[](http://crossmark.crossref.org/dialog/?doi=10.1145%2F3457682.3457695&domain=pdf&date_stamp=2021-06-21)**An Improved K-means Algorithm Based on Multiple Clustering and Density**

Yulong Ling

Key Laboratory of Big Data Storage and Management

(Northwestern Polytechnical University), Ministry of

Industry and Information Technology, Xi’an, Shaanxi

710129, China

**ABSTRACT**

The initial clustering center set of the k-means algorithm is ran- domly selected, which leads to unstable clustering results. To ad- dress this shortcoming, many improved k-means algorithms based on density have propersed, but the time complexity of these al- gorithms is too high. In order to improve clustering stability and reduce the clustering time, this paper proposes an improved al- gorithm based on multiple clustering and density. This algorithm firstly calls the k-means algorithm for many time, and adaptively selects excellent sample set according to the distance between sam- ples and the corresponding cluster center. Then the initial cluster center set is selected according to the principle of the furthest dis- tance and high density. The experiment on the UCI data sets shows that the algorithm in this paper not only improves the performance but also ensures the stability of clustering result compared with the k-means algorithm and the kmeans++ algorithm. Compare to im- proved density-based k-means algorithms, the proposed algorithm can greatly save the clustering time.

**CCS CONCEPTS**

• **Computing methodologies** → Machine learning; Learning paradigms; Unsupervised learning; Cluster analysis.

**KEYWORDS**

K-means Algorithm, Initial Cluster Center Set, Multiple Clustering, Density

**ACM Reference Format:**

Yulong Ling and Xiao Zhang. 2021. An Improved K-means Algorithm Based on Multiple Clustering and Density. In 2021 13th International Conference on Machine Learning and Computing (ICMLC ’21), February 26–March 01, 2021, Shenzhen, China. ACM, New York, NY, USA, [7](#bookmark1)pages. [https://doi.org/](https://doi.org/10.1145/3457682.3457695) [10.1145/3457682.3457695](https://doi.org/10.1145/3457682.3457695)

**1 INTRODUCTION**

The k-means algorithm is a clustering method based on a partition, which is often used to classify data [[1]](#bookmark2). The k-means method is widely used in various fields because of its many advantages such

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ICMLC ’21, February 26–March 01, 2021, Shenzhen, China © 2021 Association for Computing Machinery.

ACM ISBN 978-1-4503-8931-0/21/02. . . $15.00 <https://doi.org/10.1145/3457682.3457695>

Xiao Zhang

Key Laboratory of Big Data Storage and Management

(Northwestern Polytechnical University), Ministry of

Industry and Information Technology, Xi’an, Shaanxi

710129, China

as easy to understand, efficient grouping of large data sets, low time complexity, and so on [[2–6]](#bookmark3). However, the k-means algorithm randomly selects k samples in the sample set as the initial clustering center, which will cause many problems :(1) starting from different initial clustering centers, the clustering results will be different, which results in the clustering results relying on the selected initial clustering center set to a large extent. (2) The randomly initialized clustering center set cannot well reflect the distribution of data, which leads to the disadvantage of the clustering results; (3) In a sample set with a lot of noise points, the performance of k-means algorithm will be worse [[7–10]](#bookmark4). The above deficiencies of k-means promote the emergence of an improved k-means algorithm that selects the initial clustering center according to the data density. The idea of such algorithms is to use the density of data object to represent the distribution of data. By calculating the density of each data, this kind of algorithm selects K samples with high density as the excellent initial clustering central set. In this way, the stability of k-means algorithm can be guaranteed and the performance of k-means algorithm can be hopefully improved.

Literature [[11] selects the initial clustering center by combining](#bookmark5) the data density distribution and KD-tree. Literature [[12] calculated](#bookmark6) the data density and selected k sample points with the highest density as the initial clustering center. Literature [[13] constructed](#bookmark7) k data sets closely related to the actual distribution of data with the greedy idea, and then took the average value of the data in the data set as the initial clustering center. Literature [[14] uses the](#bookmark8) spatial distribution information of data set to define the density of data objects, and selects sample points located in the high-density area and far apart as the initial clustering center. Literature [[15]](#bookmark9) obtained the density of each data by adjusting the density threshold and then selected k sample points in the high-density region with the furthest distance from each other as the initial clustering center. Literature [[16] proposed to select the k value of clustering number](#bookmark10) based on SSE, and select the initial clustering center based on the principle that the surrounding area where the clustering center is relatively dense and the distance between the clustering center points is relatively far.

Although this type of density-based method of fixing the initial clustering center set is stable, but it has two serious drawbacks: (1) When selecting the initial clustering center, it is necessary to calcu- late the density of all data or the distance between all data, which makes the time complexity of the algorithm too large, especially when dealing with large data sets, (2) If the initial cluster center set is poorly selected, it will result in poor clustering results no matter how many times the algorithm is run.

