Improved K-Means Algorithm and Its Application to Vehicle Steering Identification

Hui Qi[1,2](✉), Xiaoqiang Di[1,2], Jinqing Li[2], and Hongxin Ma[3]

[1] National and Local Joint Engineering Research Center of Space and Optoelectronics Technology, Changchun University of Science and Technology, Changchun, China
qihui@cust.edu.cn
[2] School of Computer Science and Technology, Changchun University of Science and Technology, Changchun, China
[3] Training Department, Aviation University Air Force, Changchun, China

Abstract. K-means is a very common clustering algorithm, whose performance depends largely on the initially selected cluster center. The K-means algorithm proposed by this paper uses a new strategy to select the initial cluster center. It works by calculating the minimum and maximum distances from data to the origin, dividing this range into several equal ranges, and then adjusting every range according to the data distribution to equate the number of data contained in the ranges as much as possible, and finally calculating the average of data in every range and taking it as initial cluster center. The theoretical analysis shows that despite linear time complexity of initialization process, this algorithm has the features of an superlinear initialization method. The application of this algorithm to the analysis of GPS data when vehicle is moving shows that it can effectively increase the clustering speed and finally achieve better vehicle steering identification.

Keywords: K-means · Clustering · Vehicle steering
Vehicle navigation system

An intuitional objective function of clustering algorithms in common use is the Sum of Squares for Error (SSE), which is provided below:

$$SSE = \sum_{i=1}^{K} \sum_{\mathbf{x_j} \in P_i} \|\mathbf{x_j} - \mathbf{c_i}\|_2^2 \tag{1}$$

where: K is the number of clusters, $\mathbf{x_j}$ is the jth datum of this data set, P_i is the ith cluster, $\mathbf{c_i}$ is the center of the ith cluster ($\mathbf{c_i} = 1/|P_i| \sum_{\mathbf{x_j} \in P_i} \mathbf{x_j}$, where $|P_i|$ is the number of data in the ith cluster), $\|.\|_2$ is Euclidean distance. A clustering algorithm is aimed to find the minimum SSE. But because this non-convex optimization is NP-hard [1,2], its approximate solution in polynomial time can only be found at present. K-means algorithm is just such a clustering algorithm. It has

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2018
G. Sun and S. Liu (Eds.): ADHIP 2017, LNICST 219, pp. 378–386, 2018.
https://doi.org/10.1007/978-3-319-73317-3_44

been widely used, as its concept is simple and easy to implement. For instance, K-means algorithm is used in [3] to cluster the GPS data during vehicle driving for identifying whether the vehicle is making a turn or not, and finally to build a learning system of vehicle steering identification based on the architecture of dynamic onboard navigation system. This system sends the GPS data collected by client (onboard terminal) to the server, which, in turn, automatically calculates the steering identification model applicable to the vehicle and returns the model parameters to the client. As the server needs to create a model for many clients, the modeling speed has become an issue of great concern during the server programming, which would influence the server's quality and capability. This paper optimizes the first step of modeling, namely K-means clustering, in order to increase the rate of convergence.

The rest of this paper is organized as follows. Section 1 gives a brief introduction to the research of the initialization of K-means clustering algorithm. Section 2 presents the improved initialization method and the performance it achieves. In Sect. 3 three initialization methods are compared using the field test data and the test results are analyzed. Section 4 concludes this paper.

