A K-means Clustering Algorithm for Automatically Obtaining K Value

Yuan-qiang XIE* and Rui-ming FANG

College of Information Science and Engineering, Huaqiao University, Xiamen 361021, China

*Corresponding author

Keywords: K-means clustering algorithm, Distance matrix, Distance maximization, Automatic acquisition.

Abstract. In order to solve the problem of manual input of K value in traditional K-means clustering algorithm, a method to obtain K value automatically is proposed. Firstly, the data needed to be clustered are sampled, and the distance between the data in the sample group is calculated. Then the distance matrix is formed and the de-noising process is done, and the clustering number K is selected based on the principle of distance maximization. Finally, we take two-dimensional data as an example to simulate the K-means clustering algorithm of the number of clusters K based on the principle of distance maximization, then the simulation results are verified by MATLAB. The experimental results show that the algorithm can obtain K value automatically and improve the accuracy of clustering.

Introduction

The K-means algorithm is a classical clustering algorithm proposed by MacQueen J in 1967 [1], which is based on partitioning method. It has the advantages of high efficiency, easy to understand and implement, and it is used in many fields, such as applied mathematics, pattern recognition, image segmentation and bioengineering. However, it is generally necessary for the user to give the clustering number K value according to experience, which will increase the burden on the user.

For the selection of K-value in K-means clustering algorithm, scholars have done relevant research [2-7]. However, there are still some disadvantages of these methods. Therefore, in view of the shortcomings of the above K-means clustering algorithm for obtaining the number of clusters K, this paper proposes an automatic method to obtain K value.

K-means Clustering Algorithm

The basic idea of the algorithm is that the K-means algorithm is a hard-clustering algorithm which divides n sample data into K classes in n-dimensional Euclidean space. First, the user determines the exact number of objects to be clustered K, and selects K objects as the center randomly. Then, according to the distance between the data and the center, it is assigned to the nearest class, and the average value of the objects in each class is recalculated to form a new cluster center. Iterate over and over until the objective function converges [8,9]. The definitions are as follows:

Definition 1 The Euclidean distance between any two sample points [10]:

\[ d(x_i, x_j) = \sqrt{|x_{i1} - x_{j1}|^2 + |x_{i2} - x_{j2}|^2 + \cdots + |x_{ip} - x_{jp}|^2} \]

Where \( x_i = (x_{i1}, x_{i2}, \ldots, x_{ip}) \) and \( x_j = (x_{j1}, x_{j2}, \ldots, x_{jp}) \) are two \( p \)-dimensional data objects.

Definition 2 The centroid of multiple sample points, i.e., the cluster center of a subclass is:

\[ m_i = \frac{1}{|K_j|} \sum_{x_k \in K_j} x_k \]

where the number of samples in a class \( K_j \) is represented by \(|K_j|\).

Definition 3 The clustering end condition and criterion function are as follows:

\[ \left\| \sum_{i=1}^{K} \sigma(K_{j,i} - K_{j-1,i}) \right\| < \varepsilon \]

where \( K \) is the total number of cluster centers, \( i \) is cluster center, \( t \) is the number of iterations at present, and \( \varepsilon \) represents infinitesimal.
K-means clustering algorithm belongs to a dynamic clustering algorithm. One notable feature of the algorithm is the need for continuous iteration. In the process of iteration, first of all, the new clustering centers of each subclass are obtained; then, the conclusion condition of clustering is judged; once the end condition is satisfied, the iteration is terminated; finally, the classification results of the algorithm are obtained.

**Improved K-means Clustering Algorithm**

We do the data de-noising preprocessing, then reasonably select the clustering center, and iterative algorithm to get the number of clustering. The processing process is as follows:

1. Calculate the distance \( d(x_i, x_j) \) between data and store them in the matrix \( D_{T \times T} \).

\[
D_{T \times T} = \begin{bmatrix}
  d(x_1, x_1) & \cdots & d(x_1, x_T) \\
  \vdots & \ddots & \vdots \\
  d(x_T, x_1) & \cdots & d(x_T, x_T)
\end{bmatrix}
\]

(1)

Where \( d(x_i, x_j) \) — the distance between the-sampling data, \( D_{T \times T} \) — the T-dimensional distance matrix between the-sampling data.

