GO-DBSCAN: Improvements of DBSCAN Algorithm Based on Grid

Ling Feng, Kejian Liu, Fuxi Tang, and Qingrui Meng

***Abstract*—For it can identify the clusters with any shape and tackle the boundary points effectively, the typical density-based method of DBSCAN was widely applied to the clustering analysis. But the algorithm still has some shortcomings, such as the high time complexity, clustering effect is very dependent on the initial value of the parameter, and the low accuracy of the boundary points tackling processes. This paper put forward GO-DBSCAN, which based on the DBSCAN and OPTICS algorithms. GO-DBSCAN improved the accuracy while processing the boundary points, that cause of it introduced the minimum acceptable distance of OPTICS. In order to reduce the time complexity of clustering processes, it also proposed the method of grid-based query while it retraverse the neighborhood. At the end of this paper, we proved that GO-DBSCAN would perform better both on the accuracy of boundary points processing and time complexity.**

***Index Terms*—Clustering, DBSCAN, OPTICS, grid, boundary point.**

I. Introduction

Traditional clustering algorithm can be divided into five categories [1], [2]: (1) Partition-based clustering algorithm, which represented by K-Means, K-modes, etc. This kind of algorithm are simple and fast, but they unable to find convex spherical data set. In addition, these algorithms are very sensitive to “noisy points” and isolated points. The number of clusters must be set in advance [3]. (2) Hierarchical-based clustering algorithm, its representative algorithms are CURE, BIRCH, etc. The advantages of this kind of algorithms are able to identify the non-spherical data set and noisy points. But it’s difficult to select the consolidation points, and the scalability also lower than other algorithms [4], [5]. (3) The algorithm STING [6], CLIQUE [7], etc. are the representatives of the Grid-based algorithm. The feature of these algorithms is fast, but their quality are very dependent on the size of the grid [2]. (4) Model-based clustering algorithms can solve the noisy points and outliers, and deal with high-dimensional data effectively. The representatives are COBWEB [8], SOM [9], etc. But this kind of algorithms are unable to tackle the non-normal data set [10]. (5)Density-based clustering algorithms are able to tackle the noisy points and discover the data set with any shape [11]-[14]. But its time complexity is very high. The representatives are OPTICS [11], DBSCAN [12], etc.

DBSCAN is the most widely used algorithm among the density-based algorithms. Comparing with the traditional algorithms, DBSCAN has advantages in identifying the data set with any shape, dealing with noisy points and not needing to set the number of clusters. But DBSCAN operates the data set directly, thus would consume too much time which result to the time complexity of the algorithm be $O(n^2)$. The influence on operating efficiency would be more obviously while the data set is bigger. In addition, DBSCAN is too sensitive to the initial parameters and boundary points, which would influence the quality of clustering results directly.

In order to improve the time complexity of DBSCAN, many scholars have proposed the method based on grid, such as, LIU, etc., have presented the method that dividing the data set by grid so that they can just traverse the k-neighbor grid while searching the ε-neighborhood of objects, thus they could improve the time complexity of DBSCAN [15]. HuangDarong proposed that combining DBSCAN and CLIQUE [16]. They conversed the core objects of DBSCAN to core grid, then subject to the grid be the least object. Ho Seok Kim, Amineh Amini proposed a method, which make a combination of density-based algorithm and grid-based algorithm [17-20]. ZHOU, etc., presented we could expand the clusters by part of core objects [21], they also proposed the DBSCAN algorithm based on data sampling and partition [12], [13].

This paper improved several shortcomings. In order to tackle the problem that the boundary point might belongs to two or more clusters, for which would influence the quality and results of DBSCAN, this paper referenced the minimum reachable distance of OPTICS algorithm into the improved DBSCAN [18]. In addition, this paper introduced the method of query based on grid and applied it to improve the efficiency of DBSCAN [18]. Thus, it’s no essential to traverse whole data point in the progresses of every iteration, and it’s replaced by the progress of traverse its neighboring grid. Finally, this paper combined the two methods and proposed the improved DBSCAN algorithm, which based

Manuscript received August 4, 2016; revised October 19, 2016. Project supported by National Key Technology R&D program (No. 2011BAH26B03), National Nature Science Foundation (Grant No. 61271413, 61472329 and 61532009),t he Fund of Lab of Security Insurance of Cyberspace, Sichuan Province(szjj2015-055),Department of Education Sichuan Province key projects (16ZA0165), Key scientific research fund project of Xihua University(Z1222625), Innovation Fund of Postgraduate, Xihua University, Innovation Fund of Postgraduate, Xihua University.

