International Core Journal of Engineering Volume 7 Issue 4, 2021
ISSN: 2414-1895 DOI: 10.6919/ICJE.202104_7(4).0055

AT_DBSCAN: DBSCAN Algorithm with Self-Variable Threshold

Haifeng Ma, Aihua Wu
Shanghai Maritime University, Shanghai, China.
2791508544@qq.com

Abstract

Based on the analysis of the shortcomings of the density-based clustering algorithm DBSCAN and its improved algorithm, an improved algorithm of DBSCAN based on self-variation of threshold value is proposed: AT_DBSCAN algorithm. The algorithm calculates the distribution characteristics of the shortest neighbor distance of each node in the data set, and then calculates the distribution of different densities in the data set according to the distribution characteristics, and finally determines the different radius thresholds of different density distribution data areas based on this. In this way, different radius thresholds ε can be used for data in different areas to accept DBSCAN algorithm clustering. On the whole, it basically solves the unreasonable clustering results caused by the unfavorable selection of the parameter value ε at the initial stage of the DBSCAN algorithm, and the unreasonable clustering of data with different density distributions. The experimental results show that the algorithm has a good clustering effect, and the algorithm first obtains the radius threshold ε without manual setting, which is of great significance.

Keywords

Data Mining; kd-tree; Clustering; DBSCAN Algorithm.

1. Introduction

The DBSCAN algorithm is simple and easy to understand and can cluster density data sets of arbitrary shapes and can effectively identify abnormal points[1,2]. Therefore, the DBSCAN algorithm is widely used and studied in reality. The focus of the research is mostly based on the different adjustments of the two parameters (radius distance ε and ε-number of neighbors K) of the DBSCAN algorithm at the beginning, and how this affects the clustering results, or whether the DBSCAN algorithm can be used for special data sets Clustering is still reasonable[3]. In the DBSCAN algorithm, if the K value is not selected based on actual applications, one percent or two percent of the total data set is usually used. The ε value is the focus of research. The ε value actually characterizes the concept of the lower bound density of clustering, that is, it requires at least K objects in the neighborhood of the core object whose radius is the ε value. In other words, the clustering obtained by the DBSCAN algorithm is not uniform, but only meets a lower density threshold. If the original lower density threshold is extended to multiple lower density thresholds, different local data spaces correspond to different lower density thresholds. However, for the entire data space, the smallest local lower density threshold is its lower density threshold, that is, there is still only one lower density value. However, due to the use of multiple lower density thresholds, the clustering results are more refined. This is also the reason why clustering of areas with large density gaps with a radius distance will produce bad results. Either the low-density area will be identified as a noise value, or the adjacent high-density areas of the low-density area will be clustered together into a cluster.

For example, if a data set with a large difference in density distribution maintains a fixed radius threshold ε, then the radius threshold ε is too large, and one cluster may be planned into another

ISSN: 2414-1895 DOI: 10.6919/ICJE.202104_7(4).0055

cluster, while too small may cause the value to cluster out. Small-density clusters lose large-density clusters. Construct a data set from the shape sets data source. Initially, when the DBSCAN algorithm is performed, a radius distance threshold ε=2 and the number of neighbors K=5 are given. The result is that the two clusters on the left, the green cluster and the yellow cluster, are successfully clustered. Other sets of purple points are identified as discrete points. As shown in Figure 1, it is obvious that there should be clusters in the denser area on the right, but here the radius distance ε is set to 2 and it is not recognized, so the radius distance ε is set to 6, and the resulting atlas is shown in Figure 2. It may be that the two clusters on the right are identified but the three clusters on the left are identified as one cluster. Therefore, the DBSCAN algorithm is highly sensitive to the two parameter radius threshold ε and the number of nodes K contained in the cluster. How to set the initial radius distance threshold ε reasonably is a very important issue. Based on this goal, this paper proposes its own improved DBScan algorithm of adaptive threshold radius distance ε: AT_DBSCAN algorithm.

