Size-Constrained Clustering Using an Initial Points Selection Method

Kai Lei, Sibo Wang, Weiwei Song, and Qilin Li

Shenzhen Key Lab for Cloud Computing Technology & Applications (SPCCTA),
School of Electronics and Computer Engineering, Peking University, Shenzhen, P.R. China
leik@pkusz.edu.cn, {wangsibo,songweiwei}@sz.pku.edu.cn,
zirin.lee@gmail.com

Abstract. Size-Constrained clustering tries to solve the problem that how to classify dataset into groups based on each document's similarity with additional requirement which each group size is within a fixed range. By far, adding constraints to assignment step in K-Means clustering is a main approach. But the performance of the algorithm also depends highly on the initial cluster centers like standard K-Means. We propose an initial points selection method by recursively discovering the point with large density around it. Root Mean Square Error and convergence speed (iteration times) are the two most important evaluation standards for clustering using an iterative procedure. Our experiments are conducted on about ten thousand research proposals of National Natural Science Foundation of China and the results show that our method can reduce the iteration times by over 50% and get smaller Root Mean Square Error. The method is scalable and can be coupled with a scalable size-constrained clustering algorithm to address the large-scale clustering problem in data mining.

Keywords: size-constrained clustering, initial cluster centers, density around point.

1 Introduction

Clustering algorithm is often viewed as an unsupervised method for data analysis and it has been applied to many fields, such as data mining, statistical data analysis and knowledge discovery. K-Means clustering [1] has become a very famous method for clustering.

However, one drawback to the K-Means algorithm is that the algorithm often converges with one or more clusters which either are empty or summarize very few data points. In some cases, there are constraints about the cluster size requiring that the size of each cluster must be in a range. The solution to this problem was first introduced in the Bradley's paper [2]. In their work, they proposed adding constraints to the underlying clustering optimization problem requiring that each cluster has at least a minimum number of points based on K-Means clustering. They transformed the cluster assignment step into the Minimum Cost Flow (MCF) problem [4] and solved it by linear network optimization [3]. Jianmin Zhao [5] proposed two constrained

M. Wang (Ed.): KSEM 2013, LNAI 8041, pp. 195–205, 2013.

K-Means algorithms: Linear Programming Algorithm (LPA) and Genetic Constrained K-Means Algorithm (GCKA) in his Ph.D. Thesis. Linear Programming Algorithm modified the K-Means algorithm into a linear programming problem with constraints requiring that each cluster has m or more subjects. The most significant difference between Bradley's constrained K-Means algorithm and his LPA is that he ran the algorithm with a large number of random sets of initial points and chose the one with minimal root mean squared error (RMSE) as their final solution. Shunzhi Zhu also proposed a heuristic algorithm to transform size constrained clustering problems into integer linear programming problems in their work [6].

It is known that the iterative algorithms such as K-Means are especially sensitive to initial starting condition. As size-constrained clustering is commonly based on K-Means clustering, it also needs initial centers. Thus, the selection of initial centers has significant impact on the final result and it is also an important factor to improve the clustering solutions.

From the literature and the following experiment, we find that if initial selected centers are close to the final cluster centers, it will reduce the iteration times and get better global minimum for size-constrained clustering. While bad initial centers which with frequent change may lead to bad solution.

In this paper, we propose a method of choosing k points from dataset as the initial centers for size-constrained clustering with k cluster. This method can get the initial points nearer to the optimal result centers in the starting stage of clustering. We can decrease the final RMSE and get less iteration times to reduce the clustering time.

The remaining portion of the paper is organized as follows. In the Section 2, we provide some related work about initialization methods for K-Means clustering and size-constrained clustering. In Section 3, we discuss size-constrained clustering algorithm and our proposed method for initial points selection. Section 4 presents experiment results and discussion of the proposed method in comparison with the other two methods for initial points selection which are widely used on real datasets. Finally, Section 5 concludes the paper.

2 Related Work

2.1 Initialization Methods for K-Means Clustering

In the past, several methods were proposed to solve the cluster initialization for K-Means algorithm. A recursive method for initializing the means by running k clustering problems is discussed by Duda and Hart [7]. A variation of this method takes the entire data into account and then randomly perturbs it k times. For the initial cluster center, Jain and Dubes [8] proposed a method that selects initial values randomly with several times and selected the average of these final cluster centers at the starting stage of K-Means clustering.

