Scholars Journal of Engineering and Technology (SJET)
Sch. J. Eng. Tech., 2015; 3(6):591-594
©Scholars Academic and Scientific Publisher
(An International Publisher for Academic and Scientific Resources)
www.saspublisher.com

ISSN 2321-435X (Online)
ISSN 2347-9523 (Print)

Research Article

An efficient clustering method of assigning nearest cluster center

Hongbo Zhou
Northeast Petroleum University,Institute of Computer and Information Technology, Daqing, Heilongjiang, China

***Corresponding author**
Hongbo Zhou
Email: jiessie9@126.com

Abstract: K-means is a numerical, unsupervised, non-deterministic, simple iterative cluster method. k-means algorithm computes the distances between data point and all centers in iterative process, this is computationally very expensive especially for huge datasets. we propose an improved k-means algorithm to preprocess an efficient way for assigning data points to clusters. The improved method avoids large number of iterations, and saves running time. Experimental results show that the improved method can effectively improve the speed of clustering and accuracy, reducing the computational complexity of the k-means.
Keywords: data mining; k-means algorithm; squared-error criterion; Euclidean distance; cluster center

INTRODUCTION

Data mining [1] is to nugget out potential, hidden useful knowledge and information from abundant, cluster is an important method in data mining. It attempts to create relatively high similarity in the cluster, relatively low similarity between clusters.

K-means [2] is a numerical, unsupervised, non-deterministic, simple iterative cluster method. In many practical applications, the method is proved to be a very effective way that can produce good clustering results. However, the k-means algorithm computes the distances between data point and all centers in iterative process; this is computationally very expensive especially for huge datasets. We propose an efficient implementation method for implementing the k-means method. The improved algorithm reduce large number of iterations so it superior to the standard k-means method on running time and algorithm complexity. Our algorithm produces the same clustering results as that of the k-means algorithm, and has significantly superior performance than the k-means algorithm. By comparing the experimental results of the standard k-means and the improved k-means, it shows that the improved method can effectively shorten the running time and accuracy.

K-MEANS ALGORITHM

K-mean algorithm is based on decomposition, most widely used in data mining field. The concept is use k as a parameter, divide n object into k clusters, to create relatively high similarity in the cluster, relatively low similarity between clusters. And minimize the total distance between the values in each cluster to the cluster center. The cluster center of each cluster is the mean value of the cluster.

The calculation of similarity is done by mean value of the cluster objects. The measurement of the similarity for the algorithm selection is by the reciprocal of Euclidean distance[3].That is to say, the closer the distance, the bigger the similarity of two objects, and vise versa.

The algorithm consists of two separate phases. The first phase selects k centers randomly, where the value k is fixed in advance. The next phase is to take each data object to the nearest center. Euclidean distance is generally considered to determine the distance between each data object and the cluster centers. When all the data objects are included in some clusters, the first step is completed and an early grouping is done. Recalculating the average of the early formed clusters. This iterative process continues repeatedly until the criterion function becomes the minimum.

The process of k-means algorithm as follow:

A. Input: Number of desired clusters, k, and a database $D=\{d_1,d_2,\ldots,d_n\}$ containing n data objects.

B. Output: A set of k clusters

Steps:

a) Randomly select k data objects from dataset D as initial cluster centers.

b) Repeat;

c) Calculate the distance between each data object di($1\leq i\leq n$) and all k cluster centers cj ($1\leq j\leq k$} and assign data object di to the nearest cluster

d) For each cluster cluster center j(1≤j≤k),recalculate the cluster center.

e) Until no changing in the center of clusters.

Typically, the squared-error criterion [4] is used, defined as:

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} \| p - m_i \|^2 \qquad (1)$$

where E is the square-error sum for all objects in the data sets, p is the point in space representing a given object, and m_i is the mean of cluster C_i. This criterion tries to produce k clusters so that the objects in the same cluster are as compact as possible while the objects in different clusters are as separate as possible.

The distance of criterion function is Euclidean distance, which is used for determining the nearest distance between each data objects and cluster center. The Euclidean distance between one vector $X_i=(x_{i1}, x_{i2}, \ldots, x_{in})$ and another vector $X_j=(x_{j1}, x_{j2}, \ldots, x_{jn})$,the Euclidean distance $d(x_i,y_i)$ can be obtained as follow:

$$d_{ij} = \sqrt{(X_{i1} - X_{j1})^2 + (X_{i2} - X_{j2})^2 + \ldots + (X_{in} - X_{jn})^2} \qquad (2)$$

Cluster center is the average of cluster data objects, the nearer the cluster center between the two clusters, the more similar the two clusters, the farther the cluster center, the less these two clusters similar.

Cluster center of one cluster is calculated as follows: Let n data object containing the data set $X=\{x_1,x_2,x_3,...,x_n\}$, respectively, the cluster center $z_1,z_2,\ldots,z_k$.

$$z_i = \frac{1}{C_i} \sum_{j \in n_i} j \quad i = 1,2,\ldots, \ k \qquad (3)$$

Among them, C_i representative of the number of samples in class i, n_i behalf of class i in the sample set.

IMPROVED K-MEANS CLUSTERING

K-means algorithm computes the distances between data point and all centers, this is computationally very expensive especially for huge datasets. In this section, we introduce the proposed idea that makes k-means more efficient, especially for dataset containing large number of clusters. The proposed idea comes from the fact that the k-means algorithm discovers spherical shaped cluster, whose center is the gravity center of points in that cluster, this center moves as new points are added to or removed from it. This motion makes the center closer to some points and far apart from the other points, the points that become closer to the center will stay in that cluster, so there is no need to find its distances to other cluster centers. The points far apart from the center may change the cluster, so only for these points their distances to other cluster centers will be calculated, and assigned to the nearest center. For each data point, we can keep the distance to the nearest cluster. At the next iteration, we compute the distance to the previous nearest cluster. If the new distance is less than or equal to the previous distance, the point stays in its cluster, and it is no need to compute its distances to the other cluster centers. This saves the time required to compute distances to k−1 cluster centers.