Based on the summary of previous studies, combined with the advantages of the two types of methods: (1) randomly select the initial cluster center set; (2) select the initial cluster center set based on density. This paper proposes an improved k-means algorithm based on multiple clustering and density, namely MCAD-Kmeans algorithm. Compared with the k-means algorithm and kmeans++ algorithm, this algorithm has higher performaces and better sta- bility. Compared with the density-based algorithm, it has higher performance and greatly reduces the time required for clustering. The following chapters of this paper are arranged as follows:

The second section introduces the idea of the k-means algorithm and improved k-means algorithm based on density. The third sec- tion introduces the principle and concrete implementation of the improved kmeans algorithm in this paper, and the fourth section is the experimental results and analysis. The final section is the summary of this paper.

**2 K-MEANS ALGORITHM AND**

**DENSITY-BASED K-MEANS ALGORITHM**

The idea of the k-means algorithm is as follows: First, k samples were randomly selected as the initial cluster center set, where each sample represented the mean or center of a cluster. Then Calculate the distance of the remaining data to all the cluster centers and assign the label of each sample to the cluster closest to it. Finally, the mean value of each cluster is calculated as the cluster center for the next iteration. This process is repeated until all the cluster centers no longer change.

Among the dense-based improved k-means algorithm, a repre- sentative is a method proposed in the literature [[16]](#bookmark10). The idea of the algorithm is to select a high-density point based on a relatively far distance as the initial clustering center. The basic steps are as fol- lows: (1) calculate the mean distance between all samples, denoted as avgDist; (2) For each samples i, count the number of samples contained in a ball centered on si and with avgDist as radius, and define this number as the density of si; (3) Calculate the mean den- sity of all samples, denoted as minPoints; (4) Find all samples whose density is greater than the mean density minPoints, and form them into a new set, denoted as high density set D; (5) Find the sample point with the highest sample density in the high-density set D as the first initial clustering center point, denoted as c1 ; (6) Point c2 , the farthest point from c1 , is identified from the high density set D and used as the second initial clustering center point. According to the above method, keep searching until the kth initial clustering center point is found, which is denoted as ck. (7) Output k initial center points {c1, c2, ...ck} and the algorithm is finished.

In this paper, this algorithm is implemented and named DB- Kmeans, which is utilized to compare with the k-means algorithm and the improved algorithm in this paper.

**3 THE MCAD-KMEANS ALGORITHM**

**3.1 APPROACH**

The MCID-Kmeans algorithm in this paper combines the advan- tages of the two methods of randomly selecting initial clustering center set and selecting initial clustering center set based on density. Meanwhile, improvements are made to the shortcomings of these two algorithms. The idea is as following:

(1) The randomly initialized clustering center set used in the k-means algorithm cannot well reflect the distribution of data, while the k samples with the furthest distance from each other in the sample set are more representative.

(2) When selecting an initial clustering center point, it is likely that the noise point will be taken if only the farthest sample is considered, which can have a significant impact on the clustering results. Therefore, when selecting the initial clus- tering center set, the density of the data point needs to be considered in order to avoid getting the noise point.

(3) Samples around the final clustering center set obtained by the k-means algorithm are high-density sample points without outliers.

(4) When obtaining the initial clustering center, the density- based improved Kmeans algorithm needs to calculate the distance between pairs of all samples, which results in the time spent far exceeding that of the K-means algorithm.

The steps of the MCAD-Kmeans algorithm are as follows:

(1) Call the k-means algorithm to get the final clustering center set C = {c1, c2, ...ck}. For ci, samples of a certain proportion (denoted as P) from the sample set that belong to ciare se- lected as candidate excellent samples. Each time the k-means algorithm is called, a batch of candidate excellent samples is obtained. The excellent sample set is obtained after H times k-means is called.

(2) Count the number of occurrences of each sample in the ex- cellent sample set. In this paper, the occurrence frequency of each sample is defined as the density of the sample.Selecting the sample with the highest density from the excellent sam- ple set and add it to the initial cluster center set. Firstly, M samples with the longest distance from the initial cluster center set are obtained. Then the samples with the highest density are selected from the M samples and added to the initial cluster center set. The initial clustering center selected in this way is not only far away from the existing initial clus- tering center set, but also a high-density data point, avoiding the interference of outliers.

In this paper, the value of M is set as K (that is, the number of clusters) from the perspective of adaptive sample size. Repeat the above process until the initial cluster center set contains k samples.

(3) Input the obtained initial cluster center set into the K-means algorithm for clustering