The refinement algorithm, proposed by Bradley and Fayyad, builds a set of small random sub-samples of the data and clusters data in each sub-sample by K-Means [9]. All centroids of all sub-samples are then clustered together by K-Means using the k-centroids of each sub-sample as initial centers. The centers of the final clusters

giving minimum clustering error are to be used as the initial centers for clustering the original set of data using K-Means algorithm.

Deelers and Auwatanamongkol [10] proposed an algorithm to compute initial cluster centers for K-Means algorithm. They partitioned the data set in a cell using a cutting plane that divides cell in two smaller ones. The plane is perpendicular to the data axis with the highest variance and is designed to reduce the sum squared errors of the two cells as much as possible, keeping the two cells far apart as possible. Also they partitioned the cells once at a time until the number of cells equals to the predefined number of clusters k. In their method the centers of the k cells become the initial cluster centers for K-Means algorithm.

Khan and Ahmad [11] proposed Cluster Center Initialization Algorithm (CCIA) to solve cluster initialization problem. CCIA is based on two observations, with similar patterns to each other. It begins with calculating mean and standard deviation for data attributes, and then separates the data with normal curve into certain partitions. CCIA uses K-Means and density based on multi scale data condensation to observe the similarity of data patterns before finding out the final initial clusters. The experiment results of the CCIA performed the effectiveness and robustness to solve the several clustering problems.

D. Steinley, J. Michael and Brusco indicate several options for initializing the algorithm, compare the procedures, and make several recommendations [12]. J.A. Lozano, J.M. Pena, P. Larranaga compare empirically four initialization methods for the K-Means algorithm: random, Forgy, MacQueen and Kaufman [13]. The results of their experiments illustrate that the random and the Kaufman initialization methods outperform the rest of the compared methods, which make the K-Means more effective and more independent on initial clustering and on instance order.

2.2 The difference between Initialization for K-Means and Size-Constrained Clustering

To the best of our knowledge, there is limited work on initial selection method for size-constrained clustering. In Jianmin Zhao's Thesis [5] he adopted the method of selecting different random points and ran many times to find the best solution. But this will increase the total time to get the final result, and it hasn't gotten the best result yet.

At first thought we may apply the initialization method for K-Means to size-constrained clustering, but there are some differences between K-Means clustering and size-constrained clustering. In K-Means clustering we mainly consider assigning the similar points into the same group without noticing the size constraint of each group, which result in local optimal. While in size-constrained clustering, we need to consider the problem of cluster size in order to get a global optimal result. Sometimes, we need to take two similar points as different initial centers in size-constrained clustering, which will not be selected in K-Means clustering. Therefore we propose our method for initialization in size-constrained clustering.

3 Initial Points Selection Method

3.1 Size-Constrained Clustering

Problem description: Given a dataset $D = \{X^i\}_{i=1}^m$ of m points in R^n and cluster size constraint range [min, max], find cluster centers $C^1, C^2, \ldots C^k$ in R^n and the assgin array $T_{i,j}$ ($T_{i,j}$=1 means that X^i is assigned to cluster C^j) to minimize the sum of squared distance between every X^i and its assigned center.

specifically:

$$\underset{C,T}{minimize} \sum_{i=1}^{m}\sum_{j=1}^{k} T_{i,j}(\| X^i - C^j \|^2)$$

$$\sum_{i=1}^{m} T_{i,j} \in [\min, \max];\ j = 1,2, \ldots k$$

subject to:

$$\sum_{j=1}^{k} T_{i,j} = 1; i = 1,2, \ldots m$$

$$T_{i,j} \geq 0, i = 1,2, \ldots m; j = 1,2, \ldots k$$

Like K-Means clustering, size-constrained clustering problem is also solved in two step recursively, cluster assignment and cluster update. It adds constraints in the step of cluster assignment which requires each cluster size in a given range. The cluster assignment problem can be solved by LP(linear programming) or Simplex Network [2].