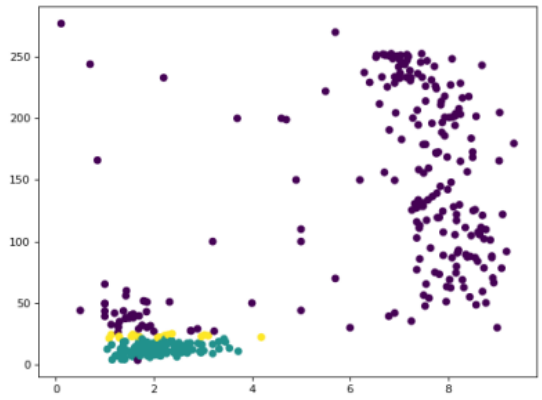


Figure 1. Radius distance ε=2

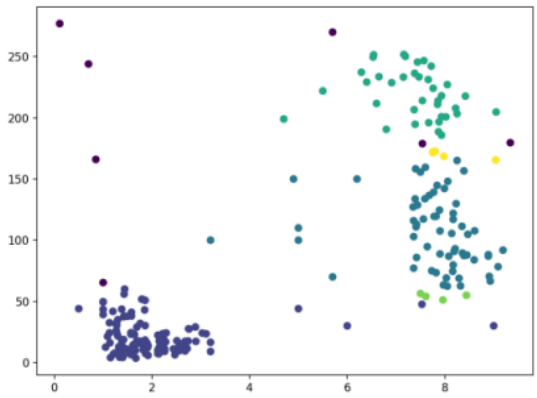


Figure 2. Radius distance ε=6

2. Related research

In recent years, there have been many ways to improve the parameter radius distance ε of the DBSCAN algorithm, some based on partitioning, and some considering the overall data distribution. No matter from any angle, each has its own advantages and disadvantages. The following is an introduction to some of the improved algorithms.

OPTICS algorithm[4]

OPTICS (Ordering points to identify the clustering structure) is a density-based clustering algorithm. This algorithm can effectively select the appropriate radius distance according to the group order characteristics of the data set. Although the algorithm initially provides the radius distance ε and K value, the algorithm process is cumbersome. Each iteration has to calculate the node core distance and reachable distance, and the actual time complexity is much greater than the DBSCAN algorithm. Although the parameters ε and K have little effect, they still have to be given artificially. Therefore, the overall process of the algorithm is not very friendly.

HDBSCAN algorithm[5]

The HDBSCAN algorithm is an improvement to the OPTICS algorithm. It extends DBSCAN by converting DBSCAN to a hierarchical clustering algorithm, and then uses the technology of extracting plane clustering based on clustering stability. Compared with DBSCAN, the biggest advantage of HDBSCAN is that there is no need to manually select the field radius distance ε and K value. Most of the time, only the size of the smallest generated cluster is selected. The algorithm can automatically recommend the best cluster result. At the same time, a new distance measurement method is defined, which can better reflect the density of points.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202104_7(4).0055

PDBSCAN algorithm[6]

The PDBSCAN algorithm effectively solves the problem of uneven density, but when there are inclusion and cross-relationships between classes, such as intertwined spiral classes and mutually included circular classes, the PDBSCAN method is difficult to take effect.

STING algorithm[7]

The STING algorithm uses a multi-resolution analysis method for clustering analysis. The quality of the clustering results when using the STING algorithm is determined by the underlying granularity of the divided hierarchy. If it is too fine, it will make the processing of clustering more difficult, and the advantages of using the STING algorithm will not be fully utilized; if the underlying granularity is too coarse, the clustering quality will be greatly reduced.

Recon-DBSCAN algorithm[8]

The Recon-DBSCAN algorithm replaces the density estimation with the density ratio estimation, and the density of the data point is the normalized result of the density of the point relative to the average density of the neighborhood. Class clusters are actually local high-density regions separated by low-density regions. The Recon-DBSCAN algorithm makes high-density clusters become lower-density clusters after normalization, and low-density clusters also become clusters after normalization. For clusters with higher density, the normalized data can be distinguished by the global radius threshold ε.

There are many other improvements or similar algorithms that are not detailed here.

