An initial clustering center optimization method based on neighbourhood density for K-means

Mingxue Luo, Yingchun Yuan and Kejian Wang
College of Information and Technology, Hebei Agricultural University, Baoding, Hebei, 071000, China
*Corresponding author’s e-mail: yych@hebau.edu.cn

Abstract. The selection of initial clustering centers of traditional K-means algorithm is random and sensitive to outliers, which leads to unstable clustering results and low accuracy. To solve the above problems, the NDK-means algorithm based on neighbourhood density is proposed. Firstly, the grid distribution characteristics of samples are obtained by multi-dimensional grid division. Then, by defining the grid density and the grid neighbourhood density, several local high-density grids are determined. At the same time, the iteration factor is introduced to merge adjacent high-density grids to obtain a candidate set of initial clustering centers. Finally, combined with grid density and distance, K initial clustering centers are obtained by using Max-Min-distance algorithm. Experiments on UCI dataset show that compared with k-means algorithm and literature algorithm, the accuracy of NDK-means algorithm is improved by nearly 11% and 4%, and the iteration speed is improved by 70% and 60% respectively. The algorithm improves the accuracy of clustering and the results have good stability.

1. Introduction

The advent of the era of big data has produced high-dimensional and large-scale data. Clustering analysis of massive data is an important direction of current research, and has been widely used in education, e-commerce, transportation and other fields. Clustering is to divide data objects into different clusters according to the idea of "things-of-one-kind-come-together", so that the data similarity within the cluster is high, while the data similarity among clusters is low. Clustering algorithms usually include partition-based method, density-based method, model-based method, graph theory-based method and so on.

K-means is a classical partition-based method. It is widely used because of its high-efficiency, easy understanding and so on. However, due to the random selection of the initial clustering centers, it is easy to cause the clustering results to fall into local optimum. At well, it makes the clustering results unstable.

2. Related research

Many scholars have done a lot of research on the selection of initial clustering centers of k-means algorithm. In reference, by introducing swarm intelligence algorithm, self-adaptive cuckoo algorithm and gravity search algorithm are used to optimize the initial clustering centers. However, considering the computational complexity of the algorithms, the algorithms have not been applied. Some scholars have also optimized the initial clustering centers based on sample data density and distance. For example, Tang Zekun considered about the impact of density and distance on clustering, they weighted the data for selecting the initial center points. However, the calculation of...
data weight increases the time consumption. Tang Dongkai [8] proposed LOF algorithm, it calculated the outlier factor of sample points based on density, and sorted sample points in descending order by the outlier factors. Then the algorithm selected the lower outlier factor as the initial cluster centers. However, it is ignored that low-density clusters are regarded as outliers, resulting in inaccurate selection of clustering centers. The LOF algorithm also increases time consumption. In reference [9], the idea of grid is introduced. According to the number of cluster types K, the sample data is mapped to a K-dimensional grid, and then K sub-grids, which have more samples and satisfy the defined distance, are selected as initial clusters. However, it only considers the density of the grid itself. At the same time, the number of candidate sets is too large to improve effectively the accuracy.

The conclusion based on the above exiting algorithm shows that: 1) the optimization algorithm based on density and distance has a good effect on the selection of initial clustering, but it consumes a lot of time to calculate the density of each sample; 2) the algorithm based on grid division can greatly reduce the calculation time consumption, but it does not filter the grid, which leads to the deviation in the selection of clustering centers when using distance calculation. To solve the above problems, this paper proposes an initial clustering centers optimization method based on neighbourhood density for K-means, namely NDK-means (optimization method of K-means initial clustering centers based on neighbourhood density) algorithm.

3. NDK-means algorithm

Based on K-means algorithm, NDK-means algorithm improves the selection of initial clustering centers based on local density and distance. The NDK-means algorithm divides the data set based on the idea of grid. By defining the grid neighbourhood and grid density, high-density grids are introduced, and they are selected as initial candidate clustering centers. Then, we expand the grid size step by step to merge adjacent centers in the candidate sets. On the one hand, the method can avoid the influence of the fixed grid size on the selection of the high-density grids, on the other hand, it also reduces the number of the candidate set. Finally, K initial clustering centers are selected based on the distance.

