Cluster Analysis Via K-means Algorithm

Wang Yuxiang
B20030119
Nanjing University of Posts and Telecommunications
School of Computer Science
Nanjing, China

***Abstract*—Clustering in data analysis means data with similar features are grouped together within a particular valid cluster. Each cluster consists of data that are more similar among themselves and dissimilar to data of other clusters. Clustering can be viewed as an unsupervised learning concept from machine learning perspective. In this paper, we have We examined the two algorithms——K-means and K-means++. We have evaluated the performances of the classical K-means approach of data clustering and the K-means++ method. The accuracy of both these algorithms were examined taking one data set taken from github repository. Their clustering efficiency has been compared in conjunction with two typical cluster validity indices, namely the Davies-Bouldin Index and the Dunn's Index for different number of clusters, and our experimental results demonstrated that the quality of clustering by K-means++ is much efficient than K-Means algorithm when larger data sets with more number of attributes are taken into consideration.**

***Index Terms*—cluster, K-means, DI**

I. Introduction

Retrieving information faster from a group has always been an important issue. Several approaches have been developed for this purpose, one of them is data clustering. Therefore much attention is now paid to invent new fast and improved clustering algorithms. The main goal of clustering is that, the objects present in a group will be much similar to one another and different from the objects present in other groups.

The definition of what constitutes a cluster is always not well defined, and in most applications clusters are not well separated from each other hence, most clustering techniques represent a result as a classification of the data into non-overlapping groups. Clustering is often confused with classification, but there are some differences between the two. In classification, the objects are assigned to some already pre-defined class, whereas in clustering the classes are to be defined.

Learning valuable information from huge volume of data makes the clustering techniques widely applicable in several domains including artificial intelligence, data compression, data mining and knowledge discovery, information retrieval, pattern recognition and pattern classification , and so on.

In this paper, We have examined two algorithms——K-means and K-means++ algorithms. The result strongly depends on the initial selection of centroids. It is very difficult to compare the quality of the clusters produced (e.g. different initial partitions of K produce different results), and also very far data from the centroid may pull the centroid away from the real one. In order to curtail such difficulties and improve the clustering quality and efficiency especially on larger data sets, K-means++ proposed a simple model. Our goal is to analyze and compare the K-Means clustering method with K-means++ algorithm and check the quality of clustering results by using Dunn's separation index (DI) [2] and Davies-Bouldin's index (DBI) [1] respectively.

This paper is organized as follows: In Section II we briefly present the basic idea of cluster validity measures and two widely used validity indices such as DI and DBI used for determining the quality of results obtained from clustering. Section III presents the efficient and productive works done by several researchers in this relevant area. The K-Means clustering method is briefly discussed in Section IV and our work about cluster Analysis Via K-means Algorithm is mentioned in Section V. Section VI concludes the paper.

II. CLUSTERING VALIDATION

Cluster validity issue by and large concerned with determining the optimal number of clusters and checking the fineness of clustering results. Assessment of clustering results is commonly referred to as cluster validation. Many different indices of cluster validity have been already proposed. In this section, we discuss briefly the Dunn's separation Index and Davies-Bouldin's Index which we have used in our proposed clustering algorithm for examining the soundness of clusters.

A. Davies-Bouldin's Index

The Davies-Bouldin's index (DBI) [1] is a function of the ratio of the sum of within-cluster distribution to between-cluster separation. The within i^{th} cluster distribution is defined as:

$$S_{i,q} = (\frac{1}{|A_i|} \sum_{x \in A_i} \|x - v_i\|_2^q)^{1/q}$$

The between i^{th} and j^{th} separation is given by:

$$d_{ij,t} = \left\{ \sum_{s=1}^{p} |v_{si} - v_{sj}|^t \right\}^{1/t} = \|v_i - v_j\|_t$$

where, vi is the i^{th} cluster center, and $(q,t) > 1$, and both $q\&t$ are integers and can be selected independently of each other. A_i is the number of elements in A_i. Next, define $R_{i,qt}$ which is given by:

$$R_{i,qt} = \max_{j \in c, j \neq i} \left\{ \frac{S_{i,q} + S_{j,q}}{d_{ij,t}} \right\}$$

Finally, Davies-Bouldin's index is given by:

$$DB(c) = \frac{1}{c}\sum_{i=1}^{c} R_{i,qt}$$

The objective is to minimize the DBI for achieving proper clustering.