3. AT_DBSCAN: DBSCAN algorithm with self-changing threshold

3.1 Algorithm idea

The AT_DBSCAN algorithm partitions the data set by density, and selects different radius distances for different partitions, and performs the DBSCAN algorithm separately to obtain the clustering results.

3.2 Related concepts

Related concepts involved in this article are shown in Table 1[9].

Table 1. Concept description

The radius distance ε	Is the radius value with the sample point x_i as the center of the circle.
ε-neighborhood	For the sample point x_i in any data set D, there is a set of all nodes within the distance of node x_i Recorded as $N\varepsilon(x_i)$.
Core node	In $N\varepsilon(x_i)$ Contain at least K samples, x_i is the core object.
Outlier	The nodes in the data set D whose distance from the core node xi is greater than the radius distance ε.
Shortest neighbor distance $minV_{ij}$	The distance between the node x_j with the shortest distance from the node x_i and the node x_i.
KD tree	A method for calculating the shortest neighbor distance.

3.3 KD tree description

The KD tree [10] is essentially a balanced binary tree, which can find the K data points that are the closest neighbors to the target point in a finite number of times. Because this algorithm needs to count the closest distances between each node and other nodes in the data set, it is necessary to set K to 2 in the KD tree, so that the nearest neighbor node must be the nearest neighbor node outside of itself. Record the second nearest neighbor node of each target node, that is, the nearest neighbor node and minVij among other nodes.

For example, there is such a set of data {(2,5), (3,4), (9,5), (4,6), (5,7), (6,6), (7,2)}, First construct the KD tree as shown in Figure 3. Then from Figure 4 we know that the pre-order nodes of the tree

ISSN: 2414-1895 DOI: 10.6919/ICJE.202104_7(4).0055

are (5,7), (2,5), (9,5), (3,4), (4,7), (7,2), (6, 6). At this time, if based on Figure 3 to find the target point (3,3).

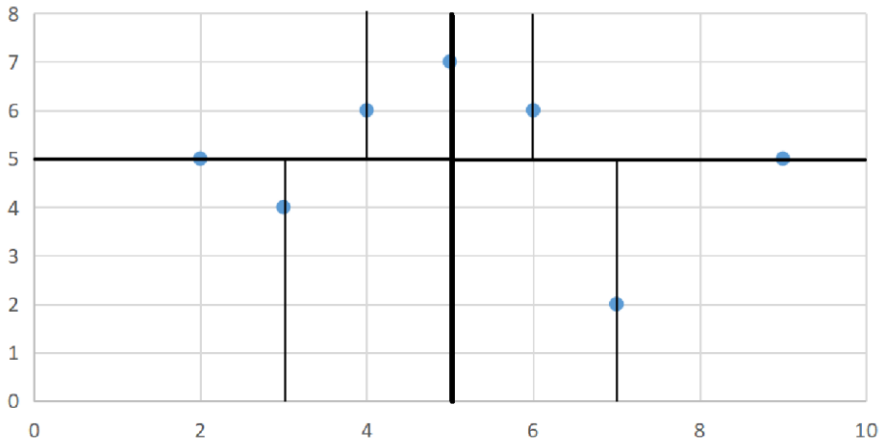


Figure 3. KD tree space diagram

Start binary search from point (7,2), and finally reach (2,3). At this time, obtaining (2,3) as the closest point of the query point, backtrack to its parent node (5,4), and judge whether there is a data point closer to the query point in the space of other child nodes of the parent node. Draw a circle with (2.1, 3.1) as the center and 0.1414 as the radius, as shown in the figure below. It is found that the circle does not intersect the hyperplane y = 4, so there is no need to enter the right subspace of the (5,4) node (the gray area in the figure) to search. Finally, go back to (7,2), take (2.1,3.1) as the center, and the circle with 0.1414 as the radius will not cross the x = 7 hyperplane, so there is no need to enter the (7,2) right subspace to search. At this point, all the nodes in the search path have been traced back, returning to the nearest neighbor (2, 3), and the nearest distance is 0.1414.

In summary, we can know that using the KD tree to find the nearest neighbor node of the target point can improve performance, because it is not necessary to find the distance between each node and the target node.