The problem can be solved iteratively and is described as follows:

Suppose cluster centers $C^{1,t}$, $C^{2,t}$, ... $C^{k,t}$ at iteration t, compute $C^{1,t+1}$, $C^{2,t+1}$, ... $C^{k,t+1}$ at iteration t + 1 in the following 2 steps:

Cluster Assignment: For each data record $X^i \in D$, assign X^i to cluster j such that

specifically:

$$\underset{T}{minimize} \sum_{i=1}^{m}\sum_{j=1}^{k} T_{i,j}(\| X^i - C^{j,t} \|^2)$$

$$\sum_{i=1}^{m} T_{i,j} \in [\min, \max];\ j = 1,2, \ldots k$$

subject to:

$$\sum_{j=1}^{k} T_{i,j} = 1; i = 1,2, \ldots m$$

$$T_{i,j} \geq 0, i = 1,2, \ldots m; j = 1,2, \ldots k$$

Cluster Update: Compute $C^{j,t+1}$ as the mean of all points assigned to cluster j.

$$C^{j,t+1} = \begin{cases} \frac{\sum_{i=1}^{m} T_{i,j}^{t} X^{i}}{\sum_{i=1}^{m} T_{i,j}^{t}} & if \sum_{i=1}^{m} T_{i,j}^{t} > 0. \\ C^{j,t} & otherwise. \end{cases}$$

Stop when $C^{j,t+1}=C^{j,t}$, j = 1, 2, ... , k, else increase t by 1 and go to step 1.

In this paper, we focus on the initialization method for the clustering. That is to say, our effort is made on finding the cluster centers at iteration 0 : $C^{1,0}$, $C^{2,0}$, ... $C^{k,0}$.

3.2 Density around the Point

We represent the k-th nearest point to X^i as $KNP(i,k)$, and represent all the k nearest neighbors to X^i as $KNN(i,k)$, where

$$KNN(i,k) = \{KNP(i,j)\}_{j=1}^{k}$$

We use k-nearest points radius (KNR) to represent the density around a point, which means the average distance of the k nearest points to it and is computed as follows:

$$KNR(i,k) = \frac{\sum_{X^j \in KNN(i)} distance(X^i, X^j)}{k}$$

where

$$distance(X^i, X^j) = \sqrt{\sum_{t=0}^{n} (X_t^i - X_t^j)^2}$$

3.3 Initial Points Selection Algorithm Description

Based on the assumption that the points with large density around them are most likely to be the final centers, we propose our method of selecting points from dataset as the initial centers.

Our algorithm is recursively finding the point which has the largest density around it. The algorithm is described as following steps.

As figure 1 described, RP represents the set of the remaining points which has not been selected or removed, and IC contains the result of selected initial centers prepared for the clustering. We assume that each cluster has $(min + max)/2$ points. We get the distance matrix of every two points in step 1, and the distance matrix is computed only once. We get the number of clusters k by dividing the number of points by assumed cluster size, that is $k = 2m/(min + max)$. In step 2, we compute the KNR of all points in RP, and find the minimum in step 3. Then we add the point with minimum KNR to IC, and remove the point and its KNN from RP. In step 4 we

refresh RP, if RP is not empty, we go on to step 1, and repetitively find the next initial point.

```
Input
  Dataset D = {X^i}_{i=1}^m, Cluster Size Constraint
  [min, max]
Output
  Initial CentersIC = {X^i}_{i=1}^k, k represents the
  number of clusters
Begin
  step 1)
      Get cluster number k = 2m/(min + max), get
      distance matrix, and initialize RP with
      D.
  step 2)
      For each point X^i in RP, Get its
      KNR(i, (min + max)/2).
  step 3)
      Find the point which has the minimum KNR,
      add it to IC, remove it and its all
      (min + max)/2 nearest neighbors from RP.
  step 4)
      Go to step 2 until RP is empty
```

Fig. 1. The proposed algorithm

Note that we get one point at each iteration. At the last iteration, if there are not enough points to get $KNR(i,(min+max)/2)$, we can calculate the KNR by using the all remaining points, choose one point which has the minimum KNR added to IC and remove all remaining points to stop this algorithm.