B. Dunn's Index

Another measure is the Dunn's index (DI) [2] measure whose main goal is to maximize the inter-cluster distances and minimize the intra-cluster distances. Dunn's index is defined as:

$$DI(c) = \min_{i \in c}\left\{\min_{j \in c, j \neq i}\left\{\frac{\delta(A_i, A_j)}{\max_{k \in c}\{\Delta(A_K)\}}\right\}\right\}$$

where,

$$\delta(A_i, A_j) = \min\{d(x_i, x_j)\ mid x_i \in A_i, x_j \in A_j\}$$

$$\Delta(A_k) = \max\{d(x_i, x_j) \mid x_i, x_j \in A_i\}$$

d is a distance function, and A_j is the set whose elements are the data points assigned to the i_{th} cluster. The number of cluster that maximizes DI is taken as the optimal number of the clusters.

The appropriate clustering algorithm and parameter settings heavily depend on the input data set taken into consideration. An ideal cluster can be said to be a set of data points that is more isolated and compact from other data points.

III. RELATED WORKS

A non-metric distance measure for similarity estimation based on the characteristic of differences [5] is presented and implemented on K-Means clustering algorithm. The performance of this kind of distance and the Euclidean and Manhattan distances were then compared. A new line symmetry based classifier (LSC) [6] deals with pattern classification problems. LSC is well-suited for classifying data sets having symmetrical classes, irrespective of any convexity, overlap and size. The shortcomings of the standard K-Means clustering algorithm can be found in the literature [7] in which a simple and efficient way for assigning data points to clusters is proposed. Their improved algorithm reduces the execution time of K-Means algorithm to a great extend. A simple and efficient implementation of K-Means clustering algorithm called the filtering algorithm [8] shows that the algorithm runs faster as the separation between clusters increases. The various types of clustering algorithms along with their applications in some benchmark data sets were surveyed in [9]. Several proximity measures, cluster validation and various tightly related topics were discussed. A new generalized version of the conventional K-Means clustering algorithm which performs correct clustering without pre-assigning the exact cluster number can be found in [10]. Based on the definition of nearest neighbor pair C. S. Li et al. [11] proposed a new cluster center initialization method for K-Means algorithm. In iterative clustering algorithms, selection of initial cluster centers is extremely important as it has a direct impact on the formation of final clusters. An algorithm to compute the initial cluster centers for K-Means algorithm was given by M. Erisoglu et al. [12] and their newly proposed method has good performance to obtain the initial cluster centers converges to better clustering results and almost all clusters have some data in it. An Efficient KMeans Clustering Algorithm for Reducing Time Complexity using Uniform Distribution Data Points [13] was proposed. The accuracy of the algorithm was investigated during different execution of the program on the input data points. Finally, it was concluded that the elapsed time taken by proposed efficient K-Means is less than K- Means algorithm.

IV. K-MEANS CLUSTERING ALGORITHM

The K-Means Clustering algorithm is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. In 1967, Mac Queen [4] firstly proposed the K-Means algorithm. During every pass of the algorithm, each data is assigned to the nearest partition based upon some similarity parameter (such as Euclidean distance measure). After the completion of every successive pass, a data may switch partitions, thereby altering the values of the original partitions. Various steps of the standard K-Means clustering algorithm is as follows:

1) The number of clusters is first initialized and accordingly the initial cluster centers are randomly selected.
2) A new partition is then generated by assigning each data to the cluster that has the closest centroid.
3) When all objects have been assigned, the positions of the K centroids are recalculated.
4) Steps 2 and 3 are repeated until the centroids no longer move any cluster.