3.4 Algorithm description

The algorithm clustering process generally has the following steps:

1) Construct a KD tree, calculate the shortest neighbor distance $min\,V\,ij$ of each node in the data set, and then analyze the distribution characteristics of these shortest neighbor distance $min\,V\,ij$.

2) Obtain the density distribution characteristics of the overall data set according to the distribution characteristics of the shortest neighbor distance $min\,V\,ij$, and then obtain the respective radius distances ε of each partition.

3) Use the DBSCAN algorithm to cluster each partition.

Description of the distribution characteristics of the shortest neighbor distance $min\,V\,ij$: Using data source 1, based on the KD tree, all the nearest neighbor distance $min\,V\,ij$ distribution characteristics will be constructed to construct a histogram. We count all the distribution intervals where $min\,V\,ij$ falls on the x-axis, and count the number of each distribution interval to form the ordinate. The histogram of the data $min\,V\,ij$ is shown in Figure 4.

Analyzing Figure 4, we found that there are mainly two humps in serial numbers 4 and 6, and the subscripts 1, 2, and 3 are also concave curves, but the difference between the 2 as the bottom part and the 1, 3 peak is too small and these three The number of parts is not large enough, so they are not partitioned separately. The corresponding distances of parts 4 and 6 are 9 and 14, then we set the radius threshold to 9 and 14. In this way, we divide the data set on both sides of the threshold into two different density regions, and then apply the DBSCAN algorithm respectively.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202104_7(4).0055

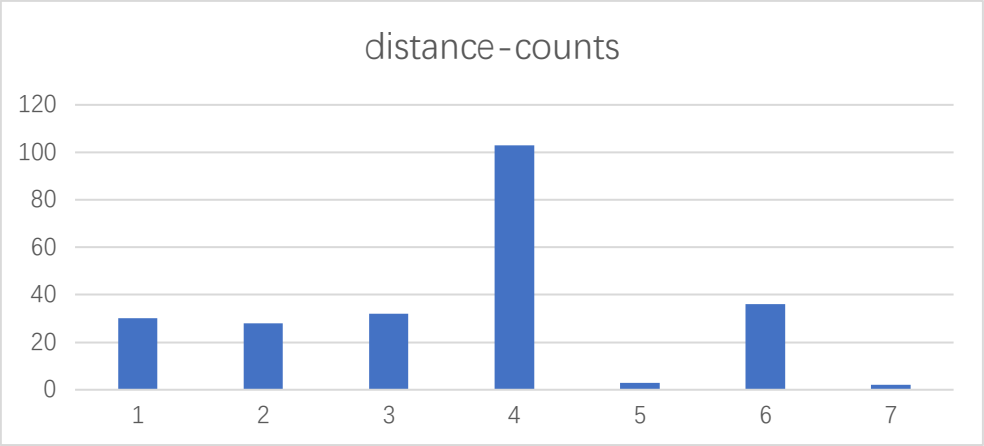


Figure 4. Distance-counts

The following gives AT_DBSCAN: the pseudo code of the DBSCAN algorithm with self-changing threshold

Input: D: A data set containing n data points

Output: Density-based clustering C

method:

1:Mark all data set points as unvisited;

2:structure KD tree;

3:Do

4:Randomly select an object x_i;

5:Mark $x_i$as visit;

6:Calculate the minV_{ij} of x_i in the KD tree

7:minV_{ij} Round up according to rounding |minV_{ij}|;

8:for Every unvisited x_j in n

9: Execution step 4-7;

10:Count each count of |minV_{ij}| as $count|\ minV_{\text{ij}}\ |$;

11:Set the distance between the density area D_i and the radius ε according to the number distribution of $count|\ minV_{\text{ij}}\ |$

12:for D_i make as unvisited

13: Mark the selected D_i as $visit$

14: for Pick any v_i from D_i and mark unvisited

15: Do

16: Make v_i as $visit$

17: if v_i of $N\varepsilon(v_i)$ contains at least K samples

18: Create new cluster C_i,v_i add it

19: else Mark as noise value

20: if All v_i in D_i is $visit$

21: end;

22: if All D_i is $visit$

22: end

23:Output all C_i

ISSN: 2414-1895 DOI: 10.6919/ICJE.202104_7(4).0055

4. Experimental performance evaluation

The experimental environment is the experimental running environment: Windows 10 operating system, 2.00GHz, 4-core processor, 8GB memory, and the language is written in Python3.