3.4 The Time Complexity and Stability of the Algorithm

The complexity of getting the distance array is $O(n^2)$ and the complexity of getting all points' KNR is $O(n^2 \log(n/k))$.

So the total complexity is

$$\boldsymbol{O}(n^2 + kn^2 log(n/k)) = \boldsymbol{O}(kn^2 log(n/k))$$

Our method is deterministic, and is independently on the instance order, for each step of the algorithm is dependent on all the remaining points instead of part of them.

4 Experiments and Discussion

4.1 Dataset

Our experiments consist of two parts. In the first part, we use two real datasets : the Johns Hopkins Ionosphere dataset and the Wisconsin Diagnostic Breast Cancer dataset(WDBC), which are commonly used in clustering and data mining. The Ionosphere dataset contains 351 data points in R^{33} and values along each dimension are normalized to have mean 0 and standard deviation 1. The WDBC data subset consists of 683 normalized data points in R^{9}. The second part is conducted on four real dataset in our real project: proposals on Electronics & Information System and Computer Science from 2008 to 2010 of NSFC(National Natural Science Foundation of China), and they are described in table 1. The size column is the number of proposals in each dataset.

Table 1. Four proposals dataset

Dataset	Category			
	Discipline	***year***	***size***	***dimensions***
1	Electronics and information system	2008	2077	438
2	Computer science	2008	2120	402
3	Computer science	2009	2777	489
4	Electronics and information system	2010	3161	603

We get the feature space of each dataset by removing stop words and noise words, and the number of features of each dataset is shown in dimensions column in table 1.

To evaluate our method, there are two other methods to compare. One is the simplest method by selecting the first k points as the initial centers. The other method is random selection which is widely used in K-Means clustering [13].

4.2 Evaluation Metrics

We use RMSE (Root Mean Square Error) [5] and iteration times as evaluation metrics.

RMSE represents the quality of the final clustering, where $\mathrm{RMSE} = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{k} T_{i,j}(\| X_i - C_j \|^2)/m}$. The less RMSE is, the nearer the result is to the optimal result.

Iteration times describe the convergence speed. The less the iteration times is, the less time it needs to finish the clustering.

4.3 Result and Discussion

We use network simplex method in the step of cluster assignment [3].

In the following tables and figures, sequential represents the method of selecting the first k points as the initial points; random represents randomly selecting k points from the dataset D; select represents our proposed method.

We illustrate the iterating process of clustering on Ionosphere and WDBC datasets with different cluster sizes in figure 2. The variable K in figure 2 represents the number of cluster during the clustering. For a specific dataset, the smaller K is, the bigger cluster size is. The figure shows the change of RMSE during the iterating process of clustering using different initial points selection methods.