3.1. Related definitions

The NDK-means algorithm divides the data set based on the idea of grid. The algorithm selects the candidate grids through the spatial neighbourhood. The relevant definitions are as follows:

Definition 1. (grid neighbourhood) for any grid $S_j$, the set of adjacent grids is called the neighbourhood of $S_j$, denoted as $Fe(S_j)$. Figure 1 shows a schematic diagram of two-dimensional mesh generation. The grid neighbourhood of $S_i$ in the figure is $Fe(S_i) = \{S_{i1}, S_{i2}, S_{i3}, S_{i4}, S_{i5}, S_{i6}, S_{i7}, S_{i8}\}$, with a total of eight adjacent grid elements[10].

![Figure 1 Grid neighbourhood diagram](image)

Definition 2. (grid density) the number of sample points in grid $S_i$ is called grid density, which is denoted as $DS_i$.

Definition 3. (neighbourhood density) for any grid $S_i$, the mean value of all grid neighbourhood densities is called neighbourhood density, denoted as $SK_i$. The formula is as follows:

$$SK_i = \frac{\sum_{j=1}^{n} DS_{ij}}{n}$$

(1)

Where $DS_{ij}$ is the density of the j-th neighborhood of grid $S_i$ and n is the number of neighbourhood.
Generally, the clustering center is located in the high-density area of the cluster. The neighbourhood density reflects the degree of sample aggregation of adjacent grids. By comparing the grid density to its neighbourhood density, we can determine whether the grid is a high-density grid and whether it is a candidate grid for the initial clustering centers.

Definition 4. (high density grid) for any grid $S_i$, if its density is greater than its neighbourhood density, then $S_i$ is called high density grid.

Let $SRK_i$ be denoted as the ratio of the grid density of $S_i$ to its neighbourhood density, then the formula is as follows:

$$SRK_i = \frac{DS_i}{SK_i}$$  \hspace{1cm} (2)

If $SRK_i > 1$, the grid $S_i$ belongs to high density grid.

It is hard to set the fixed threshold of high-density grid to meet the requirements of dense and sparse clusters at the same time. In this paper, the relationship between the grid and its neighbourhood density is used to determine the high-density grid, which reduces the complexity of threshold setting.

3.2. Algorithm description

The initial clustering centers selection of NDK-means algorithm is mainly divided into three steps: 1) dividing the multi-dimensional grid space according to the attributes of the data set; 2) determining the candidate grid set of the initial clustering centers; 3) determining the initial clustering centers based on the Max-Min-distance algorithm.

3.2.1. Dividing multidimensional grid space

The key of mapping data set to multi-dimensional grid is actually to determine the grid size. If the grid division is large, the data from different categories will be divided into the same grid, and the cluster center cannot be selected normally; otherwise, if the grid division is small, it will lead to too many grids. The grid density is similar, so the outlier data cannot be eliminated according to the density, and the calculation complexity is increased. Based on the grid partition function proposed in reference [10], the final partition function is shown in equation (3)

$$M = 0.8 \left( \frac{1}{N^3 + F^2} \right)$$  \hspace{1cm} (3)

Where $N$ is the size of the dataset, $F$ is the number of attributes, and $M$ is the number of each dimension grid of the multidimensional grid.

3.2.2. Determine the candidate grid set of initial clustering center points

After dividing the multi-dimensional grids, the candidate set can be obtained by high-density grid. Multiple candidate grids may be from the same cluster, so the merging idea is proposed. By increasing the grid size step by step, the adjacent grids in the same cluster are merged until the quantity is satisfied. The steps to determine the initial clustering center candidate set are as follows.

1) According to the formula (3) in section 3.2.1, the multi-dimensional grid is divided and the neighborhood density $DS_i$ of the grid is calculated.

2) According to the formula (1) and (2), the high-density grid is obtained. The candidate grid set is obtained.