The main objective of K-Means is the minimization of an objective function that determines the closeness between the data and the cluster centers, and is calculated as follows:

$$J = \sum_{j=1}^{K}\sum_{i=1}^{N} \|d(X_i, C_j)\|$$

where $\|d(X_i, C_j)\|$ is the distance between the data X_i and the cluster center C_j . The downside of K-Means algorithm is that, the result of clustering mostly depends on the initially selected centroids. Spherical data sets cannot be efficiently clustered using K-Means. And only numerical values attributes can be ably clustered.

K-means remains a popular and efficient clustering algorithm, especially for well-behaved datasets with relatively simple structures, although it has some limitations:

1) **Sensitivity to Initial Centroids:** K-means is sensitive to the initial placement of centroids. Different initializations can lead to different final cluster assignments. This sensitivity can result in suboptimal or even incorrect clustering solutions.
2) **Dependence on the Number of Clusters (K):** The algorithm requires the user to specify the number of clusters (K) beforehand. Choosing an inappropriate value for K

can lead to poor clustering results. Various methods, such as the elbow method or silhouette analysis, are used to estimate an optimal K, but these are not foolproof.

3) **Sensitive to Outliers:** Outliers or noise in the data can significantly impact K-means results. The algorithm tries to minimize the sum of squared distances, making it sensitive to outliers, which can distort the centroid positions and affect cluster assignments.
4) **Global Optimum:** K-means aims to find a global optimum, but it can get stuck in local minima. Multiple initializations with different starting points are often used to mitigate this issue, but it doesn't guarantee finding the true global optimum.

V. CLUSTER ANALYSIS

A. Dataset

K-means is a common clustering algorithm. Clustering belongs to unsupervised learning in machine learning classification. In the case that data sets are not marked, it is convenient to group data. In k-means, K refers to dividing the data set into K subsets.

Data sets for clustering can be downloaded from GitHub, address to https://github.com/mubaris/friendly-fortnight/blob/master/xclara.csv. So we choose this dataset to train and evaluate our model. The data scale is 3000*2, as shown below:

TABLE I
DATASET

	A_x	A_y
1	2.072345	-3.241693
2	17.93671	15.78481
3	1.083576	7.319176
4	11.12067	14.40678
5	23.71155	32.02478
...	...	...

B. Experiment Setup

According to the steps of the algorithm of kmeans, we unfold the experimental procedure. Firstly the prepared data is read in, by printing the shape of the data we can know that the dataset consists of 3000 points in total, each point has two coordinate values x and y. The purpose of the algorithm is to classify these 3000 points into K classes with high quality.

*1) **based on K-means**:* First we define a function called show_cluster which is used to display the clustering results. The function takes three arguments: dataset, cluster, and centre.

- dataset is a two-dimensional array of points in the dataset.
- cluster is a one-dimensional array of cluster labels for each point.

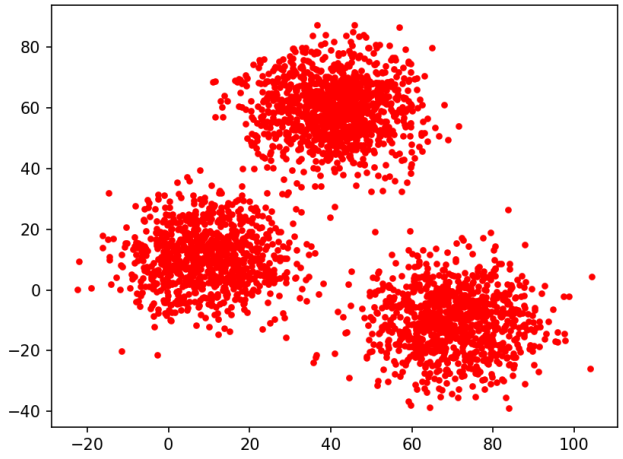

Fig. 1. k-means results when K = 1.

