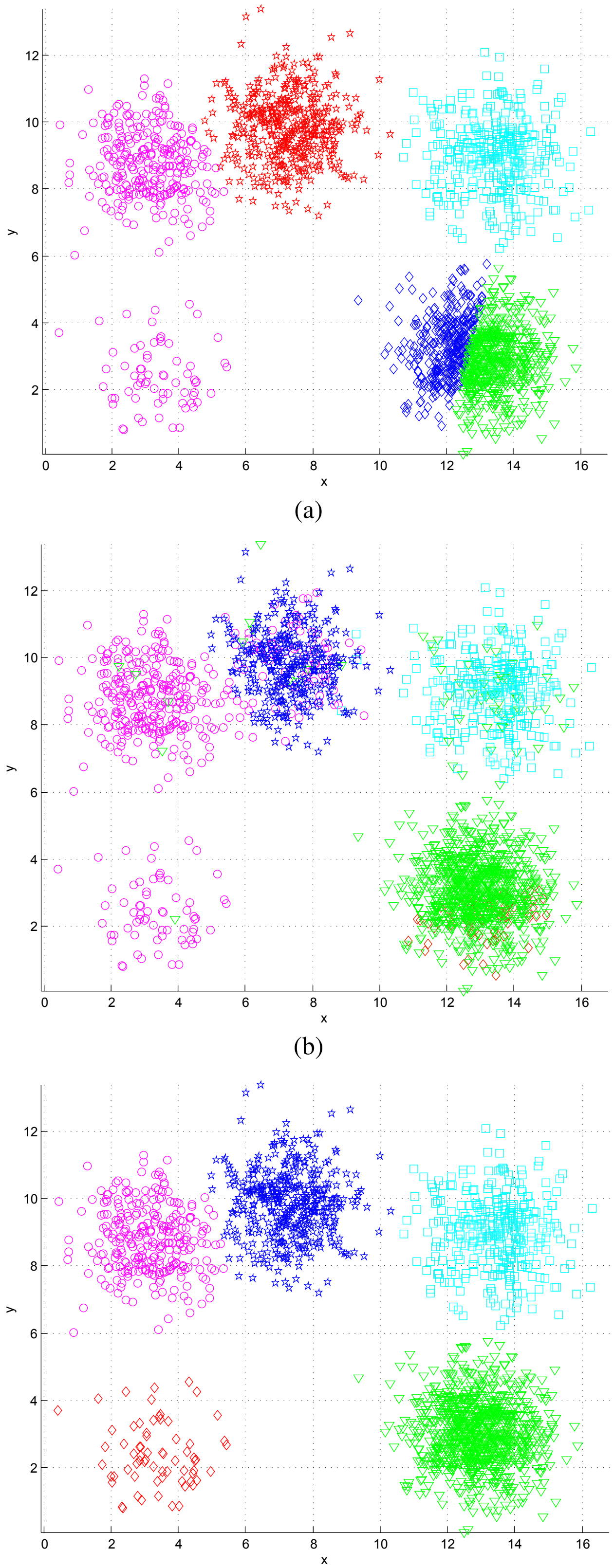
TABLE II. SIMULATION RESULTS.

Sim.	Cluster index (j)	Cluster size constraint (ζ_j)	Final cluster size	
			k-means	modified k-means
I	1	280	343	280
	2	431	435	431
	3	67	347	67
	4	891	544	891
	5	331	331	331
II	1	100	200	100
	2	100	102	100
	3	300	198	300
	4	600	600	600

the parameters given in TABLE I. Simulation results are summarized in TABLE II. Spatial distribution of the data used in Simulation I and Simulation II are graphically illustrated in Fig. 1 and Fig. 2, respectively. Data points which are represented in same color and symbol belong to the same cluster.