3) The center of each candidate grid is calculated to form set DR. It is as follows:

For any grid $S_i$, it contains samples $x = \{x_{i1}, x_{i2}, ..., x_{ip}\}$, where $p$ is the number of sample points in the grid. The mean value of all sample attributes in the grid is the center of the grid, which is recorded as $sc_j$. The formula is as follows:

$$sc_j = \frac{x_{i1} + x_{i2} + ... + x_{ip}}{p}$$  \hspace{1cm} (4)
4) If the number of the candidate set is not less than the iteration factor, the adjacent grids in the candidate set are merged. The number of grids is reduced by \( M = M-1 \), and a new grid partition is obtained. The grid centers are calculated to form a new set of candidate centers. The iteration factor is calculated as follows:

\[
\alpha = 2 \times k
\]

Where \( k \) is the number of cluster species.

5) Repeat step 4 until the iteration factor is satisfied. The final candidate center set \( DR \) is obtained.

3.2.3. Determine the initial clustering centers
Since the closer high-density grids may come from the same cluster, we select the center based on the idea of the Max-Min-distance algorithm. Firstly, the center of the grid with the highest density is selected as the first initial cluster center, and then the remaining cluster centers are selected according to the Max-Min-distance algorithm, which eliminates the randomness of the initial cluster center and has stable accuracy. The algorithm is described as follows:

- **Input**: candidate set \( DR \), the number of clustering \( k \)
- **Output**: \( k \) initial clustering centers
- **Step 1.** Select the first element in \( DR \) as the initial cluster center \( C_1 \), add it to the cluster center set, then \( C = \{C_1 \} \). Delete the first element in \( DR \).
- **Step 2.** According to the principle of Max-Min-distance algorithm, calculate the i-th initial center \( C_i \), add it to the set \( C = \{C_1, ..., C_i \} \), delete the element in \( DR \);
- **Step 3.** When the length of set \( C \) is less than \( k \), return to Step 2;
- **Step 4.** Output \( C \).

3.3. NDK-means algorithm
The overall process of the NDK-means algorithm is as follows:

- **Input**: data set \( D \), the number of clusters \( k \)
- **Output**: \( k \) clusters
- **Step 1.** For the data objects in data set \( D \), divide the multidimensional grid according to the grid division formula, and calculate the grid density.
- **Step 2.** According to the grid density and the grid neighbourhood density, the initial candidate set is determined. By continuously merging, the final candidate set is obtained.
- **Step 3.** On the candidate data sets, \( k \) initial clustering centers are selected by using the Max-Min-distance algorithm.
- **Step 4.** Use the initial cluster centers, execute the K-means algorithm to obtain the clustering result. The process is shown in Figure 2.
4. Results & Discussion

In order to fully verify the effect of the algorithm in this paper, the experiment is divided into two parts: the first part uses simulation data to visually compare the traditional K-means algorithm with the NDK-means algorithm; the second part uses three real data from UCI laboratory Set, compare the traditional K-means algorithm, the algorithm in reference[9] and the algorithm in this paper. The experimental environment configuration is: CPU is Intel Core i7-4510U, memory is 8G, hard disk is 1T, and Windows 10 operating system. All algorithms are implemented using Python 3.2.

4.1. Simulation data set

The simulation data is generated by make_blobs module in python. The data set has 4 categories, and each category contains 100 two-dimensional data points. Clustering experiments are performed on the K-means algorithm and the NDK-means algorithm. After 50 experiments for each algorithm, two groups are randomly selected from the clustering results.