- centre is a two-dimensional array representing the centre of each cluster.

The two lists, colours and markers, are used to set the colours and markers of the different clusters. Then function computes the number of categories K of the clusters and iterates over each category. In each traversal, the plt.scatter function is used to draw all the points that belong to the current category. If centre is not None, the centre of each cluster need to draw.

When all the points are of the same type, i.e. when the centre cluster of all the points is 1, the result can be printed as shown in Figure 1. Also this figure shows the distribution of all the points in the dataset.

When we use the simple k-means algorithm, the initial cluster centres can be determined by randomly selecting out the city centre points among the sample points. This step is achieved through the function Random_init. Using the Euclidean distance as a metric, for each sample point in the dataset, the closest centroid to it is found and the cluster to which the current sample belongs is updated. The iteration is terminated when the classification of all samples no longer changes or when the number of iterations reaches the currently set threshold.

For example, when we set the value of K to 3, the initial cluster centre is obtained by randomly selecting two centroids. The final result can be obtained by iteration just like fig2.

Observation: It can be found that the number of iterations obtained from multiple runs of the program is not the same when a fixed value of K is selected, which is mainly related to the selection of the initial cluster centre. When the initial cluster centre is selected near the optimal solution, the number of iterations is significantly reduced, so the selection of the initial cluster centre is important. As shown in Fig. 3, when K is also 3, a suitable initial solution reduces the number of iterations to 3.

Based on this observation that the selection of initial points has a great impact on the results, we propose a further

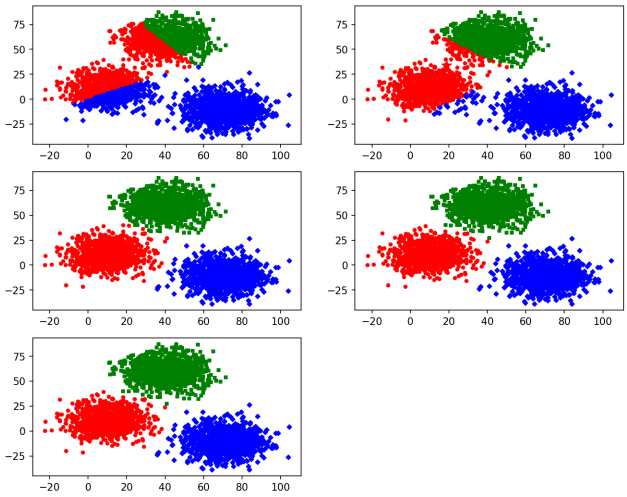

Fig. 2. The result obtained by randomly selecting the initial centre point when K = 3

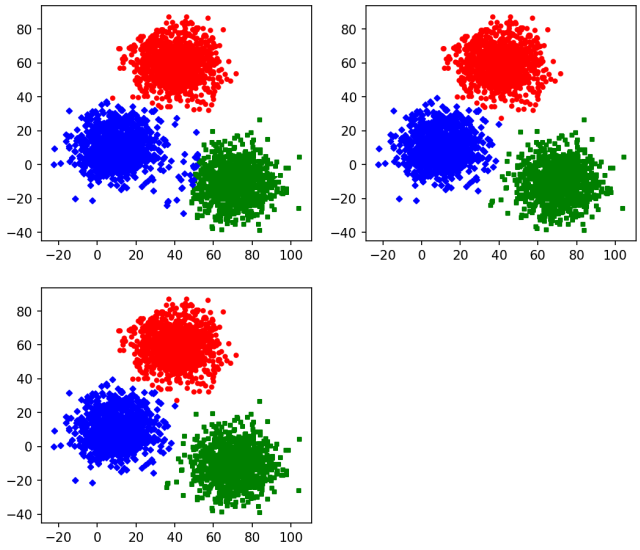

Fig. 3. Suitable initial solutions lead to fewer iterations

optimisation for the selection of initial points, the basic idea is that the distance between the initial cluster centroids should be as far as possible, firstly, a seed point is selected empirically, and let a seed point be randomly selected from the data that are farther away from the first seed point based on the weights until the number of seed points is equal to K. We call this method the **K-means++** method, which is an improved algorithm based on K-means.