Ling Feng, Kejian Liu, and Fuxi Tang are with the Xihua University, Sichuan, China (e-mail: lyn_ling_feng@163.com, liukejian@gmail.com, Fuxi_tang@163.com).

Qingrui Meng is with Tibet Feiyue Intelligence Technology CO., Ltd, China (e-mail: 414893358@qq.com).

DOI: 10.7763/IJCTE.2017.V9.1129

on grid and named GO-DBSCAN.

II. DBSCAN Algorithm and OPTICS Algorithm

Wherever Times is specified, Times Roman or Times New Roman may be used. If neither is available on your word processor, please use the font closest in appearance to Times. Avoid using bit-mapped fonts if possible. True-Type 1 or Open Type fonts are preferred. Please embed symbol fonts, as well, for math, etc.

DBSCAN is a typical algorithm density based clustering [2], its connectivity of density can detect clusters with any shape. OPTICS is an improved algorithm of DBSCAN. They both can detect any clusters with any shape through its connectivity of density, for addition, they also effective to deal with the noisy points.

A. Basic Conceptions

1) Definition 1 Core object: In the neighborhood of radius ε contains at least minpts objects

2) Definition 2 Boundary point: The neighborhood of a particular object contains less than minpts [13].

3) Definition 3 Noisy point: The neighborhood of a particular object contains 0.

4) Definition 4 Direct density reachable: if object *p* is a core object, object *q* is contained in the neighborhood of *p*, then call *q* and *p* is direct density reachable.

5) Definition 5 Density reachable: There exists a data link p_1, $p_2 \ldots p_n$, where $p_1=p_2$, $p_n=q$, if object p_i and object p_{i+1} is direct density reachable, then we call *q* and *p* is density reachable(Fig. 1).

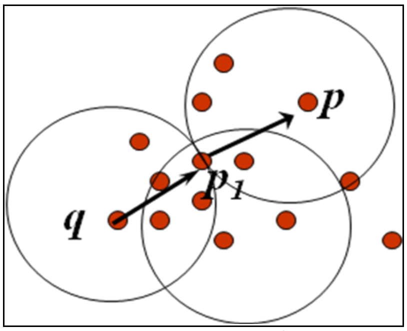

Fig. 1. Density reachable of *p* and *q*.

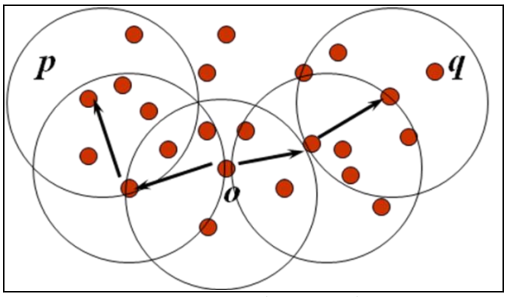

Fig. 2. Connectivity of density of *p* and *q*.

6) Definition 6 Connectivity of density: if there is an object o exists in dataset, which make the object *p* and *q* are density reachable for *o*, then object *p* and *q* are density reachable (Fig. 2).

7) Definition 7 Core distance: The minimum neighborhood radius, which make an object to become a core object called core distance, if object *p* is not core object, there is no core distance.

8) Definition 8 Reachable distance: The Euclidean distance of object *q* to core object *p* is called reachable distance, if object p is not core object, there is no reachable distance between p and *q*.

B. Basic Conceptions

We can obtain a core object p while traverse the data set, then classify unprocessed objects to the same cluster with *p*. And then, iterate the process above in the ε-neighborhood of *p*, until seeding objects are handled, thus we can obtain a complete cluster. Repeating the actions and detecting other clusters. While the data points were processed, the points were treated as noisy points if there is no cluster they belonging.