Figure 3 shows the two experimental results of the traditional K-means algorithm. Figure 4 shows the two experimental results of the algorithm in this paper. The triangle represents the actual clustering centers, and the cross represents the initial clustering centers. It can be seen that the traditional K-means algorithm selects the initial clustering centers randomly, and the number of iterations is 7 and 4 times respectively, and the number of iterations and clustering results is unstable. The number of iterations of NDK-means algorithm is 2 times. This is because the clustering centers selected by the improved algorithm is close to the actual clustering centers, the number of iterations is small and the clustering result is stable.
4.2. Standard data set

In order to verify the NDK-means algorithm, this paper selects the UCI database specially used to test the performance of the clustering algorithm. The UCI database is proposed by University of California. It is a commonly used standard test data set for machine learning. Each data set has a clear classification, and the clustering can be directly obtained. The experiment tests three data sets of Iris, Seeds, and New_thy, each algorithm runs 20 times on the three test sets, and record all the results. The specific information description of the experimental data set is shown in Table 1.

| data set  | Number of samples | Number of attributes | Number of categories |
|-----------|-------------------|----------------------|----------------------|
| Iris      | 150               | 4                    | 3                    |
| Seeds     | 210               | 7                    | 3                    |
| New_thy   | 215               | 5                    | 3                    |

4.2.1. Accuracy

Table 2 shows the accuracy of the three algorithms on different data sets. It can be obtained that the accuracy of the NDK-means algorithm on the test data set is higher than the traditional K-means and the reference[9] algorithm. The accuracy of the algorithm in this paper is higher 10.6%, 10.7%, and 13% than the original K-means algorithm on the Iris, Seeds, and New_thy datasets respectively. This is because we use the grid neighbourhood density to filter the candidate center set. Thus, the selected clustering centers are usually located in the high-density area, which can reduce those outliers in the data set as the initial clustering centers. As well, NDK-means algorithm uses the Max-Min-distance
algorithm to select the final clustering centers to avoid the similar initial clustering centers. Secondly, it can be found which the clustering results based on K-means algorithm are unstable, the difference between the best result and the worst result is 10%-27%, this is because the traditional K-means algorithm randomly selects the initial clustering centers. At the same time, it can be seen that the accuracy of the algorithm in this paper is evenly 4.3% higher than the algorithm in the reference [9] on the three data sets. Especially, the proposed algorithm is 10% higher than that in the reference [9] on data set New_thy. This is because individual outliers are far away from other data points in the New_thy data set. Although the algorithm in reference [9] also optimizes initial clustering based on grid division, the number of grids is determined by clusters K. the fixed grid size is sensitive to outlier data and also affects the final clustering results. NDK-means algorithm selected the initial clustering center based on local density, it may avoid the influence of outliers, so it can obtain higher accuracy and the clustering results are stable.

Table 2. The accuracy of the three algorithms on different data sets

| algorithms          | Accuracy       |         |         | Iris Seeds New_thy |         |         |         |         |         |         |         |         |         |         |         |
|---------------------|----------------|---------|---------|-------------------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
|                     | Max     | Min     | AVG     | Max     | Min     | AVG     | Max     | Min     | AVG     | Max     | Min     | AVG     | Max     | Min     | AVG     |
| K-means             | 90.0%   | 66.7%   | 80.1%   | 89.5%   | 61.9%   | 78.8%   | 78.6%   | 69.8%   | 73.5%   |         |         |         |         |         |         |
| reference [9]       | 89.3%   | 89.3%   | 89.3%   | 89.1%   | 89.1%   | 89.1%   | 75.4%   | 75.4%   | 75.4%   |         |         |         |         |         |         |
| NDK-means           | 90.7%   | 90.7%   | 90.7%   | 89.5%   | 89.5%   | 89.5%   | 86.5%   | 86.5%   | 86.5%   |         |         |         |         |         |         |

4.2.2. Number of iterations

The number of iterations indicates the convergence speed of the algorithm. The faster the convergence speed is, the closer the initial clustering centers is to the real clustering centers. Therefore, the iterative improvement rate is defined as

\[
IR_1 = \frac{KL - NI}{KL} \times 100\%
\]

(6)

\[
IR_2 = \frac{RI - NI}{RI} \times 100\%
\]

(7)

Among them, \(KL\) is the number of iterations of the k-means, \(NI\) is the number of iterations of the NDK-means, \(RI\) is the number of iterations of the reference [9] algorithm.