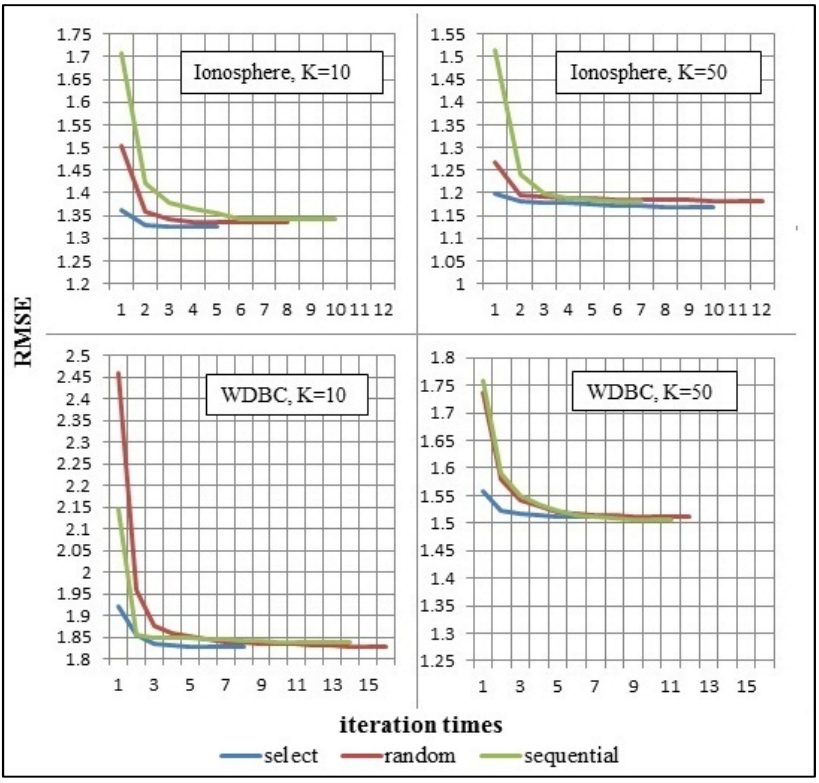

Fig. 2. The iterating process of clustering on Ionosphere and WDBC with different cluster size

As the figure 2 above shows, it's apparent that clustering on the same dataset with bigger cluster size, which uses our proposed method of initial points selection, needs less iteration times and smaller RMSE. That means the convergence is faster and the clustering is more effective. In contrast, the clustering with smaller cluster size needs almost the same iteration times with the other two initial points selection methods. These experiments prove our assumption that the points with large density are most likely to be the final centers. That's because, it's more probable to separate most points into k clusters using our proposed method on dataset with big cluster size. However, clustering with small cluster size may take more noise points into the initial points. Although it can reduce the RMSE at the beginning of the clustering, the

centers of clusters may change frequently as size-constrained added while clustering. Therefore more iteration times are required to converge.

We arrange another experiment using four real datasets of NSFC in real project. We also illustrate the iterating process and RMSE changing of clustering on each of the four dataset using different points selection methods in figure 3. We can directly compare the differences in total iteration times and RMSE between the three methods through the figure. This is different from the normal K-Means clustering in initial points selection.

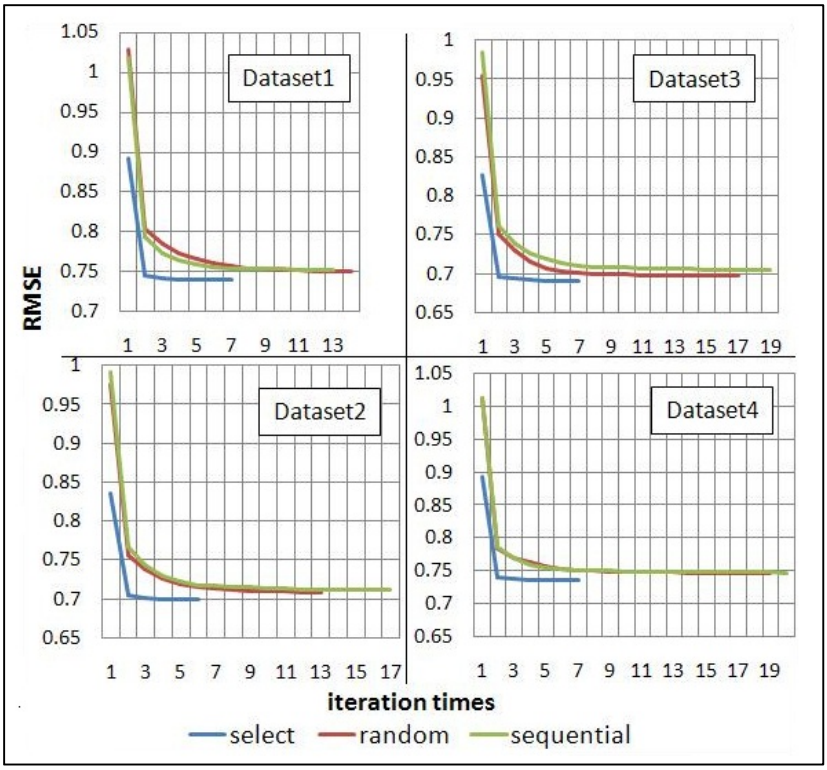

Fig. 3. The iterating process of clustering on each 4 real datasets in our real project

In figure 3, it's apparent that the effect of our proposed initial points selection method is better than the other ones. We can find that our proposed method reduces the iteration times of clustering to only 6 or 7 times, while other two random and sequential methods result in 13 to 21 iteration times. The extra time is mainly in iteration, finishing the clustering and clustering large data. We aim at high dimensions of data in the selection method. We also notice that random selection does not always get better result than sequential method. Sometimes it needs more iteration times to converge. From the table we can see that experiment using sequential method on

dataset 1 only needs 13 iteration times to get the final result while random method needs 14 iteration times, as the uncertain initial centers selected by random selection method may be worse than sequential selection method.