1 Related Work

K-means clustering algorithm is implemented through two steps: initialization and subsequent iterations. Initialization is to select the initial cluster center, namely $\mathbf{c_i}$ of the first iteration, while subsequent iterations are to continuously change the cluster center until it won't change any more or the number of iterations reaches its maximum. As pointed out by [4], K-means clustering algorithm is so sensitive to the cluster center selected during initialization that the selection of a different initial cluster center will influence the algorithm performance. Whats more, improper initialization may result in empty clustering, slower convergence and a higher risk of being caught in the locally optimal solution [5]. Therefore, improving the initialization process has become an important means of K-means performance improvement. In the [4], various initialization methods are analyzed and divided into two categories: linear time complexity and ultralinear time complexity. The linear method is often non-deterministic or sensitive to sequence [6], while the superlinear method is usually deterministic. In other words, by clustering the same data set repeatedly with the K-means algorithm based on linear initialization, different clustering results will be obtained; by clustering the same data set with the K-means algorithm based on superlinear initialization, only one clustering result will be obtained, no matter how many times the data set is clustered. Therefore, with the superlinear method, only one clustering, rather than repeated clustering to select the optimal clustering result, is needed. Besides, the superlinear method often enables fast convergence of k-means algorithm and applies to the clustering of a large data set. It is just these advantages that attract extensive attention to the superlinear method. For example, in the [7], a variance-based method is proposed to sequence all the data according to the attribute with the maximum variance, then to divide the sorted data into K groups, and finally to choose the middle datum in every group as initial cluster center. In the [8], the kd-tree of data points is built for density estimation, and then the modified maximin method is used to select K cluster centers from the densely generated leaves. In the [9], a robust initialization method is proposed to use a local outlier factor that can prevent an abnormal datum from being taken as cluster center. In the [10], an initialization method with k iterations is proposed to at first establish k sets and then during the ith $(1 \leq i \leq k)$ iteration, to channel the nearest data pairs from the data sets into the ith set continuously until the number of data in the set exceeds a certain threshold, suggesting the end of the ith iteration and the start of the $(\mathrm{i}+1)$th iteration. In the [11], a method based on attribute transformation is proposed to at first change the negative attribute of all the data into positive, then to sequence all the changed data according to their distances to the origin and divide the sorted data into K groups, and finally to choose the middle datum in every group as initial cluster center. The idea of [12] is similar to that of [11], with the exception of using the averages to choose the cluster center. The time complexity of all the above superlinear methods is $O(n \log n)$, except for that in the [10], where the time complexity is $O(n^2)$.

2 Improved Initialization Method

This paper proposes an improved initialization method that uses the ideas of [11,12] for reference and needs to change the negative attribute of all the data in a way shown in [11,12].

After changing the attribute, the calculation of the distances from data to the origin is also needed. But next, unlike the methods in [11,12], the proposed method no longer needs to sequence all the data according to their distances to the origin, but to choose the minimum (d_{min}) and maximum (d_{max}) distances. The time complexity of this step is $O(n)$.

Next, divide the range $[d_{min}, d_{max}]$ into K subranges evenly, each with the following interval:

$$interval = \frac{d_{max} - d_{min}}{K}$$

The range of the ith subrange $(1 \leq i \leq K)$ is $[d_{i,min}, d_{i,max}]$, where:

$$d_{i,min} = d_{min} + (i-1) \times interval$$

$$d_{i,max} = d_{min} + i \times interval$$

Then group all the data by subrange in the following way. Suppose d_j is the distance from the datum $\mathbf{x_j}$ to the origin, then $\mathbf{x_j}$ is in the range i if $d_{i,min} \leq d_j \leq d_{i,max}$. During the data grouping, the total of data c_i in every subrange is also counted. The time complexity of this step is $O(n)$.

Next, adjust the range of every subrange. The reason for implementing this step is that the data may be distributed among various subranges so unevenly and differently that the ranges will be empty or composed of abnormal data to finally affect the clustering performance. The method of subrange adjustment is as follows:

Step 1: Define the variables i and p_i, and initialize i as 1 and p_i as 0.
Step 2: If i = K, end the subrange adjustment; otherwise, go to the step 3.
Step 3: Suppose $p_i = p_i + c_i$, $p = p_i + c_{i+1}$, $p_1 = p_i/p$, $p_2 = c_{i+1}/p$, $l_1 = i/(i+1)$ and $l_2 = 1/(i+1)$. To better describe the process of subrange adjustment, the range i and the pre-i ranges may be called by a joint name "pre-i ranges". Then p_i is the total of data in the pre-i ranges, and p is the total of data in the pre-i + 1 ranges (or the total of current data). p_1 and p_2 are the ratios between data totals: p_1 is the ratio of the data total of pre-i ranges to current data total, and p_2 is the ratio of the data total in the range i + 1 to current data total. By the same token, l_1 and l_2 are the ratios between range lengths: l_1 is the ratio of the total length of pre-i ranges to that of current ranges, and l_2 is the ratio of the length of range i + 1 to that of current ranges.
Step 4: If $p_1 > l_1$, it means the data density in the pre-i ranges is bigger than that in the range i + 1 so that the pre-i ranges need to be scaled down by $dl = ((p_1 - l_1)/p_1) \times l1$; otherwise, the data density in the pre-i ranges is smaller than that in the range i + 1 so that the pre-i ranges need to be scaled up by $dl = ((p1 - l_1)/p_2) \times l2/l1$.
Step 5: Calculate $d_{j,max} = d_{j,max} - d_{j,max} \times dl$ for every pre-i range, where there is $1 \leq j \leq i$.
Step 6: Suppose i = i + 1. Then go to the step 2.