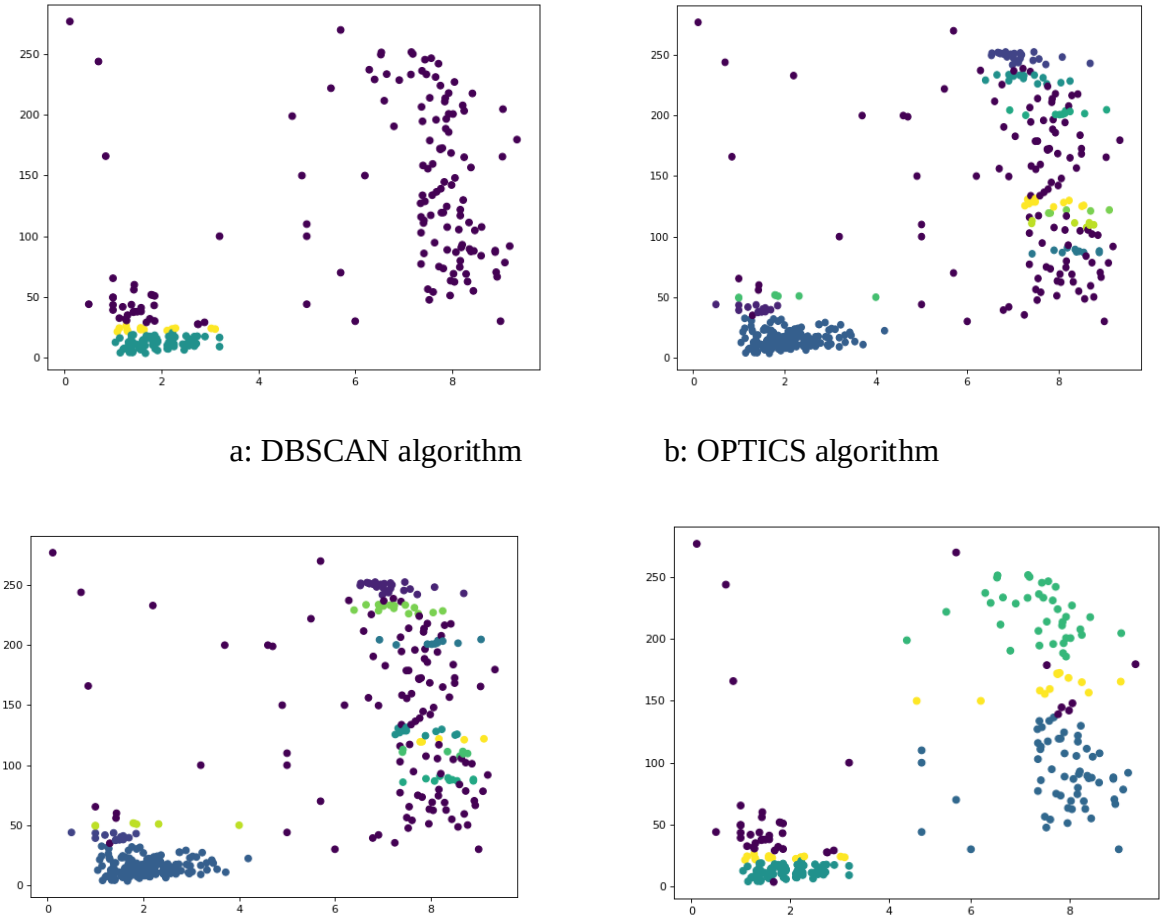
Experimental data: This paper selects two groups of shape sets data set and adds some artificial nodes to construct data set one with a total of 235 data. Data sets two and three are two sets of data sets constructed by the urben land data set in the ICU database, which contain 507 data points and 1014 data point sets respectively.