*2) **based on K-means++**:* The main work of the K-means++ algorithm is embodied in the selection of seed points, the basic principle is to make the distance between each seed point as large as possible, but need to exclude the influence of noise. The following is the basic idea:

a. Randomly select a point as the first cluster centre from the input set of data points (K clusters are required).
b. For each point x in the data set, calculate the distance D(x) between it and the nearest cluster centre which means the selected cluster centre.
c. Select a new data point as the new clustering centre, the principle of selection is that the point with larger D(x) has a higher probability of being selected as the clustering centre.
d. Repeat steps (b) and (c) until K clustering centres are selected.
e. Use these K initial clustering centres to run the standard K means algorithm.

Algorithm 1 K-means++ algorithm

Input: K and $DataSet$;
Output: **center cluster**;

A sample point c_1 from the data set is randomly selected as the initial cluster centre
for all $i = 1, 2, \cdots, K-1$ **do**
  for all $j = 1, 2, 3, \cdots, 3000 - i$ **do**
    $D_j^2 = \min(x_j - c_i)^2$
  end for
  choose a new c_{i+1} with the probability of $P_{c_{i+1}} = \frac{D_{c_{i+1}}^2}{\sum(D_x^2)}$
end for
for all $i = 1, 2, \cdots, Iters$ **do**
  $dist_{i,j} = \|dataset_i - center_j\|$;
  $cluster_i = \arg\min_j dist_{i,j}$;
  $new_center_j = mean(dataset[cluster == j])$
  if $center = new_center$ **then**
    break;
  end if
end for
return cluster center.

Based on the algorithmic flow of K-means, it is relatively simple to implement K-means++. We only need to change part of the selection of the initial point, from the previous random selection, changed to similar to the farthest point sampling algorithm.

We implement a function to do the initial cluster centre selection. The function first creates an array of zeros, center, of shape (K, dataset.shape[1]) to store the cluster centres. Then, the function uses the $np.random.choice$ function to randomly select a point from the dataset as the first cluster centre. Next, the function performs K-1 loops, choosing a new cluster centre each time. In each loop:

- The function calculates the distance from each data point to the cluster centre that has been chosen: $dist = np.array([np.linalg.norm(dataset - c, axis = 1) forcincentre[: i]])$. The Euclidean distance is used here.
- The function selects the value with the smallest distance: $dist = np.min(dist, axis = 0)$.
- The function calculates the probability that each data point will be selected as a new cluster centre: $prob =$

$\frac{dist}{np.sum(dist)}$. This probability is proportional to the minimum distance from the data point to the already selected cluster centre.

- The function chooses a new cluster centre from the dataset based on this probability: $center[i] = dataset[np.random.choice(len(dataset),\ p = prob)]$. Finally, the function returns the chosen cluster centre.

C. Experimental Results

We examined the performance of the above described algorithms on the data sets taken from the repository addressed to https://github.com/mubaris/friendly-fortnight/blob/master/xclara.csv. To assess the efficiency of both methods, we compare the results obtained by general K-means clustering method against the improved clustering results returned by the K-means++ algorithm on the data set.

The performance of traditional K-means and K-means++ algorithms are measured in terms of two standard validity measures namely Dunn's index (DI) [2] and Davies-Bouldin's index (DBI) [1] on the data set. Table 2 gives a comparative analysis of the above said facts.

Since the data in the dataset is very healthy, when K is 3, the results always converge to an optimal solution whether the initial points are randomly selected or the farthest points are obtained by sampling the farthest points, so the DI and DBI metrics present the **same results** in the experiment.

On the other hand, we can see the advantage of the K-means++ algorithm, when the procedure is executed several times, it can be noticed that the K-means++ algorithm converges much faster, with a significant increase in the notion of converging to an optimal solution through only three iterations (when K is 3).