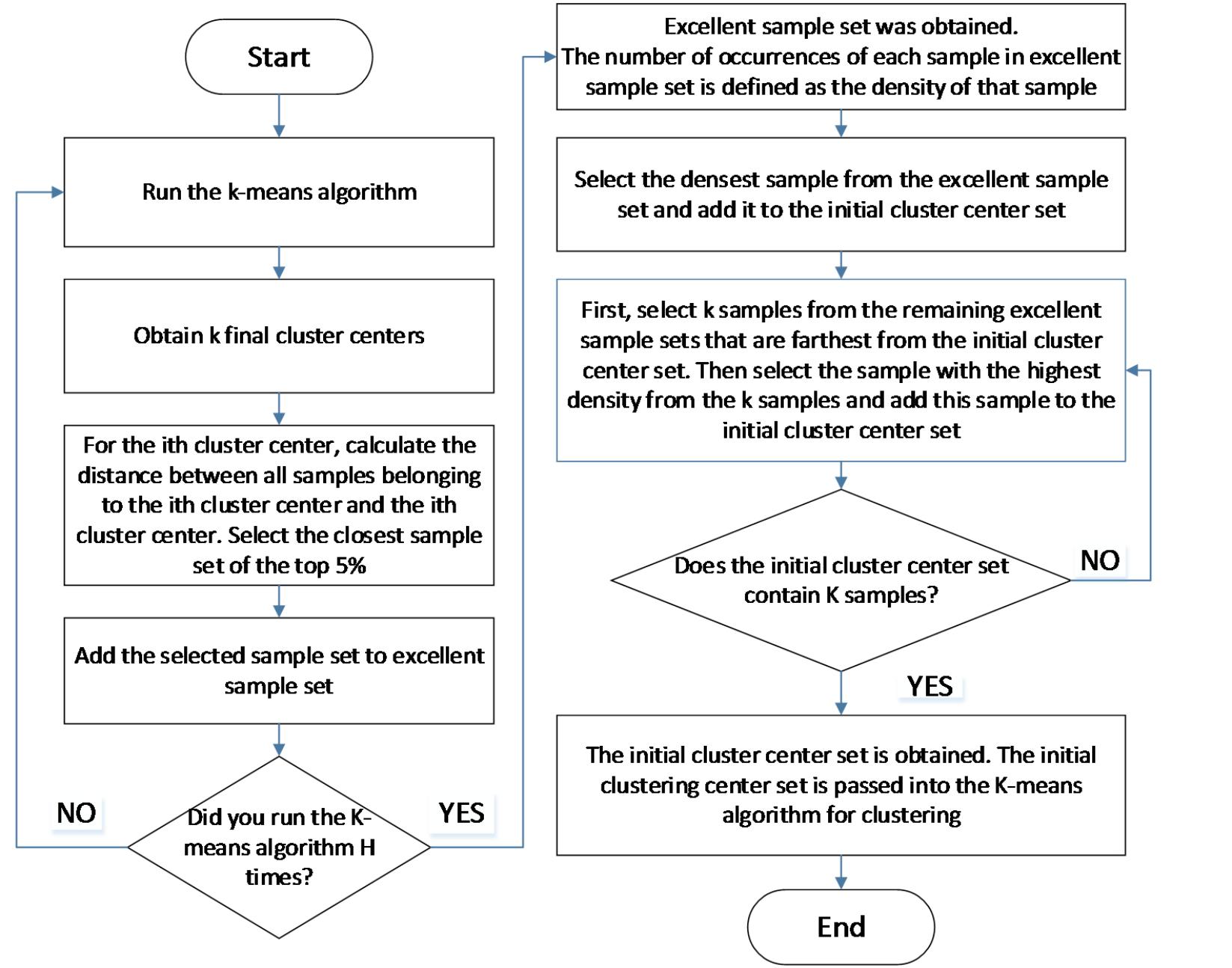
The workflow of the MCAD-Kmeans algorithm is shown in Figure [1](#bookmark11)

The MCAD-kmeans algorithm involves two parameters H and P, and the setting scheme of these two parameter values will be discussed in detail in the experimental part.

**3.2 Time complexity analysis**

The time complexity of each algorithm is given in Table [1,](#bookmark12) and the detailed analysis process is shown below. Let n be the number of samples, kbe the number of class, and T be the number of iterations of k-means algorithm. Each iteration of the k-means algorithm calculates the distance from n samples to the center of k clusters,

An Improved K-means Algorithm Based on Multiple Clustering and Density ICMLC ’21, February 26–March 01, 2021, Shenzhen, China



**Figure 1: The workflow of the MCAD-Kmeans algorithm**

**Table 1: Time complexity of four clustering algorithms**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | k-means | kmeans++ | DB-KMeans | MCAD-Kmeans |
| Time complexity | O(n) | O(n) | O(n2) | O(n) |

with a time complexity of O(nk). The k-means algorithm requires T iterations, so the total time complexity is O(nkT) = O(n).

The time complexity of the kmean++ algorithm is the same as that of the k-means algorithm, which is not analyzed here.

In the DB-Kmeans algorithm, it is necessary to calculate the distance between all sample pairs when calculating the sample density, so the time complexity is O(n2). Input the obtained initial clustering center into the k-means algorithm and run it to obtain the final clustering result. The corresponding time complexity is O(nkT), so the total time complexity of the DB-Kmeans algorithm isO(nkT) + O(n2) = O(n2).

In the algorithm in this paper, we first need to run the k-means algorithm for H times to get an excellent sample set, with time complexity O(HnkT). Then, the excellent sample set is used to obtain the initial cluster center with time complexity O(npk2). Finally, input the obtained initial clustering center into the K- means algorithm to obtain the final clustering result, with time complexityO(nkT). So the total time complexity of our algorithm is O(HnkT) + O(npk2) + O(nkT) = O(n).

From the above analysis, it can be seen that the algorithm in this paper has the same order of magnitude of time complexity as the k-means algorithm and kmeans++ algorithm. The algorithm in this paper has more time advantage than the DB-Kmeans algorithm, especially in the case of large-scale data sets..