Table 3 shows the comparison of iteration times of three algorithms on different data sets. It can be seen that from Table 3 that the number of iterations of NDK-means algorithm on the three data sets is far less than other algorithms. Among them, the iterative efficiency of NDK-means algorithm is nearly 70% higher than that of k-means algorithm, and nearly 60% higher than that of the reference [9]. On the data set New_thy, the iterative efficiency improved 75% and 71% respectively. It reflects the advantages of the algorithm in this paper for sparse data sets. It shows that the NDK-means algorithm accelerates the clustering iteration process and improves the clustering efficiency.

Table 3. Number of iterations of the three algorithms on different data sets

| Data Set      | K-means | reference [9] | NDK-means | \(IR_1\) | \(IR_2\) |
|---------------|---------|----------------|-----------|---------|---------|
| Iris          | 7       | 4              | 2         | 71.4%   | 50.0%   |
| Seeds         | 8       | 6              | 3         | 62.8%   | 50.0%   |
| New_thy       | 16      | 14             | 4         | 75.0%   | 71.4%   |
| AVG           |         |                |           | 69.7%   | 57.1%   |
5. Conclusions
Aiming at the problem of random selection of the initial clustering centers of the K-means algorithm, which easily leads to local optimum, unstable clustering results, and sensitivity to outliers, an improved K-means algorithm based on neighbourhood density is proposed. It divided sample space by grid in reference [9], and proposed a novel candidate set selection strategy based on high density grid. At the same time, it merges grids by expanding the grid size to reduce the number of clustering centers and improve the accuracy of initial clustering centers. Experiments on UCI dataset show that the accuracy of the proposed algorithm is improved by 10% and 4% respectively compared with k-means algorithm and reference [9] algorithm. At the same time, the iteration rate is greatly improved by 70% and 60% respectively, so the clustering results of NDK-means have better accuracy, efficiency and stability.

Acknowledgments
The Hebei Province Higher Education Teaching Reform Research and Practice Project "Research on the Confidence of College Students' Evaluation of Teaching in the Context of Big Data" (2018GJJG140).

References
[1] Zhou R W, Li Z Y, Chen S M, et al. (2016) Parallel optimization sampling clustering K-means algorithm for big data processing [J]. Journal of Computer Applications,36(02):311-315+329.
[2] Du J Y, Duan L Z, Duan W Y, et al. (2020) Parallel implementation of improved K-means algorithm based on Spark [J]. Application Research of Computers, 37(02):434-436+497.
[3] Wan J, Zhang Y, He Y B, et al. (2015) Dynamic clustering algorithm based on KD-tree and K-means method [J]. Application Research of Computers,32(12):3590-3595.
[4] Zhou B J, Tao Y Z, Ji B, et al. (2018) Optimizing k-means initial clustering centers by minimizing sum of squared error[J]. Computer Engineering and Applications, 54(15):48-52.
[5] Yang H H, Wang K, Li L Q, et al. (2016) K-means clustering algorithm based on adaptive cuckoo search and its application [J]. Journal of Computer Applications,36(08):2066-2070.
[6] Wei K Y, He Q, Xu Q S. (2019) Novel K-means clustering algorithm based on improved gravitational search algorithm [J]. Application Research of Computers,36(11):3240-3244.
[7] Tang Z K, Zhu Z Y, Yang Y, et al. (2020) D-K-means algorithm based on distance and density [J]. Application Research of Computers, 37(06):1719-1723.
[8] Tang D K, Wang H M, Hu M, et al. (2018) Optimizing Initial Cluster Center of Improved K-means Algorithm [J]. Journal of Chinese Computer Systems,39(08):1819-1823.
[9] Shao L, Zhou X Z, Zhao C P, et al. (2018) Improved K-means clustering algorithm based on multi-dimensional grid space [J]. Journal of Computer Applications,38(10):2850-2855.
[10] Wang L Y, Shi L, Yi J, et al. (2020) NLOF: two-stage outlier detection algorithm based on grid filtering [J]. Application Research of Computers, 37(04):990-993+998.