C. OPTICS Algorithm

We can obtain a core object p while traverse the data set, then computing the core distance and the reachable distance between p and the objects in its ε-neighborhood. And putting p in the sequence of results, and putting the objects of the ε-neighborhood of p in the ordering sequence and then order them by reachable distance. Next, we extract the objects of ordering sequence, of which own the minimal reachable distance. Then repeating the operations of object p until all the objects of reachable sequence were done and put into the sequence of results. Repeating all the operations above until all the data objects were finished and obtain a sequence of results, which ordering by the reachable distance.

To DBSCAN, OPTICS algorithm reduced the sensitivity about the parameters greatly though it has two parameters ε and *minpts*. The ordered sequence obtained by OPTICS can gain any result that clustered by DBSCAN with any parameter.

III. GO-DBSCAN: Improvements of DBSCAN Algorithm Based on Grid

For time complexity and the boundary point would belongs two clusters in DBSCAN, this paper proposed an improved algorithm GO-DBSCAN. The realization thought of GO-DBSCAN includes two steps: firstly, we divide the data set into different grids based on the query thought based on Grid; then, clustering the data set through the improved cluster algorithm.

A. GO-DBSCAN Divide the Data Set Based on Grid

Definition of grid cell in this paper: this paper choose the double of neighborhood radius as the length of the grid cell, then divide the data set in each dimension with the same length 2ε. Thus, the data space was divided into different grid cells, and denoted by *d*1, *d*2…*dn*, where *n* is the data dimension, d1denotes the sequence number of grid cell in different dimensions, $1 \le i \le n$ [11].

Definition the grid cell of p belonging: *x*1, *x*2…*xn* denotes the n coordinates of object *p* in each dimension, the *i-th* grid number *d*1 should fulfill: $x_i / 2\varepsilon \le d_i \le (x_i + 2\varepsilon) / 2\varepsilon$ and *d*1 is an integer.

The process just need traverse 9 grid cells at most while computing the ε-neighborhood of objects after the grid cell was set. Take two dimensional data as an example, object $p_{(x,y)}$ belongs to grid (*d*1, *d*2), the traversing scope is shown as Fig. 3:

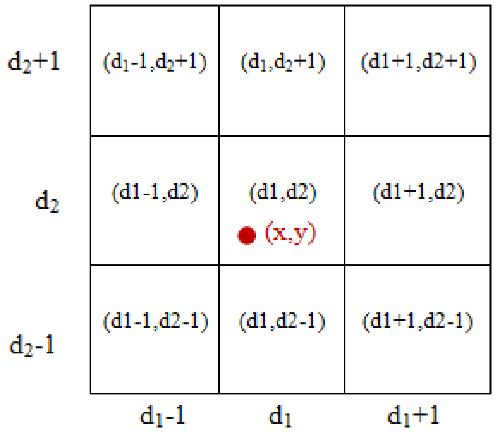

Fig. 3. Neighborhood grid in two dimensional space.

B. GO-DBSCAN Clustering Data Sets

Each object would be operate only once while DBSCAN is running, every object's label was certainly determined and it's unique and unable to change. Thus, the mistakes that the boundary points belongs two clusters are unable to change, too. Therefore, this paper put forward that combining the critical factor of OPTICS—minimal reachable distance(MRD) with DBSCAN, thus we can identify data point belongs to which cluster. In the condition that ε-neighborhood unchanged, every point will continuously update MRD rather than operated only once. The clustering label changes with the change of MRD, what's means the label of data point and the label of its closest core object are the same.

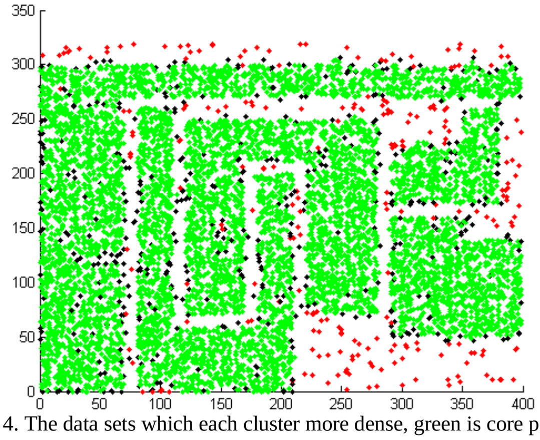

Fig. 4. The data sets which each cluster more dense, green is core points, black is boundary point, red is noisy.

Such an improvement in GO-DBSCAN can be more accurate to deal with the boundary point that belongs to clusters, and reduce the dependence on initial parameters.