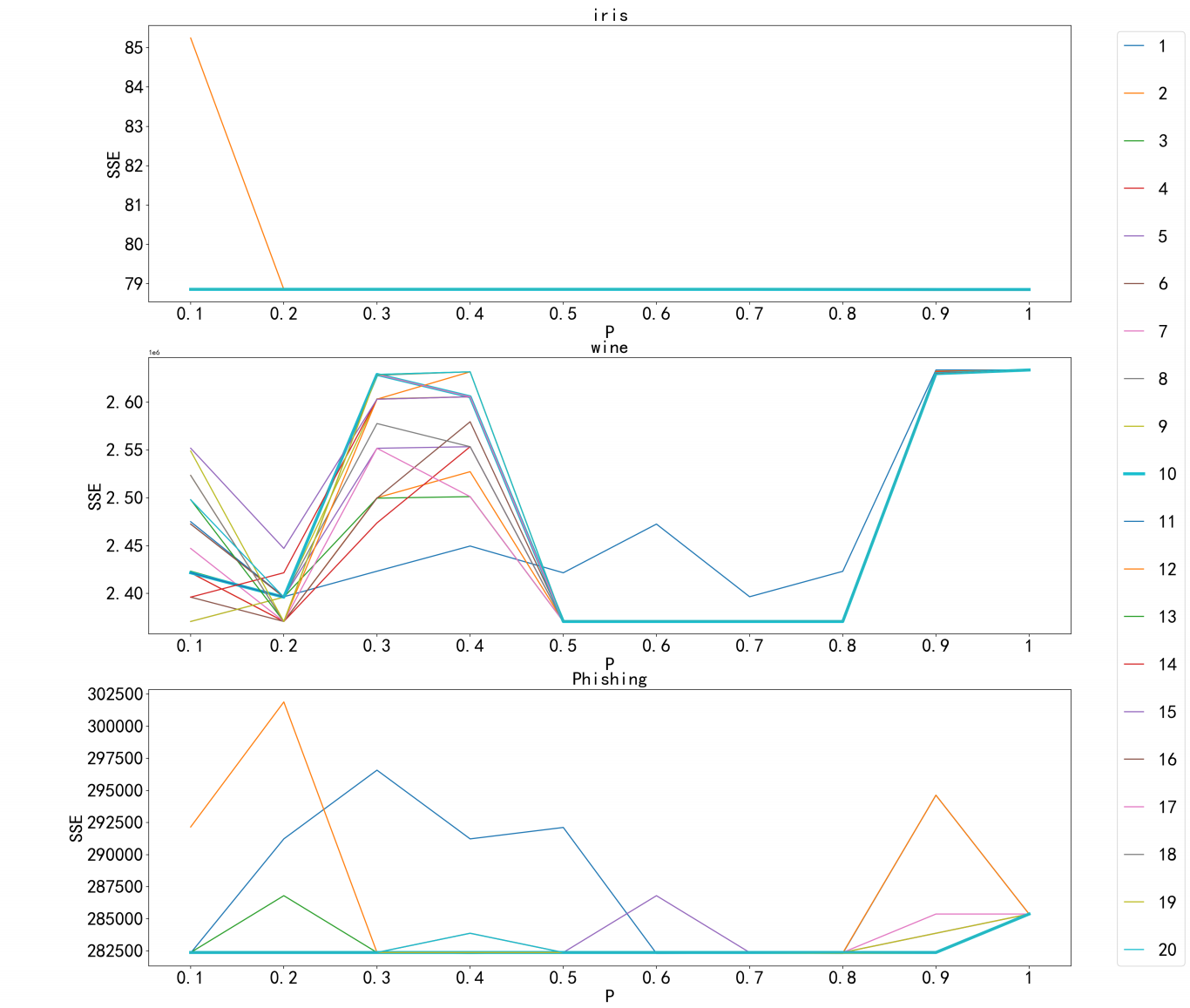
**4 EXPERIMENT AND ANALYSIS**

The experimental environment of this paper: Intel Corei7- 7700@3.66GHz, 16G memory, 200G hard disk, Window7 operating system.the experimental tool is the Spyder editor that come with Anaconda. In order to verify the performance of the algorithm, this paper selects 3 data sets from the UCI common databases. The description of data set is given in Table [2](#bookmark13)

In this paper, each algorithm is run 50 times on all data set to decrease experimental errors. In the next section, we will analyze the performance of the MCAD-Kmeans algorithm from three an- gles: clustering performance, clustering stability, and clustering time.

**Table 2: Properties of Dataset.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Set | Number of Attributes | Number of Classes | Number of Instances | Date Donated |
| Iris | 4 | 3 | 150 | 1988-07-01 |
| Wine | 13 | 3 | 178 | 1997-07-01 |
| Phishing Websites Features | 30 | 2 | 11055 | 2015-03-26 |



**Figure 2: SSE under different H and P values**

**4.1 Clustering performance**

In order to confirm the value of the parameters H and P in this algorithm, the line graph of the sum of the error squares (SSE) of the MCAD-Kmeans algorithm on all data sets is drawn in the case of different values in the (H,P) parameter pair, as shown in Figure [2.](#bookmark14) We set H ∈ {1, 2, 3...20}, P ∈ {0.1, 0.2, 0.3... 1}, from the actual situation.