The time complexity of subrange adjustment is $O(K^2)$.

Regroup the data by using new subranges, and calculate the average of every group of data, which is just the initial cluster center. The time complexity of this step is $O(n)$.

Here the proposed initialization method comes to an end. Next is the subsequent iterations of K-means algorithm. The total time complexity of this initialization is $O(3n + K^2)$, which is actually linear $O(n)$, as K is a constant and $K \ll n$. But the method proposed by this paper features superlinear initialization rather than linear initialization. In other words, this method is deterministic, because no matter how many times the method is executed, the ranges for the same data set remain unchanged, so does the final clustering result.

The core of the proposed initialization method is subrange adjustment, whose aim is to enable uniform distribution of data in every subrange. This method applies to continuously distributed data, such as the data in [3], as the GPS direction during driving often changes continuously.

The algorithm in this paper, the algorithms in [11,12], and the K-means algorithm based on random initialization are used to cluster one data set in [3] respectively. Suppose m = 4 and K = 4. The learning curve shown in Fig. 1, where the vertical axis is SSE value, can be obtained. In the Fig. 1, "range" is the algorithm in this paper, "median" is the algorithm in [11], "mean" is the algorithm in [12], and "random" is the algorithm based on random initialization. It can be obviously seen from the figure that, the algorithm in this paper converges

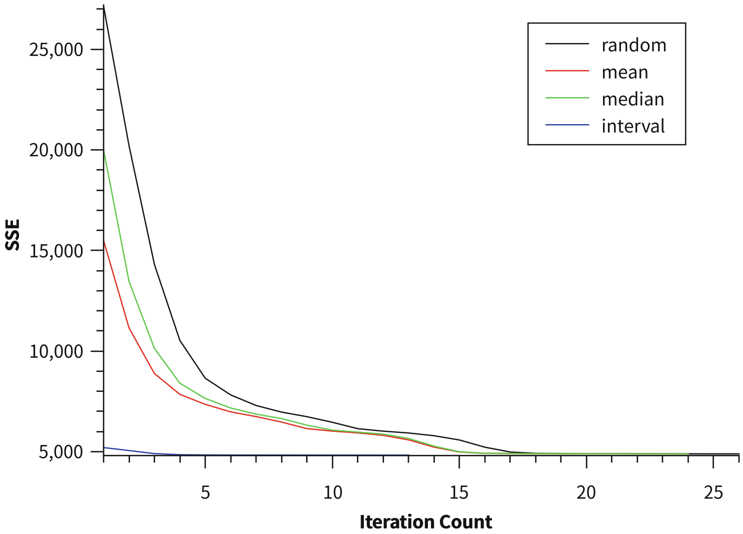

Fig. 1. Learning curves of four K-means clustering algorithms

fastest. In fact, it is iterated for 13 times, the algorithms in [11,12] for 24 times respectively, and the algorithm based on random initialization for 26 times. In addition, the final SSE is 4822.19 in the proposed algorithm and 4884.18 in the other three algorithms respectively.

3 Analysis of Experimental Results

The proposed algorithm can apply to the learning system of vehicle steering identification designed in [3] in order to speed up the identification modeling. To verify the actual application effect of the algorithm, this section introduces it into the learning system designed in [3] and through an experiment, evaluates the performance of the finally generated identification model as well as the execution speed of the algorithm.

The experiment uses the data in [3] for testing. The data are contained in two data sets, each sampled at a frequency of 1 Hz. The sampling mileage of data set 1 is 18.23 km, covering 2960 GPS points; whereas the sampling mileage of data set 2 is 11.58 km, covering 2370 GPS points.

The comparison objects in the experiment include the algorithm in this paper, the algorithms in [11,12], and the K-means algorithm based on random initialization. The comparison indicators include F_1 and the number of subsequent iterations of K-means algorithm, with the former reflecting the performance of identification model and the latter indirectly showing the speed of K-means clustering (i.e. the execution speed of the algorithm).