4.1 Clustering effect

The three data sets are clustered using the DBSCAN algorithm, OPTICS algorithm, HDBSCAN algorithm and AT_DBSCAN algorithm respectively, and the results obtained are shown in Figures 5, 6, and 7.

The result of data set 1 clustering using AT_DBSCAN algorithm is shown in Figure 5_d, and the right data set is clustered into three clusters. The denser data set on the left is divided into two clusters. So the data set is divided into 5 clusters. The purple ones are outliers.

Observe the running results of DBSCAN on unprocessed data as shown in Figure 5_a. Obviously, the data set on the right is identified as discrete points because the overall density is greater than that on the left, and the data points on the left are both identified as purple points. Therefore, the effectiveness of the adaptive threshold improved DBSCAN algorithm in this article is very impressive. We compare this algorithm with the traditional DBSCAN algorithm and the more famous OPTICS algorithm and HDBSCAN algorithm to see the results.

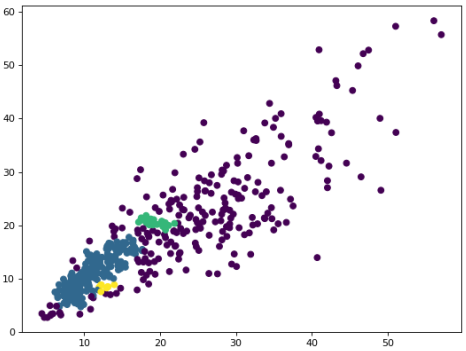


a: DBSCAN algorithm b: OPTICS algorithm

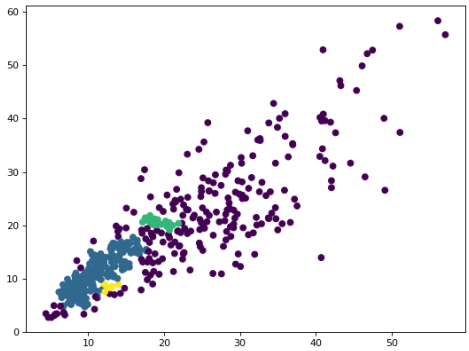
c: HDBSCAN algorithm d: AT_DBSCAN algorithm

Figure 5. Clustering results of the data set one

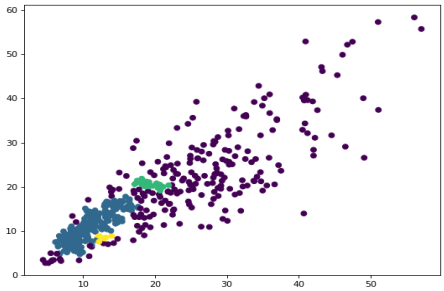
ISSN: 2414-1895 DOI: 10.6919/ICJE.202104_7(4).0055