TABLE II
COMPARISON OF K-MEANS AND K-MEANS++ CLUSTERING ALGORITHMS BY CONSIDERING DUNN'S AND DAVIES-BOULDIN'S INDEX ON DIFFERENT SIZED DATA SETS (WITH K=3).

	DI	DBI
K-means	0.42056	0.04661
K-means++	0.42056	0.04661

VI. SUMMARY

In this paper, we have examined two clustering algorithms – the customary K-means algorithm and K-means++ algorithm. It can be seen from the experimental result that, generally speaking, K-means algorithm can do a pretty good job in clustering data sets in any K numbers of clusters. However, the algorithm is significantly responsive to the preliminary selection of cluster centroids. Hence, in order to minimize such complexities K-means++ algorithm proposed a simple and efficient method for finding the initial cluster centers. Further, considering both the DI and DBI parameters for cluster validation on the data set, although the results obtained by K-means++ algorithm produces the same quality of clustering as compared to K-means algorithm, we need to recognise that K-means++ is good at convergence.

REFERENCES

[1] D.L. Davies and D.W. Bouldin, "A Cluster Separation Measure," "IEEE Trans. Pattern Analysis and Machine Intelligence,"vol.1, pp.224-227, 1979.

[2] J.C. Dunn,"A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters," J.Cybernetics, vol. 3, pp. 32- 57, 1973.

[3] Qinghao Hu, Jiaxiang Wu, Lu Bai, Yifan Zhang, and Jian Cheng. 2017. "Fast K-means for Large Scale Clustering. "In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM '17). Association for Computing Machinery, New York, NY, USA, 2099–2102. https://doi.org/10.1145/3132847.3133091

[4] J. Mac Queen, "Some methods for classification and analysis of multivariate observations", "Fifth Berkeley Symposium on Mathematics, Statistics and Probability", pp.281-297, University of California Press, 1967.

[5] Z. Li, J. Yuan, H. Yang and Ke Zhang,"K-Mean Algorithm with a Distance Based on the Characteristic of Differences", "IEEE International conference on Wireless communications, Networking and mobile computing", pp. 1-4, Oct.2008.

[6] S. Saha S. Bandyopadhyay and C. Singh, "A New Line Symmetry Distance Based Pattern Classifier", "International joint conference on Neural networks as part of 2008 IEEE WCC", pp.1426-1433, 2008.

[7] Shi Na, L. Xumin, G. Yong, "Research on K-Means clustering algorithm-An Improved K-Means Clustering Algorithm", "IEEE Third International Symposium on Intelligent Information Technology and Security Informatics", pp.63-67, Apr.2010.

[8] T. Kanungo, D. Mount, N. Netanyahu, C. Piatko and A. Wu, "An Efficient K-Means Clustering Algorithm: Analysis and Implementation", "IEEE Transactions on Pattern analysis and Machine intelligence", vol. 24, no.7, 2002

[9] R. Xu and D. Wunsch, "Survey of Clustering Algorithms", "IEEE Transactions on Neural networks", vol. 16, no. 3, May 2005.

[10] Y.M. Cheung, "A New Generalized K-Means Clustering Algorithm", "Pattern Recognition Letters, Elsevier",vol.24,issue15, 2883–2893, Nov.2003.

[11] C. S. Li, "Cluster Center Initialization Method for K-means Algorithm Over Data Sets with Two Clusters", "2011 International Conference on Advances in Engineering, Elsevier", pp. 324-328, vol.24, 2011.

[12] M. Erisoglu, N. Calis and S. Sakallioglu, "A new algorithm for initial cluster centers in K-Means algorithm","Published in Pattern Recognition Letters", vol. 32, issue 14, Oct.2011.

[13] D. Napoleon and P. G. Laxmi, "An Efficient K-Means Clustering Algorithm for Reducing Time Complexity using Uniform Distribution Data Points", "IEEE Trendz in Information science and computing", pp.42-45, Feb.2011.