Boundary point belongs to two clusters is commonly seen while the distribution of each cluster is dense and the number of boundary is quantity, illustrated by Fig. 4. There would occur some mistakes while operated by DBSCAN, but not for GO-DBSCAN.

GO-DBSCAN Pseudo-code Implementation is shown as below:

```
Input:
  D:sample data set
  ε:neighborhood radius
  minpts: neighborhood density threshold
Output:
  Data sets which completed clustering
Method:
sample data set be meshing， and mark the grid number
of every data objects
mark all objects as unvisited
for  each object p of D
        mark p as visited
        if  the number of objects in E>= minpts  (E denotes
        the ε-neighborhood of p)
                Create a new cluster C, add p into C
                for  each object q of E
                         make MRD as the minimum reachable
                         distance of q;
                        if  q is unvisited
                                mark q as visited;
                                MRD=the Euclidean distance of q to
                                p;
                                add q into C;
                                  if  the number of the objects in the
                                      ε-neighborhood of q >=minpts
                                        add all objects of the ε-
                                neighborhood of q into E;
                                end if
                         else
                             mark d as the Euclidean distance of q
                             to p
                             if  d<MRD
                                     MRD=d;
                             if  C does not contain q
                                     add q to C;
                                             delete q from original
                                             cluster;
                               if  the number of the objects in the
                                     ε-neighborhood of q >=minpts
                                       add all objects of the ε-
                                       neighborhood of q into E;
                                       delete all objects of the ε-
                                       neighborhood of q from
                                       original cluster;
                                     end if
                             end if
                     end if
             end for
     end if
```

This paper proposed the conception of boundary clustering accuracy rate based on the accuracy rate of boundary point:

$$\text{rate} = correct\_num / all\_num \tag{1}$$

where *correct_num* denotes the number of correct clustered , *all_num* means all the boundary points

To judge the result is right or not, this paper according to the reachable-distance between objects and its core object. While the clustered result of boundary points, we just need compare the RD to judge which method is right, that means the less one is right.

This paper use the Euclidean distance to compute the distance between objects, the equation shown as below:

$$d = \sqrt{\sum_{i=1}^{n}(x_{1i} - x_{2i})^2} \qquad (2)$$

where n denotes the dimensions of the data objects,

x_{1i} expresses the first point of the *i-th* dimensional coordinates, ε denotes the second point of the *i-th* dimensional coordinates.

IV. Experiment and Results

This paper references the DBSCAN implemental thought of Michal Daszykowshi [22], etc., we implemented the algorithm. This paper compared the quality of GO-DBSCAN and DBSCAN, and compared the efficiency of GO-DBSCAN and OPTICS. The time that GO-DBSCAN determines the data point which grid belongs to is not included in the operating time.

Firstly, we made a comparison of clustered quality. Table I contains the result of clustering process in the simulated database, which shown as Fig. 3. In this experiment, we set minpts equals 5. As shown in Table I, there would be some mistakes while DBSCAN operating the boundary points and we can also obtain the conclusion that GO-DBSCAN can reduce the dependence on the initial parameters while the initial parameters is large.

TABLE I: The Clustering Quality Comparison of DBSCAN Algorithm and GO-DBSCAN Algorithm

Value of ε	Algorithm	Number of cluster	Boundary clustering accuracy rate
4	DBSCAN	35	88.21%
	GO-DBSCAN	42	96.36%
5	DBSCAN	17	90.19%
	GO-DBSCAN	19	97.55%
6	DBSCAN	5	95.87%
	GO-DBSCAN	5	98.62%
7	DBSCAN	3	32.09%
	GO-DBSCAN	5	98.65%

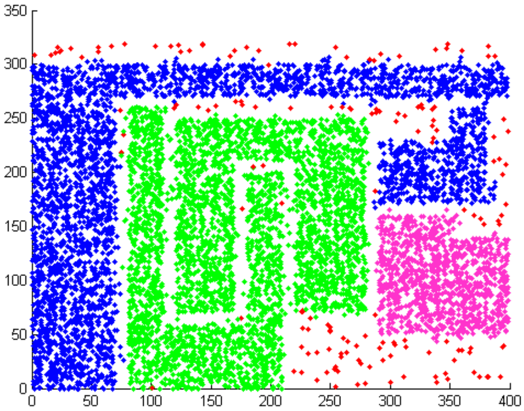

Fig. 5. The clustering results of DBSCAN in ε= 7.