In the proposed method, we write a functions called assignNearestCenter(),which is shown as follows. Line 2 finds the distance between the current point xi and the exits cluster center assigned to it in the previous iteration, if the computed distance is smaller than or qual to the distance to the old center, the point stays in its cluster that was assigned to in previous iteration, and there is no need to compute the distances to the other k−1 centers. Lines 3~5 will be executed if the computed distance is larger than the distance to the old center, this is because the point may change its cluster, so Line 4 computes the distance between the current point and all k centers. Line 6 searches for the closest center, Line 7 assigns the current point to the closest cluster and increases the count of points in this cluster by one, Line 8 updates mean squared error. Lines 9 keep the cluster id, for the current point assigned to it, and its distance to it to be used in next call of that function (i.e. next iteration of that function).This information is kept in Line 9 allows this function to reduce the distance calculation required to assign each point to the closest cluster, and this makes the function faster than the old algorithm. Line 10~12 updates k new clustering center.

```
Function assignNearestCenter()
1 E=0;
2 for i=1 to n
   Compute squared Euclidean distance d($x_i$,clusterid[i]);
     if(d($x_i$,clusterid[i])<=pointdis[i]
     point stay in its cluster;
    else
3   for j=1 to k
4     compute squared Euclidean distance d($x_i$,$m_j$);
5   endfor
6 find the closest centroid mj to $x_i$;
7 $m_j=m_j+x_i$;$n_j=n_j+1$;
8 E=E+d($x_i$,$m_j$);
```

```
9 clustered[i]= number of the closest centroid;pointdis[i]=Euclidean distance to the closest centroid;
    endfor
10 For j=1 to k
11  $m_j=m_j/n_j$;
12 endfor
```

The k-means algorithm converges to local minimum. Before the k-means converges, the cluster center computed number of times, and all points are assigned to their nearest center, i.e., complete redistribution of points according to new centroids, this takes O(nkm),where n is the number of cluster points, k is the number of clusters centers, and m is the number of iterations. In our k-means algorithm, to obtain initial clusters, this process requires O(nk)(n,k and m are the same as above).Here, some cluster points remain in its cluster, the others move to another cluster. If the point stays in its cluster this require O(m),otherwise require O(k).If we suppose that half points move from their clusters, this requires O(nk/2),since the algorithm converges to local minimum, the number of points moved from their clusters decreases in each iteration. Even for large number of iterations, we expect the total cost is $nk\sum_{i=1}^{m} 1/i, nk\sum_{i=1}^{m} 1/i$ is much less than nkm.So the cost of using improved k-means algorithm approximately is O(nk).

EXPERIMENTAL RESULT

The experiment environment of this context was Intel i53470,3.2GHZ,1T hard disk and Windows 7 OS. Used Matlab [5] to validate the validity of improved algorithm. The experiment introduced four data sets provided by UCI [6] public database. They are soybean, zoo, wine and breast cancer (BC).the types of attribution in soybean and zoo were categorical and binary, and the types in wine and BC were ratio-scaled. Following content was experiment result. Table 1 was the comparison of the efficiency of average running time.

Table-1: The efficiency of average running time(/s)

dataset	standard k-means algorithm	improved k-means algorithm	percentage of save time
Soybean	0.8211	0.4202	48.9%
Zoo	1.2572	0.5216	58.6%
Wine	3.0136	1.3221	56.1%
BC	11.2692	4.0127	64.4%

Initial cluster centers were randomly selected in standard K-means algorithm, leading to the different clustering result in each time and the unstable quality. It is can be seen from the table 1,in the aspect of clustering time and efficiency, the improved k-means algorithm reduces running time and saves about 50 percent of time in four data sets. The percent of time saving will be more evident with the increase of data scale. Compared with the standard K-means algorithm, the improved k-means algorithm can reduce the iterative times and improved running efficiency.

CONCLUSION

K-means algorithm is one of the most widely used clustering algorithm in spatial clustering analysis, but the standard algorithm which selects k objects randomly from population as initial centoids cannot always give a good and stable clustering. The computational complexity of the standard k-means algorithm is objectionably high owing to the need to reassign the data points a number of times during every iteration, which makes the efficiency of standard k-means clustering is not high. This paper presents a new method to preprocess the datasets, and an efficient way for assigning data points to clusters. Experimental results show that improved k-means clustering algorithm can lead to a better clustering.

ACKNOWLEDGMENT

The paper is supported by the Education Department of Heilongjiang province science and technology research projects (No.1253G014).

REFERENCES

1. MacQueen J; Some methods for classification and analysis of multivariate observations[C]. Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability.Berkeley:University of California Press,1967:281-297.
2. Jain A K, Dubes RC; Algorithms for Clustering Data.Englewood Cliffs,NJ, USA: Prentice-Hall, 1988.
3. Xu R, Wunsch D; Survey of clustering algorithm.IEEE Trans.Neural Networks, 2005; 16(3): 645-678.
4. Milligan GW, Cooper MC; An examination of procedures for determining the number of clusters in a data set.Psy-chometrika, 1985, 50: 159-179.

5. UCI Repository of machine learning databases and domain theories[EB/OL].FTP address: ftp://ftp.ics.uc.i edu/pub /machine-learning-databases
6. Xie XL, Beni GA; validity measure for fuzzy clustering. IEEE Trans.Pattern Anal. Machin. Intell, 1991; 13(8): 841-847.