According to Fig. 1, all the algorithms under test have detected five clusters in the first simulation. The modified k-means algorithm with random initialization results in poor clustering even though the cluster sizes are constrained (see Fig. 1(b)). It is obvious that the modified k-means algorithm with selective initialization has achieved more intuitive results in this experiment against the standard algorithm (see Fig. 1(c)). The fourth cluster which consists of 891 data points (triangles in Fig. 1(c)) has partitioned into two clusters by the standard k-means algorithm (triangles and rhombuses in Fig. 1(a)). The main reason behind this counterintuitive result is the random initialization of the cluster centroids. In the initialization step, two centroids are initialized with random data points from this cluster, which is comparatively larger and has considerable spatial gap with other clusters. After several iterations, as we can observe in the results, it has trapped in a local minimum. Since the number of clusters are fixed in both of these algorithms, one false detection of clusters may lead to another false detection. This can be observed in Fig. 1(a) where clusters 1 and 3 are merged into a single cluster (represented by using circles). According to the results given in TABLE II, the modified k-means algorithm (with random or selective initialization) has detected all the clusters correctly according to the size constraints while the standard algorithm has considerably deviated from these cluster constraints. However, it is not a surprise as the standard algorithm does not incorporate any size constraints on the clusters.

In the second experiment, the algorithms under test were tested using three dimensional data with four clusters. Similar to the first experiment, the modified k-means algorithm with random initialization results in poor clustering. However, with selective initialization, the modified algorithm has achieved more intuitive results in this experiment against the standard algorithm. The third cluster which consists of 300 data points (squares in Fig. 2(b)) partitioned to two different clusters by the standard k-means algorithm (squares and rhombuses in Fig. 2(a)). It is the same reason that the random initialization of centroids has caused this counterintuitive result. In the initialization step, two centroids are initialized with random data points from this cluster, which is comparatively larger and has considerable spatial gap with other clusters. After several iterations, it has trapped in a local minimum where it is partitioned into two different clusters while two small clusters are merged together to achieve the fixed number of clusters. This can be observed in Fig. 2(a) where clusters 1