2. The average value of each column vector in the distance matrix \( D_{T \times T} \) is calculated.

3. The noise points whose error percentage is more than 30% are removed by formula (4).

\[
R_i = \frac{\sum_{j=1}^{T} d(x_i, x_j)}{T-1} (i=1,2,\ldots,T)
\]

(2)

\[
\overline{R} = \frac{\sum_{i=1}^{T} R_i}{T} (i=1,2,\ldots,T)
\]

(3)

\[
\delta = \frac{|R_i - \overline{R}|}{\overline{R}}.
\]

(4)

Where \( T \) - the number of the data, \( R_i \) - the average value of the \( ith \) column vector in the matrix, \( \overline{R} \) - the average value of \( T \) \( R_i \), \( \delta \) - the percentage of errors between \( R_i \) and \( \overline{R} \).

4. The remaining data form a data set \( S_n \) containing several data sets \( n(n \leq T) \), calculate the distance \( d(x_w, x_v) \) and store them in the distance matrix \( D_{non} \). Firstly, two data objects \( w,v \) with the largest distance are selected as the initial clustering center, \( d_{wv} = \max d(x_w, x_v) \), supposed \( x_1^* = x_w, x_2^* = x_v, d_{wv} = d_i^* \).

5. The remaining data objects classified by cluster centers \( x_1^*, x_2^* \), \( \forall l \in \{1,2,\ldots,n/w,v\} \). If \( d(x_i, x_1^*) < d(x_i, x_2^*) \), \( x_i \) will be divided into classes \( x_1^* \); otherwise, \( x_i \) will be divided into classes \( x_2^* \). \( S_n \) is divided into two categories by cluster centers \( x_1^*, x_2^* \), separately recorded as \( S_{21}^*, S_{22}^* \).

6. Take \( i = 2 \) and \( i \in l \), calculate the distance of all data objects in this class \( S_{ii}^* \) to \( x_i^* \), we can get \( d_i = \max \{d(x_j, x_i^*) \}, j \in S_{ii}^* \) ... calculate the distance of all data objects in this class \( S_{ii}^* \) to \( x_i^* \), we can get \( d_u = \max \{d(x_j, x_i^*) \}, j \in S_{ii}^* \), take \( d_i^* = \max \{d_i, d_{i1}, \ldots, d_{iu} \} \).

7. If \( d_i^* > m \cdot \text{average}(d_i^* + d_{i1}^* + \cdots + d_{iu}^*) \) ( \( m \) is the parameter of distance maximization algorithm [11], the value of \( m \) is inversely proportional to the number of initial clustering centers obtained by the algorithm, \( 0.5 \leq m \leq 1 \), generally take 0.5), then the \((i+1)\) cluster center is \( x_{i+1}^* \), repeat steps (6); otherwise, output \( i \), i.e., the number of clusters \( K = i \), and the value of \( K \) is obtained.
Example Simulation Analysis

Simulation Experimental Data

Taking two-dimensional data as an example, we give a data set with 28 data objects. We use the improved clustering algorithm to obtain the clustering number $K$ of the data set, and then use the $K$ value as the input of the clustering number of the traditional K-means clustering algorithm. Finally, the clustering results are verified by MATLAB. The following 28 data points as shown in Table 1.