From Figure [2, we can clearly find that when P is 0.2, SSE values](#bookmark14) on all data sets of theMCAD-Kmeans algorithm are relatively small. so it is very reasonable to set P as 0.2 in this paper.

For the choice of parameter H, this paper analyzes (1) Since the clustering result of k-means algorithm is unstable, the number of times of the kmeans algorithm running H should not be too small or too large. If the value of H is too small, the number of samples in the excellent sample set will be insufficient. If the value of H is too large, noise data points will be introduced, (2) In practical application scenarios, the time spent on clustering should not be

too long, (3) The value of SSE is relatively better when P is 0.2 and H is 10. We set the value of H as10 in this paper.

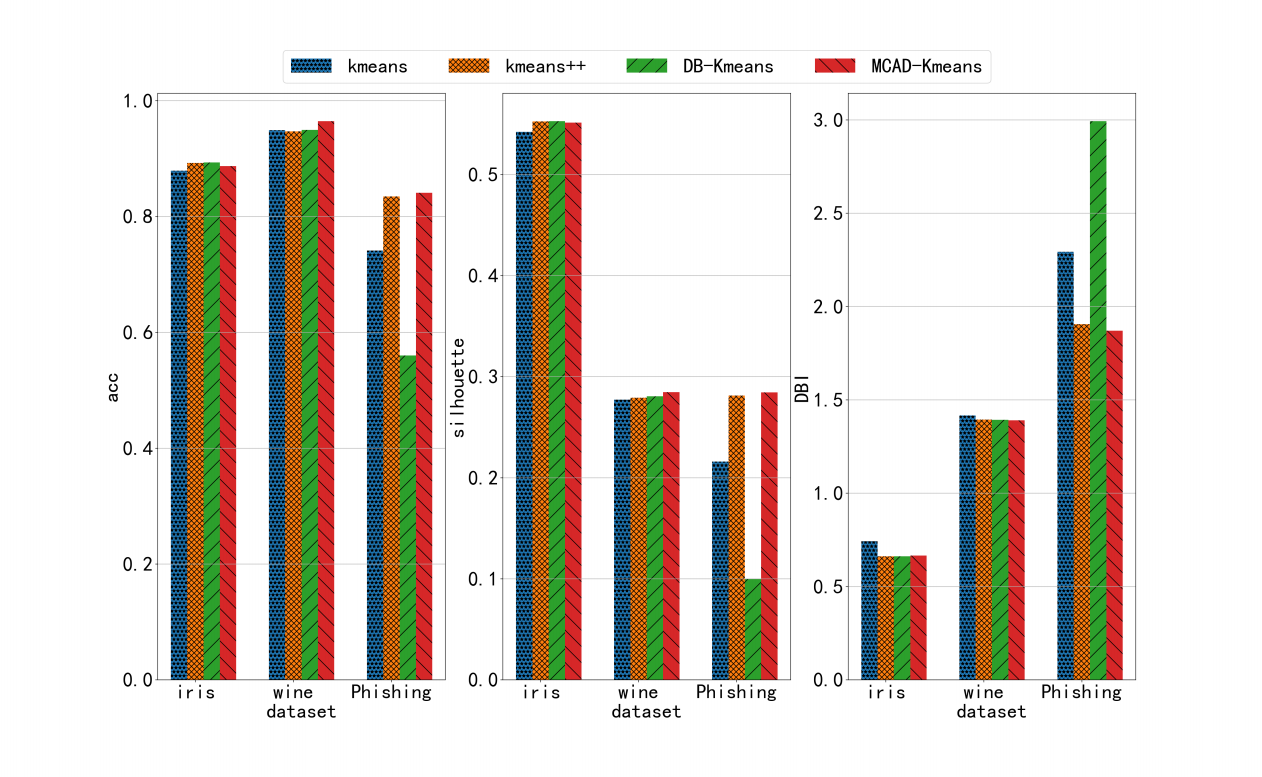
After setting the H and P parameters in the MCAD-Kmeans algorithm, we select three indicators commonly used to mea- sure clustering effect: ACC(Accuracy), Silhouette coefficient, and DBI(Avies Bouldin Index) for evaluating algorithm performance. The mean value of three indicators obtained by the four k-means algorithms running 50 experiments on three data sets is shown in Table [3, Figure](#bookmark15)[3](#bookmark16)is the bar chart corresponding to the data in Table [3](#bookmark15)

From Figure [3, we can see that (1) compared with the k-means](#bookmark16) algorithm, the accuracy rate, silhouette coefficient, and DBI of the MCAD-Kmeans algorithm in this paper are all better on all data sets. The improvement is particularly noticeable in phishing data set; (2) Compared with the kmeans++ algorithm, the performance of MCAD-Kmeans algorithm on iris data set is slightly lower. But the MCAD-Kmeans algorithm performs better on wine and Phishing

An Improved K-means Algorithm Based on Multiple Clustering and Density ICMLC ’21, February 26–March 01, 2021, Shenzhen, China