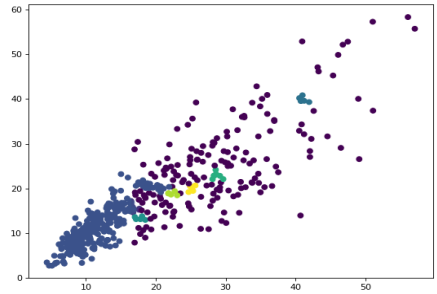
a: DBSCAN algorithm



b: OPTICS algorithm

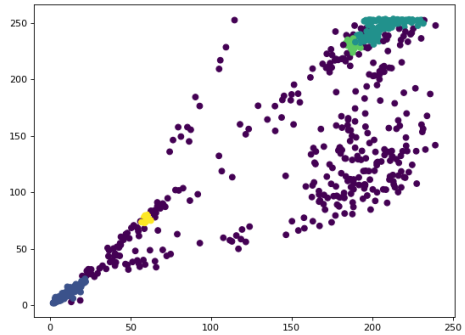


c: HDBSCAN algorithm

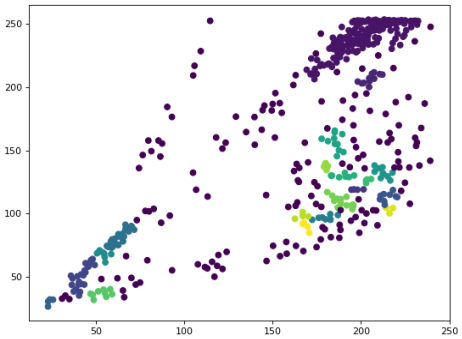


d: AT_DBSCAN algorithm

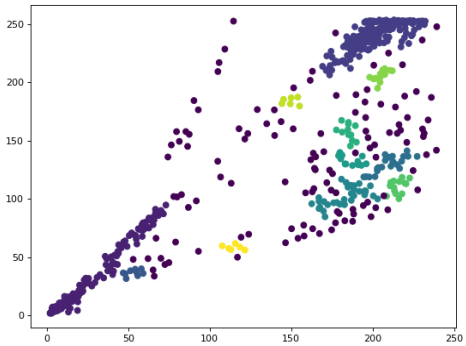
Figure 6. Clustering results of the data set two



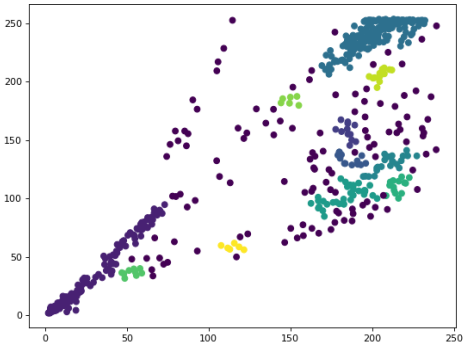
a: DBSCAN algorithm



b: OPTICS algorithm



c: HDBSCAN algorithm



d: AT_DBSCAN algorithm

Figure 7. Clustering results of the data set three

ISSN: 2414-1895 DOI: 10.6919/ICJE.202104_7(4).0055

For the general applicability of the experiment, the other two data sets are used to compare the four algorithms.

The clustering effect of data set two is shown in four pictures in Figure 6. Compared with the results of the other three algorithms, the clustering effect shown in Figure 6_d has more clusters in the low-density area, which shows that the AT_DBSCAN algorithm is more sensitive to the density distribution.

The clustering effect of data set three is shown in four graphs in Figure 9. The results of the four algorithms in Figure 7 show that except for the clustering effect of the DBSCAN algorithm, the other three algorithms have good clustering results. All clusters on the right side of the data set except for more clusters.

4.2 Experimental performance

The time performance is shown in Figure 8. We can find that as the amount of data increases, the performance of the DBSCAN algorithm and the AT_SBSCAN algorithm is better than the other two algorithms. This is because although they are all based on the improvement of the DBSCAN algorithm, since the K in the KD tree preprocessed by the AT_DBSCAN algorithm is 1, the performance will not decrease.