<table>
<thead>
<tr>
<th>data points1~7</th>
<th>data points8~14</th>
<th>data points15~21</th>
<th>data points22~28</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_1(4.8)$</td>
<td>$X_8(9.5,7)$</td>
<td>$X_{15}(7,10)$</td>
<td>$X_{22}(7.5,9)$</td>
</tr>
<tr>
<td>$X_2(8.5,10)$</td>
<td>$X_9(4.6)$</td>
<td>$X_{16}(10,6)$</td>
<td>$X_{23}(5,9)$</td>
</tr>
<tr>
<td>$X_3(5.6)$</td>
<td>$X_{10}(4,10)$</td>
<td>$X_{17}(4.5,3.5)$</td>
<td>$X_{24}(8.5,5.5)$</td>
</tr>
<tr>
<td>$X_4(8.6,5)$</td>
<td>$X_{11}(7,9)$</td>
<td>$X_{18}(9,7)$</td>
<td>$X_{25}(3.5,5)$</td>
</tr>
<tr>
<td>$X_5(4.9)$</td>
<td>$X_{12}(4.5,5)$</td>
<td>$X_{19}(10,8)$</td>
<td>$X_{26}(4.5,6)$</td>
</tr>
<tr>
<td>$X_6(7.8)$</td>
<td>$X_{13}(9.5,6)$</td>
<td>$X_{20}(4.5,9)$</td>
<td>$X_{27}(4.5,8)$</td>
</tr>
<tr>
<td>$X_7(5.5)$</td>
<td>$X_{14}(4.5,8.5)$</td>
<td>$X_{21}(9,6)$</td>
<td>$X_{28}(4.5)$</td>
</tr>
</tbody>
</table>

Calculation of $K$ Value Based on Improved Algorithm

The distance $d(x_i, x_j)$ $(i, j = 1,\ldots,28)$ between the 28 data points is obtained and stored in the distance matrix $D_{28*28}$. Then the 28 data points are de-noised, and the results show that all the 28 data points come from the same cluster, and no de-noising processing is needed.

28 data points are clustered according to the improved K-means clustering algorithm. The clustering process is shown in the following figure:

![Figure 1. Cluster partitioning of 28 data points.](image)

When clustering to four clustering centers, the algorithm ends, and the output number of clustering is $K=4$. Figure 1 above shows the classification of 28 data points in each cluster, and the number of data points in red font is the cluster center of each cluster.

Experimental Verification and Analysis of Clustering Results

We set the cluster number $K=4$ as the $K$ value input of the traditional K-means clustering algorithm, and cluster the data sets of the 28 data points on the MATLAB software platform. The result of clustering center and clustering is shown in Figure 2.
Among them, the first category contains point number: 1, 5, 10, 14, 20, 23, 27; 
The second category contains point number: 2, 6, 11, 15, 22; 
The third category contains point number: 3, 7, 9, 12, 17, 25, 26, 28; 
The fourth category contains point number: 4, 8, 13, 16, 18, 19, 21, 24.

In summary, for the same data set, the clustering results obtained by the improved algorithm are the same as those obtained by using the traditional K-means clustering algorithm directly on the software MATLAB. The number of clusters obtained by the improved algorithm is used as the K-value input of the traditional K-means clustering algorithm. The clustering effect is very good and meets the practical requirements. The feasibility of the improved algorithm can be verified. The problem that the clustering results are unstable due to the manual input of K value in the traditional K-means clustering algorithm is well solved, and the problem that noise affects the efficiency of the algorithm is also avoided.

Conclusion

In order to solve the problem of selecting K value in the traditional K-means clustering algorithm, this paper presents a method to automatically obtain K value based on the improved K-means algorithm. Through the simulation analysis, we can see that the improved clustering algorithm is used to cluster 28 data points, and the number of clusters K is used as the input value of the traditional K-means clustering algorithm. The clustering results are obtained by MATLAB simulation. Good clustering effect can be obtained, which accords with the actual requirements, and provides a new idea for the selection of K-value in clustering algorithm to some extent, and has certain engineering application value. To some extent, the improved algorithm avoids the initial clustering center being misselected in the same class, and overcomes the problem that the result of the algorithm falls into a local optimal solution and depends on K value. When the amount of data is very large, the algorithm is inefficient because it has to calculate the distance frequently to select the appropriate cluster center. How to improve the accuracy and efficiency of the K-value automatic generation algorithm based on the principle of distance maximization is the main research direction in the next step.

References