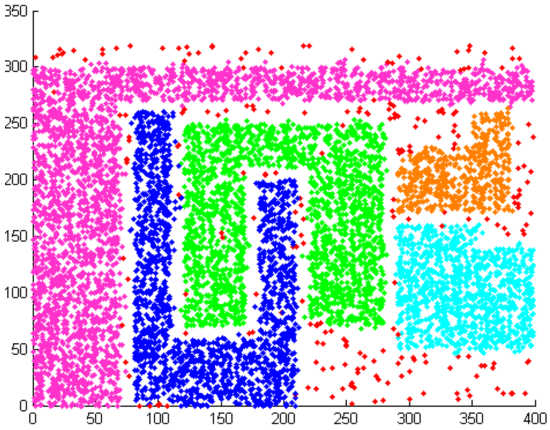

Fig. 6. The clustering results of GO-DBSCAN in ε= 7.

Fig. 5 and Fig. 6 are the clustering results of DBSCAN and GO-DBSCAN in ε=7. The experiment demonstrated that the best parameter is ε=6 for the simulated database, but while the ε is big, the DBSCAN is more easy affected by the initial parameters than GO-DBSCAN. That's means GO-DBSCAN reduced the sensitivity to parameters to a certain extent.

Fig. 7 shows the comparison of operating time among GO-DBSCAN, DBSCAN and OPTICS, but for the limits of PC memory, the experiments chose 10000 data objects at most. Fig. 7 demonstrates the time that GO-DBSCAN needed is less than DBSCAN obviously.

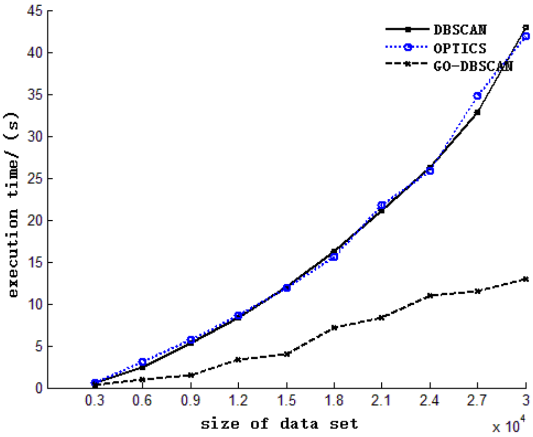

Fig. 7. Running time comparison of three algorithms.

V. Summary

DBSCAN is a common algorithm which based on density, this paper mainly discussed the problem that the boundary points belongs two clusters, which would lead to the problem of clustered quality. And we also analyzed the time complexity of DBSCAN and proposed an improved algorithm--GO-DBSCAN. GO-DBSCAN algorithm effectively solves the problem of boundary points, and improves the operating efficiency of the algorithm, and can reduce the influence of initial parameters to some extent while the initial parameter was set too big. Experiments demonstrates that GO-DBSCAN is more accurate and efficiency than DBSCAN while the clusters distribute densely and the data set has more boundary points between two clusters.

References

[1] J. G. Sun, J. Liu, and L. Y. Zhao, "Clustering algorithms research," *Journal of Software*, vol. 19, no. 1, pp. 48-61, 2008.

[2] T. Zhou, and H. L. Lu, “Clustering algorithm research advances on data mining,” *Computer Engineering and Applications*, vol. 48, no. 12, pp. 100-111, 2012.

[3] Q. Wang and C. Wang, “Review of K-means clustering algorithm,” *Electronic Design Engineering*, vol. 20, no. 7, pp. 21-24, 2012.

[4] X. L. Chen and P. H. Lou, “The application of improved hierarchical clustering algorithm to analyze literature,” *Journal on Numerical Methods and Computer Application*, vol. 30, no. 4, pp. 277-287, 2009.

[5] G. Y. Wei and X. X. Zheng, “Research on CURE algorithm of hierarchical clustering method,” *Science Technology and Industry*, vol. 5, no. 11, pp. 22-24, 2005.

[6] W. Wang, J. Yang, and R. Muntz, “STING: A statistical information grid approach to spatial data mining,” in *Athens Proc. the 23rd Conference on CLDB*, 1997, pp. 186-195.