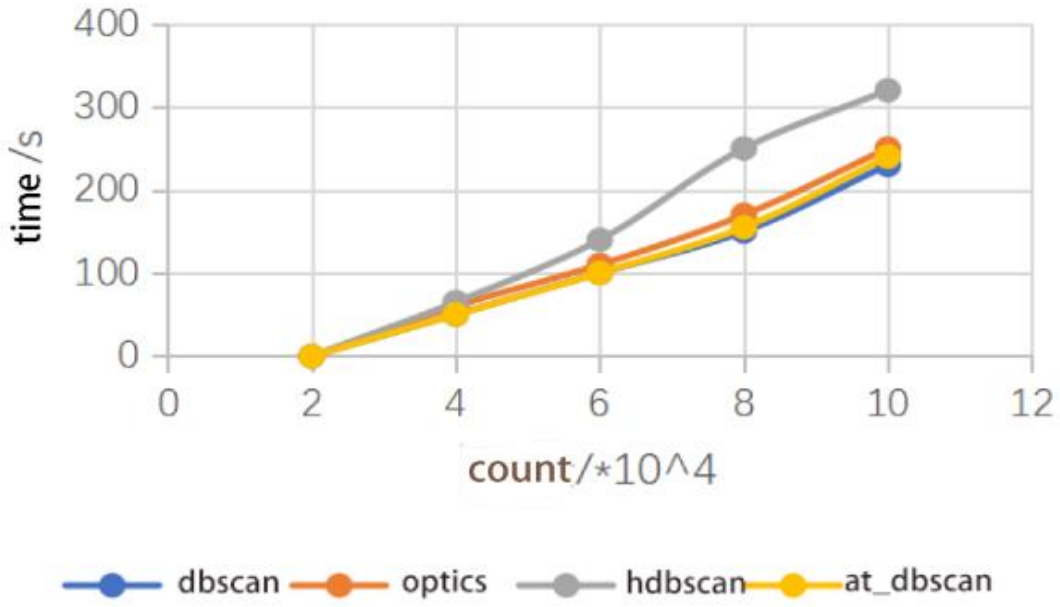


Figure 8. Time performance

In terms of algorithm accuracy, we found that comparing the four algorithms under multiple data sources, the DBSCAN algorithm cannot handle the problem of poor multi-density region clustering results. Other improved algorithms and this algorithm are all handled reasonably. The results are shown in Table 2. And the clustering results of this algorithm are almost the same as those of other improved algorithms, which also confirms the accuracy of the algorithm.

Table 2. Clustering results

	Data One	Data two	Data Three
DBSCAN	2	3	4
OPTICS	5	3	5
HDBSCAN	5	3	6
AT_DBSCAN	5	5	6

ISSN: 2414-1895 DOI: 10.6919/ICJE.202104_7(4).0055

5. Conclusion

In view of the difficulty of selecting the parameters of the DBSCAN algorithm and the difficulty of finding clusters with large differences in density, an adaptive multi-density threshold DBSCAN improved algorithm is proposed. This algorithm performs reasonable derivation and processing according to specific data sets to obtain different density distributions. Different radii should be set Threshold, and then divide the original data set into different regions according to different densities, and finally each region has its own different local ε value. Produces more refined clustering results, and solves the problem that data sets with different density distributions are not clustered due to the use of global ε values.

References

[1] Ester M, Kriegel H P, Sander J, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of ACM SIGKDD'96, Portland, 1996. 226–231.

[2] Est er M et al. A d ensit y-bas ed algorithm fo r discovering clusters in large s patial dat abas es with noise. In: Proc of 2nd Int 'l Conf on Knowl edge Discov ering in Dat abas es and Data Mining (KDD-96). Po rtl and: AAAI Press, 1996.

[3] Mcinnes L , Healy J, Astels S. hdbscan: Hierarchical density based clustering[J]. The Journal of Open Source Software, 2017, 2(11).

[4] Ankerst M, Breunig M, Kriegel H-P, et al. Optics: Ordering pointsto Identify the Clustering Structure. In: Alex D, Christos F, Shahram G, eds. Proc ACMSIGMOD Int Conf on Management of Data. Philadephia: ACMPress, 1999. 49~ 60.

[5] Mcinnes L, Healy J. Accelerated Hierarchical Density Based Clustering[J]. 2017.

[6] Yonhghong Xie,Yanhui Ma,Fang Zhou,Yinan Liu.PDBSCAN: Parallel DBSCAN for Large-Scale Clustering Applications[J]. Journal of Donghua University(English Edition), 2012,29(01):76-79.

[7] Zhan W et al. S TIN G:A st atistical in formation grid app roach to spati al dat a mining. In: Proc of th e 23rd V LDB Conf erence. Ath ens: Morgan Kaufmann, 1997. 186~ 195

[8] Shultz J M, Marcelin L H, Madanes S B, et al. The"Trauma Signature":understanding the psychological consequences of the 2010 Haiti earthquake[J]. Prehos-pital & Disaster Medicine, 2011, 26(5):353-366.

[9] Ester M, Kriegel H-P, Sander J, et al. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databas- es with Noise. In: Proc 2nd Int Conf on Knowledge Discovery and Data Mining (KDD-96). Portland: ACM Press, 1996. 226~ 231.

[10] K Nearest Neighbor Query Based on Improved Kd-Tree Construction Algorithm Chen Xiao-kang Liu Zhu-son[].