**Table 3: The performance of four K-means algorithms**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Data Set | k-means | kmeans++ | DB-Kmeans | MCAD-Kmeans |
| Accuracy | Iris | 0.8791 | 0.8927 | 0.8933 | 0.8872 |
| Wine | 0.9489 | 0.9472 | 0.9494 | 0.9646 |
| Phishing | 0.7411 | 0.8345 | 0.5599 | 0.8412 |
| Silhouette | Iris | 0.5422 | 0.5526 | 0.5528 | 0.5513 |
| Coefficient | Wine | 0.2773 | 0.2790 | 0.2806 | 0.2848 |
| Phishing | 0.2159 | 0.2813 | 0.099 | 0.2845 |
| DBI | Iris | 0.74249 | 0.6623 | 0.6619 | 0.6657 |
| Wine | 1.4169 | 1.3943 | 1.3931 | 1.3905 |
| Phishing | 2.2926 | 1.9053 | 2.9929 | 1.8708 |



**Figure 3: The performance of four kmeans algorithms.**

data sets; (3) Compared with the DB-Kmeans algorithm, the per- formance of the MCAD-Kmeans algorithm is obviously better on wine data set and Phishing data set. The performance of theMCAD- Kmeans algorithm is similar to that of DB-Kmeans algorithm on iris data set.

**4.2 Stability of clustering**

This section mainly analyzes the stability of four clustering algorithms. Figure [4](#bookmark17) shows the accuracy, silhouette coefficient, and DBI of each algorithm on all data sets, where the X-axis represents the number of experiments and the Y-axis represents each index respectively.

It can be seen from Figure [4](#bookmark17) that the k-means algorithm fluctuates more sharply in accuracy, silhouette coefficient, and DBI on all data sets. The accuracy, silhouette coefficient, and DBI of the kmeans++ algorithm on wine data set and phishing data sets fluctuate sharply. The MCAD-Kmeans algorithm in this paper basically does not fluctuate on all data sets. Since the initial clustering center set found by the DB-Kmeans algorithm are fixed,

no matter how many times the experiment is conducted, the value of each indicator will not fluctuate.