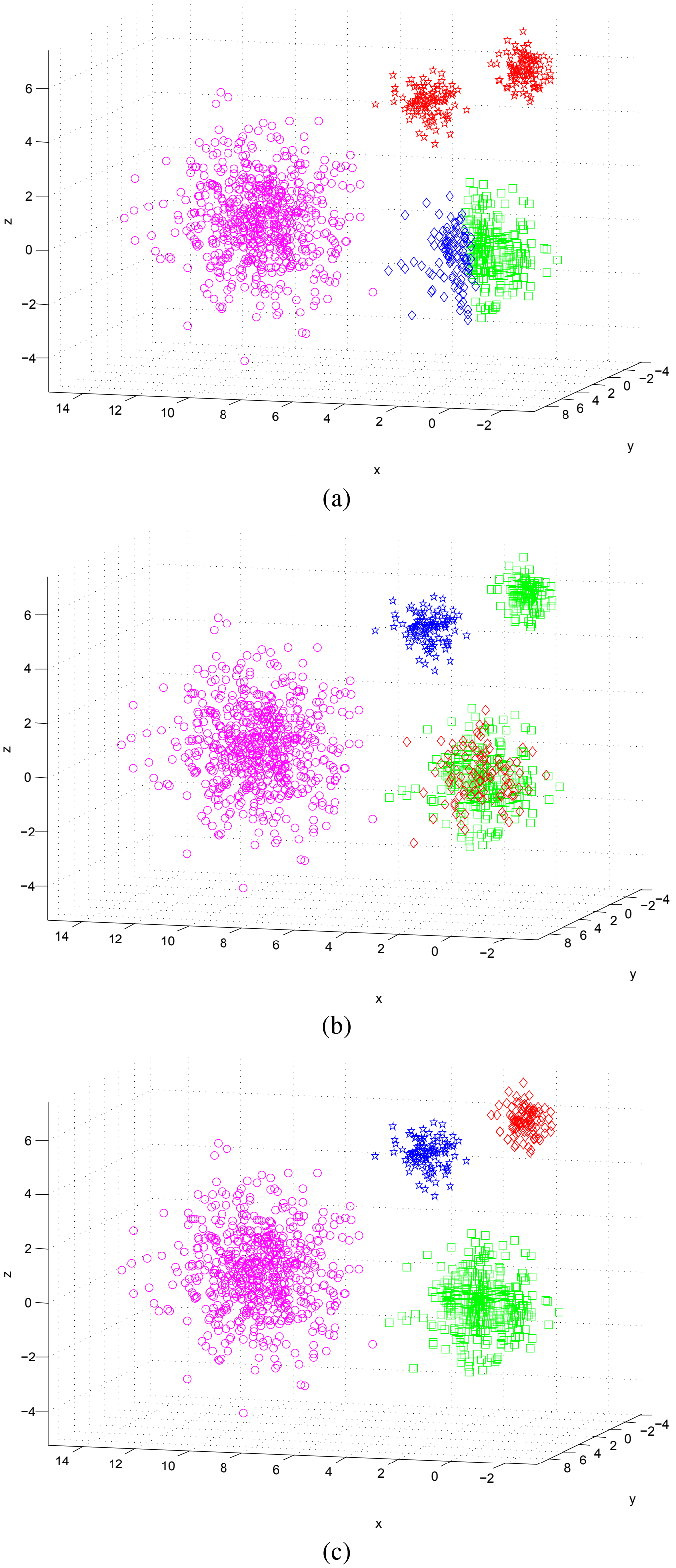


(a)

(b)

(c)

Fig. 1. Results of Simulation I: (a) using a standard k-means algorithm, (b) using the modified k-means algorithm with random initialization, and (c) using the modified k-means algorithm with selective initialization.



(a)

(b)

(c)

Fig. 2. Results of Simulation II: (a) using a standard *k*-means algorithm, (b) using the modified *k*-means algorithm with random initialization, and (c) using the modified *k*-means algorithm with selective initialization.

and 3 are merged into a single cluster (represented by using pentagrams). According to the results given in TABLE II, the modified *k*-means algorithm has detected all the clusters correctly according to the size constraints in the second simulation while the standard algorithm has considerably deviated from these cluster constraints.

One should note that Fig. 1(a) and Fig. 2(a) represent only two possible outcomes of the standard *k*-means algorithm. Depending upon the initial data points selected for initial centroids, results of different experiments with same data set can be worse or better, even similar to the results of the modified algorithm. However, in the case of modified algorithm with selective initialization, results are more stable because it uses prior knowledge for initialization of centroids. Moreover, it incorporates fixed cluster sizes while the cluster sizes are arbitrary in the original one.

VI. Conclusions

Traditional data clustering methods cannot fulfill the size constraints on clusters. In this paper, we introduce a robust algorithm for data clustering with constrained cluster sizes. The proposed algorithm is developed based on the the standard *k*-means algorithm. We modified the standard algorithm such that it can incorporate cluster size constraints. In the initialization step of the modified algorithm, it uses the prior knowledge to assign data points as the the initial centroids of the clusters, unlike random data point assignment in a standard *k*-means algorithm. In the assignment step, it assigns a new data point to the cluster whose centroid yields the least within-cluster sum of squares only if the current cluster size has not violated its size constraint. Otherwise, it go for the next best option till it finds a cluster which has not yet met its size constraint. Simulation results verify the superior performance of the modified algorithm over the standard *k*-means algorithm.

References

[1] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining, (First Edition)*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2005.

[2] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM computing surveys (CSUR)*, vol. 31, no. 3, pp. 264–323, 1999.

[3] H. Tong and U. Kang, "Big data clustering," *Data Clustering: Algorithms and Applications*, p. 259, 2013.

[4] M. Steinbach, G. Karypis, V. Kumar *et al.*, "A comparison of document clustering techniques," in *KDD workshop on text mining*, vol. 400, no. 1. Boston, 2000, pp. 525–526.

[5] D. L. Pham, "Spatial models for fuzzy clustering," *Computer vision and image understanding*, vol. 84, no. 2, pp. 285–297, 2001.

[6] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Computer communications*, vol. 30, no. 14, pp. 2826–2841, 2007.

[7] C.-T. Cheng, C. K. Tse, and F. C. Lau, "A Clustering Algorithm for Wireless Sensor Networks Based on Social Insect Colonies," *Sensors Journal, IEEE*, vol. 11, no. 3, pp. 711–721, 2011.

[8] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 281-297. California, USA, 1967, p. 14.

[9] H. Steinhaus, "Sur la division des corps matériels en parties," *Bull. Acad. Polon. Sci. Cl. III.*, vol. 4, pp. 801–804, 1957, (in French).

[10] S. Zhu, D. Wang, and T. Li, "Data clustering with size constraints," *Knowledge-Based Systems*, vol. 23, no. 8, pp. 883–889, 2010.

[11] D. Knuth, "Section 5.2. 4: Sorting by merging," *The Art of Computer Programming*, vol. 3, pp. 158–168, 1998.