[7] P. S. Segundo, F. Matia, and D. Rodriguez-Losada, “ An improved bit parallel exact maximum clique algorithm,” *Optimization Letters*, vol. 7, no. 3, pp. 467-479, 2013.

[8] B. Mittmann and C. Wolff, “Embryonic development and staging of the cobweb spider Parasteatoda tepidariorum,” *Development Genes and Evolution*, vol. 222, no. 4, pp. 189-216, 2012.

[9] S. Som and P. Schniter, “Compressive imaging using approximate message passing and a Markov-tree prior,” *IEEE Transactions on Signal Processing*, vol. 60, no. 7, pp. 3439-3448, 2012.

[10] H. Y. Song, “Study on model-based clustering methods,” *Journal of Chongqing University of Science and Technology: Natural Science Edition*, vol. 10, no. 3, pp. 71-73, 2008.

[11] Y. L. Zeng, H. B. Xu, and S. Bai, “OPTICS-plus for text clustering,” *Journal of Chinese Information Processing*, vol. 22, no. 1, pp. 51-56, 2008.

[12] M. Patwary, D. Palsetia, and A. Agrawal, “A new scalable parallel DBSCAN algorithm using the disjoint-set data structure,” in *Proc. International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, 2012, pp. 1-11.

[13] S. G. Zhou, A. Y. Zhou, and Y. Cao, “A data-partitioning-based DBSCAN algorithm,” *Journal of Computer Research & Development*, vol. 37, vol. 10, pp. 1153-1159, 2000.

[14] Y. F. Yu and A. W. Zhou, “An improved algorithm of DBSCAN,” *Computer Technology And Development*, vol. 21, no. 2, pp. 30-34, 2011.

[15] S. F. Liu, D. X. Meng, and X. Y. Wang. “DBSCAN algorithm based on grid cell,” *Journal of Jilin University (Engineering and Technology Edition)*, vol. 44, no. 4, pp. 1135-1139, 2014.

[16] D. R. Huang and P. Wang, “Grid-based DBSCAN algorithm with referential parameters,” *Pthysics Procedia*, vol. 24, pp. 1166-1170, 2012.

[17] M. Selim, and E. Aksehirli, “Improving DBSCAN’s execution time by using a pruning technique on bit vector,” *Pattern Recognition Letters*, vol. 32, pp. 1572-1580, 2011.

[18] P. Viswanath and V. S. Babu, “Rough-DBSCAN: A fast hybrid density based clustering method large data sets,” *Pattern Recognition Letters*, vol. 30, pp. 1477-1488, 2009.

[19] A. Amini and T. Y. Wah, “A study of density-grid based clustering algorithms on data stream,” in *Proc. 2011 Eighth International Conferrence on Fuzzy Systems and Knowledge Discovery*, 2011, pp. 1652-1656.

[20] K. M. Kumar and A. R. M. Reddy, “A fast DBSCAN clustering algorithm by accelerating neighbor searching using Groups method,” *Pattern Recognition*, 2016.

[21] S.-G. Zhou and A.-Y. Zhou, “FDBSCAN: A fast DBSCAN algorithm,” *Journal of Software*,vol. 11, no. 6, pp. 735-744, 2000.

[22] N. Thanh, T. K. Drab, and M. Daszykowski, “Revised DBSCAN algorithm to cluster data with dense adjacent cluters,” *Chemometrics and Intelligent Laboratory Systems*, vol. 120, pp. 92-96, 2013.

Ling Feng was born in Sichuan, China in 1992. She is a postgraduate student in the School of Computer and Software Engineering, Xihua University. Her research interests include computer internet, data mining.

Kejian Liu was born in Hubei, China in 1974. He is an associate professor in the School of Computer and Software Engineering, Xihua University. His research interests include cloud computing and big data analysis, computer internet, distributed computing, data mining.

Fuxi Tang was born in Chongqing, China in 1991. He is a postgraduate student in the School of Computer and Software Engineering, Xihua University. His research interests include WEB data mining, intelligence information process.

Qingrui Meng was born in Sichuan, China 1973. She is an engineer of Tibet FeiYue Intelligence Technology CO., LTD, China. Her research interests include enterprise informationization.