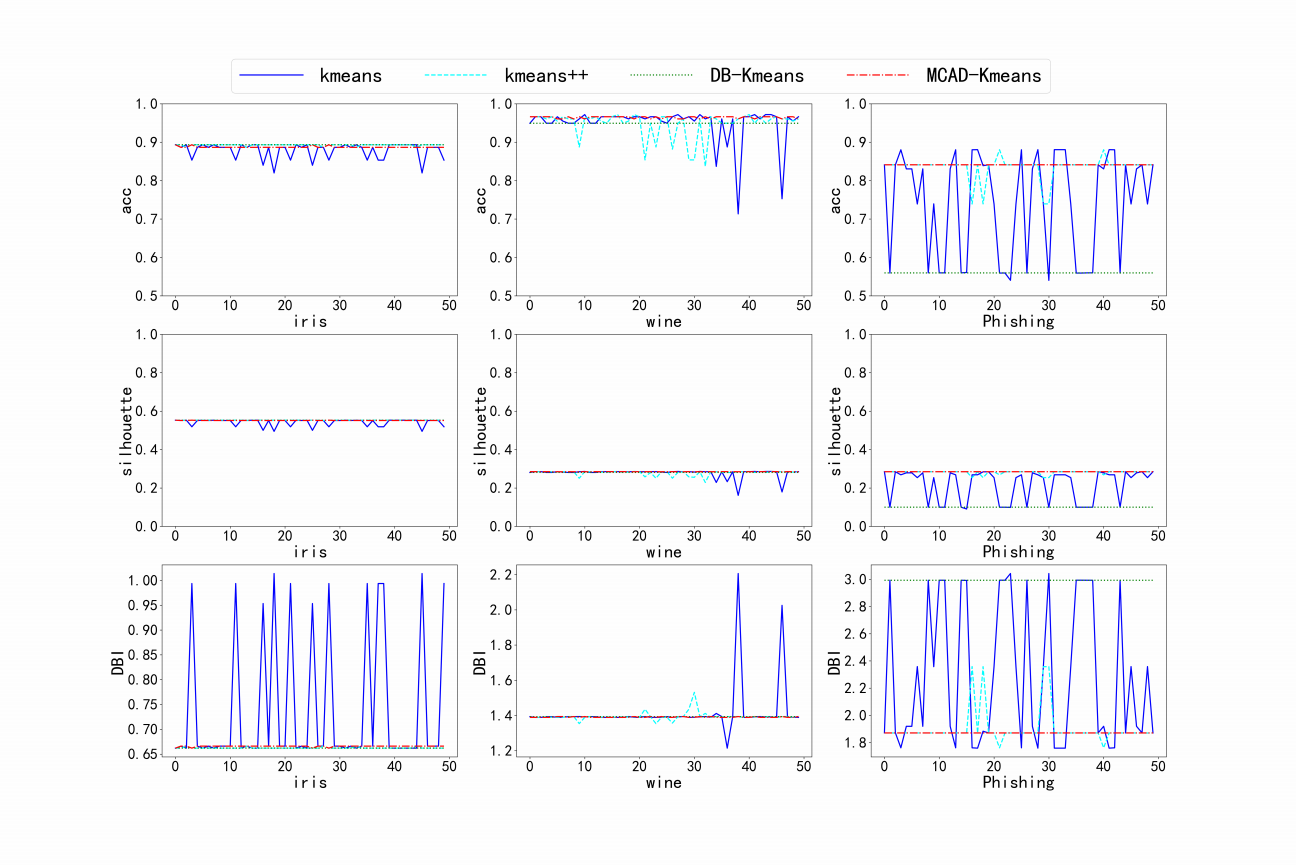
**4.3 Clustering time**

Table [4](#bookmark18)displays the clustering time (in seconds) of the four algo- rithms on all data sets. Figure [5](#bookmark19)is the line graph corresponding to Table [4](#bookmark18)

By comparing the clustering time of each algorithm in Table [4,](#bookmark18) it can be found that the time of the DB-Kmeans algorithm and the MCAD-Kmeans algorithm is greater than that of the k-means algorithm and the kmeans++ algorithm. This is mainly because these two algorithms both spend more time in calculating the initial clustering center set.

It can also be seen from Table [4](#bookmark18)that the clustering time of the MCAD-Kmeans algorithm is significantly lower than that of the DB-Kmeans algorithm on the same data set.

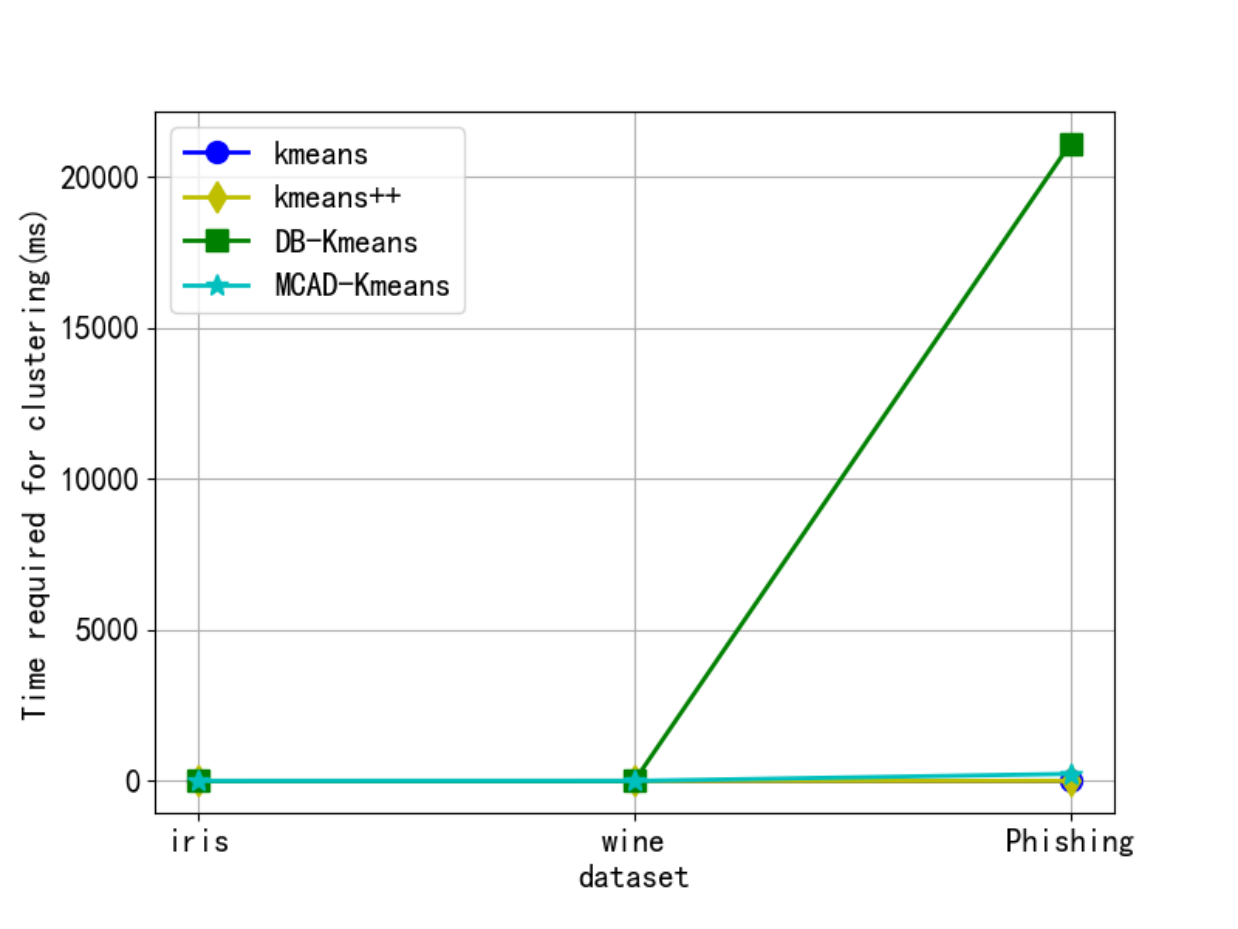
It can be seen from Figure [5](#bookmark19) that the clustering time of the k- means algorithm, the kmeans++ algorithm, and theMCAD-Kmeans algorithm increases linearly with the increase of the number of samples in the data set, while the DB-Kmeans algorithm increases