Table 2 shows the final RMSE of size-constrained clustering using three different initial points selection methods on the four dataset. The results show that, on all four dataset, our method can get less final RMSE than the other two methods. The result also shows that the problem, which the clustering result depends highly on initial centers in K-Means, also appears in size-constrained clustering.

Table 2. Final RMSE

Dataset	1	2	3	4
sequential	0.752	0.7118	0.7042	0.747
random	0.7502	0.7091	0.6974	0.7463
select	0.7395	0.6991	0.691	0.7362

Table 3. RMSE at first iteration

Dataset	1	2	3	4
sequential	1.0187	0.9914	0.985	1.0129
random	1.0277	0.9756	0.9555	1.0134
select	0.8925	0.8354	0.8274	0.8945

Table 3 shows that the initial centers of our method are nearer to the final optimal centers, for the RMSE after the first iteration is less than other two methods, about 0.8~0.9, while other two values about 1.

5 Conclusion and Future Work

The clustering result depends highly on initial centers in K-Means clustering method and also in size-constrained clustering. Thus improving the convergence speed and reducing RMSE are two important considerations to optimize size-constrained clustering. In this paper, we propose a method which recursively finds the initial center, which is a point in the max density group of its closest neighbors, during the initial points selection stage. By comparing our method with the other two methods, the sequential one and the random one, we can conclude that that our method can reduce the iteration times by over 50% and get smaller RMSE.

There are two things to consider in our future work. First of all, we need to reduce the time complexity of our points selection algorithm by removing the redundant calculations. Secondly, we would like to conduct experiments on more complex data sets from real applications.

Acknowledgments. This work was supported by NFSC project (Grant No. 61103027), 973 project (No. 2011CB302305) and Shenzhen Gov Projects (JCYJ20120829170028558 and ZYA201106080025A).

References

1. Dubes, R.C., Jain, A.K.: Algorithms for Clustering Data. Prentice Hall (1988)
2. Bradley, P.S., Bennett, K.P., Demiriz, A.: Constrained K-Means Clustering. MSR-RT-2000-65, Microsoft Research (2000)
3. Bertsekas, D.P.: Linear Network Optimization. MIT Press, Cambridge (1991)
4. Kelly, D.J., O'Neill, G.M.: The Minimum Cost Flow Problem and The Network Simplex Solution Method (September 1991)
5. Zhao, J.: Optimal Clustering: Genetic Constrained *K-Means* and Linear Programming Algorithms (2006)
6. Zhu, S., Wang, D., Li, T.: Data clustering with size constraints. Knowledge-Based Systems 23(8), 883–889 (2010)
7. Duda, R.O., Hart, P.E.: Pattern Classification and Scene Analysis. John Wiley and Sons, NY (1973)
8. Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice Hall, Englewood Cliffs (1988)
9. Bradley, P.S., Fayyad, U.M.: Refing initial points for *K-Means* clustering. In: 15th Internat. Conf. on Machine Learning (1998)
10. Deelers, S., Auwatanamongkol, S.: Enhancing K-Means algorithm with initial cluster centers derived from data partitioning along the data axis with the highest variance. Internat. J. Comput. Sci. 2, 247–252 (2007)
11. Khan, S.S., Ahmad, A.: Cluster center initialization algorithm for *K-Means* clustering. Pattern Recognition Letters 25(11), 1293–1302 (2004)
12. Steinley, D., Brusco, J.M.: Initialization *K-Means* Batch Clustering: A Critical Evaluation of Several Techniques. Journal of Classification 24(1), 99–121 (2007)
13. Lozano, J.A., Pena, J.M., Larranaga, P.: An empirical comparison of four initialization methods for the *K-Means* algorithm. Pattern Recognition Lett. 20, 1027–1040 (1999)