By testing the data set 1 with the four algorithms respectively, the results in Tables 1 and 2 can be obtained. It is observed from Table 1 that, the proposed

Table 1. F_1 values obtained from testing the data set 1 with the four algorithms

m	K	Random	Algorithm in [11]	Algorithm in [12]	Our algorithm	The best
3	4	0.86792	0.86792	0.86792	0.87711	Our algorithm
3	5	0.78226	0.78226	0.78226	0.87097	Our algorithm
4	4	0.89164	0.92141	0.92141	0.93175	Our algorithm
4	5	0.94461	0.89710	0.89710	0.94461	Our algorithm
5	4	0.93421	0.93421	0.93421	0.80000	Other algorithms
5	5	0.93421	0.95484	0.95484	0.93421	Other algorithms
6	4	0.80000	0.80000	0.80000	0.76316	Other algorithms
6	5	0.95971	0.95971	0.95971	0.82988	Other algorithms

Table 2. Number of subsequent iterations when clustering the data set 1 with the four algorithms

m	K	Random	Algorithm in [11]	Algorithm in [12]	Our algorithm	The best
3	4	26	28	27	18	Our algorithm
3	5	42	41	41	12	Our algorithm
4	4	26	24	24	13	Our algorithm
4	5	36	40	39	12	Our algorithm
5	4	19	21	20	6	Our algorithm
5	5	39	40	39	6	Our algorithm
6	4	23	27	26	5	Our algorithm
6	5	27	29	28	12	Our algorithm

algorithm performs best in 4 of all the 8 models. The average F_1 of the 4 models is 0.90611, 0.02706 higher than the algorithm in the second place; while the average F_1 of the other 4 models is 0.83181, 0.08038 lower than the algorithm in the first place. Besides, when $\mathrm{m} = 4$ or $\mathrm{m} = 5$, the F_1 values of optimal models are all greater than 0.9 and average 0.93941. It can be seen from the Table 2 that, the proposed algorithm is executed much faster and all the models are executed fastest, with 18.9 (or 64.3%) iterations fewer than the algorithm in the second place on average.

By testing the data set 2 with the four algorithms respectively, the results in Tables 3 and 4 can be obtained. It is observed from Table 3 that, the proposed algorithm performs best in 5 of all the 8 models. The average F_1 of the 5 models is 0.93818, 0.07999 higher than the algorithm in the second place; while the average F_1 of the other 3 models is 0.93351, 0.01947 lower than the algorithm in the first place. Besides, when $\mathrm{m} = 4$ or $\mathrm{m} = 5$, the F_1 values of optimal models are all greater than 0.9 and average 0.95117. It can be seen from the Table 4 that,

Table 3. F_1 values obtained from testing the data set 2 with the four algorithms

m	K	Random	Algorithm in [11]	Algorithm in [12]	Our algorithm	The best
3	4	0.84685	0.84685	0.84685	0.90716	Our algorithm
3	5	0.79832	0.79832	0.79832	0.90765	Our algorithm
4	4	0.93617	0.93617	0.93617	0.93293	Other algorithms
4	5	0.85787	0.85787	0.85787	0.94260	Our algorithm
5	4	0.96689	0.96689	0.96689	0.93426	Other algorithms
5	5	0.94631	0.94631	0.94631	0.95973	Our algorithm
6	4	0.92913	0.95588	0.95588	0.93333	Other algorithms
6	5	0.97358	0.97358	0.97358	0.97378	Our algorithm

Table 4. Number of subsequent iterations when clustering the data set 2 with the four algorithms

m	K	Random	Algorithm in [11]	Algorithm in [12]	Our algorithm	The best
3	4	40	40	40	8	Our algorithm
3	5	28	39	39	6	Our algorithm
4	4	19	28	27	5	Our algorithm
4	5	26	38	38	11	Our algorithm
5	4	15	19	19	12	Our algorithm
5	5	16	18	28	10	Our algorithm
6	4	26	18	18	5	Our algorithm
6	5	19	23	23	14	Our algorithm

the proposed algorithm is executed much faster and all the models are executed fastest, with 16.5 (or 62.8%) iterations fewer than the algorithm in the second place on average.