**Figure 4: The fluctuation of results obtained by four kmeans algorithms.**

**Table 4: Clustering Time of Four Kmeans Algorithms.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Set | k-means | kmeans++ | DB-Kmeans | MCAD-Kmeans |
| Iris | 0.027 | 0.0225 | 3.6010 | 2.0222 |
| Wine | 0.0812 | 0.0424 | 5.2366 | 2.3517 |
| Phishing | 1.9528 | 2.5421 | 21098.163 | 235.2473 |



**Figure 5: Clustering time of four kmeans algorithms.**

exponentially. It will take too long to use the DB-Kmeans algorithm in a larger sample size.

From the above experimental analysis, we can conclude that:

(1) Compared with the k-means algorithm and kmeans++ algo- rithm, the MCAD-Kmeans algorithm in this paper not only improves the performance, but also ensures the stability of clustering results. The MCAD-Kmeans algorithm also has the same order of time complexity as the k-means algorithm and the kmeans++ algorithm;

(2) Compared with the DB-Kmeans algorithm, the MCAD- Kmeans algorithm saves a lot of time while having similar or even higher performance.

**5 CONCLUSIONS**

Aiming at the problem that the k-means algorithm relies on the initial clustering center set, we proposes a new improved algorithm based on multiple clustering and density, and conducts comparative verification on the UCI data sets. Experimental results show that compared with the k-means algorithm and the kmeans++ algorithm, the algorithm in this paper not only improves performance but also ensures the stability of clustering results. Meanwhile, compared with the improved density-based k-means algorithm, the algorithm in this paper greatly saves clustering time.

**ACKNOWLEDGMENTS**

We thank the anonymous reviewers for their valuable feedback. This work is supported by the MIIT Key Laboratory of Big Data Storage and Management, the National Key Research and Devel- opment Program (No.2018YFB1004401), and the Natural Science Foundation of Beijing(No. L192027).

**REFERENCES**

[1] Jain A K . Data clustering: 50 years beyond K-means[J]. Pattern Recognition Letters, 2010, 31(8):651-666.

[2] Heng Li, Xiao Zhang, Xia Li, Yanqiu Wang, Yulong Ling, and Di Mu. 2020. Con- struct the Students’ Consumption Portrait Based on Large-scale Campus Data. In Proceedings of the 2020 12th International Conference on Machine Learning and Computing (ICMLC 2020).

[3] Guoqin Li,Youwei Sun,Yongping Chai. Congestion control method based on K-means clustering in VANET[J]. Computer Engineering and Design, 2020, v.41;No.398(02):42-46.

[4] S. Lu et al., "Clustering Method of Raw Meal Composition Based on PCA and k-means," 2018 37th Chinese Control Conference (CCC), Wuhan, 2018, pp. 9007- 9010, doi: 10.23919/ChiCC.2018.8482823.

[5] Agarwal J, Nagpal R, Sehgal R.Crime Analysis using K-Means Clustering[J]. International Journal of Computer Applications, 2018, 83(4):1-4.

[6] Peng K , Leung V C M , Huang Q . Clustering Approach Based on Mini Batch k- means for Intrusion Detection System over Big Data[J]. IEEE Access, 2018:11897- 11906.

[7] Zhong C , Malinen M , Miao D , et al. A fast minimum spanning tree algorithm based on K-means[J]. Information ences, 2015, 295:1-17.

[8] Sami Sieranoja, Pasi Fränti. Random Projection for k-means Clustering[C]// International Conference on Artificial Intelligence & Soft Computing. Springer, Cham, 2018.

[9] Pasi, Fr?nti, Sami, et al. K-means properties on six clustering benchmark datasets[J]. Applied Intelligence, 2018.

[10] Wei Y, Jang-Jaccard J, Sabrina F, et al. MSD-k-means: A Novel Algorithm for Efficient Detection of Global and Local Outliers[J]. 2019.

[11] Redmond SJ , Heneghan C . A method for initialising the K-means clustering algorithm using kd-trees[J]. Pattern Recognition Letters, 2007, 28(8):965-973.

[12] Lingbo Han,Qiang Wang , et al. Improved k-means initial clustering center selection algorithm.Computer Engineering and Applications, 2010, 46)17(:150- 152.

[13] Xuejiao Tong,Fanrong Meng,Zhixiao Wang. Optimization to k-means initial cluster centers[J]. Computer Engineering and Design, 2011(08):2721-2723.

[14] Juanying Xie,Wenjuan Guo,Weixin Xie, et al. K-means clustering algorithm based on optimal initial centers related to pattern distribution of samples in space[J]. Application Research of Computers, 2012, 29(003):888-892.

[15] DeshengFu,Chen Zhou. Improved K-means algorithm and its implementation based on density[J].Journal of Computer Applications, 2011(02):432-434.

[16] Sujie Zhang. Algorithm research of optimal cluster number and initial cluster center[J].Application Research of Computers,2017(06).