It is observed from the above two groups of test results that, the algorithm proposed by this paper performs best in 9 of all the 16 models. The average F_1 of the 9 models is 0.92393, 0.0418 higher than the algorithm in the second place; while the average F_1 of the other 7 models is 0.87540, 0.05428 lower than the algorithm in the first place. For the m value commonly used in practical application (m = 4 or m = 5), the average F_1 of its optimal models is 0.94529. Moreover, the subsequent iterations of K-means clustering based on the proposed algorithm are significantly reduced, with 16.3 (or 62.7%) iterations fewer than the algorithm in the second place on average. It is thus clear that, the identification model built upon the clustering algorithm proposed by this paper performs basically as well as the other 3 algorithms, while the common identification models using this algorithm perform slightly better but much faster.

4 Conclusion

This paper improves the initialization process of K-means clustering algorithm to effectively reduce subsequent iterations without compromising the clustering performance, which makes it suitable for large-scale data clustering [13,14]. The application of this algorithm to the learning system of vehicle steering identification can speed up the modeling of steering identification and guarantee the performance of identification model. The core concept of this algorithm is to calculate the value range of a data set in a certain aspect and then to reasonably group the data in this range in order to choose the initial cluster center. This paper uses the distances from data to the origin as the criterion of data division, which, in practical use, may be one dimension of those data as well. The selection of this criterion depends mainly on data distribution - an area to be explored more deeply.

Acknowledgment. This work is supported in part by the National High Technology Research and Development Program (863 Program) of China under Grant No. 2015AA015701, the Science and Technology Planning Project of Jilin Province under Grant No. 20150204081GX.

References

1. Aloise, D., Deshpande, A., Hansen, P., Popat, P.: NP-hardness of Euclidean sum-of-squares clustering. Mach. Learn. **75**(2), 245–248 (2009)
2. Mahajan, M., Nimbhorkar, P., Varadarajan, K.: The planar-means problem is NP-hard. Theoret. Comput. Sci. **442**, 13–21 (2012)
3. Qi, H., Liu, Y., Wei, D.: GPS-based vehicle moving state recognition method and its applications on dynamic in-car navigation systems. In: 2014 IEEE 12th International Conference on Dependable, Autonomic and Secure Computing, pp. 354–360 (2014)
4. Celebi, M.E., Kingravi, H.A., Vela, P.A.: A comparative study of efficient initialization methods for the K-means clustering algorithm. Expert Syst. Appl. **40**(1), 200–210 (2013)
5. Celebi, M.E.: Improving the performance of K-means for color quantization. Image Vis. Comput. **29**(4), 260–271 (2011)
6. Arthur, D., Vassilvitskii, S.: K-means++: the advantages of careful seeding. In: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, Philadelphia, PA, USA, pp. 1027–1035 (2007)
7. Al-Daoud, M.B.: A new algorithm for cluster initialization. Int. J. Comput. Control Quantum Inf. Eng. **1**(4), 1016–1018 (2007)
8. Redmond, S.J., Heneghan, C.: A method for initialising the K-means clustering algorithm using kd-trees. Pattern Recogn. Lett. **28**(8), 965–973 (2007)
9. Hasan, M.A., Chaoji, V., Salem, S., Zaki, M.J.: Robust partitional clustering by outlier and density insensitive seeding. Pattern Recogn. Lett. **30**(11), 994–1002 (2009)
10. Nazeer, K.A.A., Sebastian, M.P.: Improving the accuracy and efficiency of the K-means clustering algorithm. In: World Congress on Engineering, WCE 2009, Hong Kong, China, vol. 1, pp. 308–312 (2009)

11. Yedla, M., Pathakota, S.R., Srinivasa, T.M.: Enhancing K-means clustering algorithm with improved initial centre. Int. J. Comput. Sci. Inf. Technol. **1**(2), 121–125 (2010)
12. Goyal, M., Kumar, S.: Improving the initial centroids of K-means clustering algorithm to generalize its applicability. J. Inst. Eng. (India): Ser. B **95**(4), 345–350 (2014)
13. Broder, A., Garcia-Pueyo, L., Josifovski, V., Vassilvitskii, S., Venkatesan, S.: Scalable K-means by ranked retrieval. In: Proceedings of the 7th ACM International Conference on Web Search and Data Mining, New York, NY, USA, pp. 233–242 (2014)
14. Cap, M., Prez, A., Lozano, J.A.: An efficient approximation to the K-means clustering for massive data. Knowl.-Based Syst. **117**, 56–69 (2017)