International Journal of Database Theory and Application
Vol.8, No.2 (2015), pp.119-128
http://dx.doi.org/10.14257/ijdta.2015.8.2.12

MRG-DBSCAN: An Improved DBSCAN Clustering Method Based on Map Reduce and Grid

Li Ma[1, 2, 3], Lei Gu[1, 2], Bo Li [1, 4], Shouyi Qiao[1, 2], Jin Wang[1, 2, 3]

[1] Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science & Technology, Nanjing 210044
[2] School of Computer & Software, Nanjing University of Information Science & Technology, Nanjing 210044
[3] Key Laboratory of Meteorological Disaster of Ministry of Education Nanjing University of Information Science & Technology, Nanjing 210044
[4] CMA Research Centre for Strategic Development, Beijing 100081

Abstract

DBSCAN is a density-based clustering algorithm. This algorithm clusters data of high density. The traditional DBSCAN clustering algorithm in finding the core object, will use this object as the center core, extends outwards continuously. At this point, the core objects growing, unprocessed objects are retained in memory, which will occupy a lot of memory and I/O overhead, algorithm efficiency is not high. In order to ensure the high efficiency of DBSCAN clustering algorithm, and reduce its memory footprint. In this paper, the original DBSCAN algorithm was improved, and the G-DBSCAN algorithm is proposed. G-DBSCAN algorithm reduces the number of query object as a starting point. Put the data into the grid, with the center point of the data in the grid to replace all the grid points as the algorithm input. The query object will be drastically reduced, thus improving the efficiency of the algorithm, reduces the memory footprint. In order to make the G-DBSCAN algorithm can adapt to large data processing, we will parallelize the G-DBSCAN algorithm, and combining it with Map Reduce framework. The results prove that G-DBSCAN and MRG-DBSCAN algorithm are feasible and effective.

***Key words**: cluster analysis, DBSCAN, Grid, G-DBSCAN, Map Reduce*

1. Introduction

Data mining [1], also known as knowledge discovery, it was from a large number of incomplete, noisy, fuzzy, random data, extract potentially useful information and knowledge processes. The excavated knowledge can be applied to information management, decision support, process control, and many other applications [2, 3]. Now, data mining is one of the hot research fields in Computer Science.

Clustering analysis is an important part of data mining [4-6]. Clustering is the dataset into multiple classes, and has a high degree of similarity between objects in the same class, the larger the gap between objects of different classes. By clustering, people can identify the dense and sparse regions. Therefore, we can find the relationship between the data global distribution and data attribute. The traditional cluster analysis methods are mainly five kinds, namely partitioning clustering methods, hierarchical clustering methods, density-based clustering methods, grid-based clustering methods and model-based clustering methods. At present, the cluster analysis has been widely used in pattern recognition, data analysis, image processing, market research, machine learning, image segmentation, information retrieval, biology and other fields [7, 8].

Density-based method clustering objects according to the density. It generates clusters based on the density of the neighborhood object, or some kind of density function

DBSCAN (Density - Based Spatial Clustering of Application with Noise) is a simple, effective density-based clustering algorithm [9-11]. Unlike the partitioning and hierarchical clustering method, DBSCAN defines clusters as the maximum density-reachable points. The high-density region can be divided into clusters. DBSCAN can find clusters of arbitrary shape, while fast clustering.

However, the DBSCAN algorithm is also has limitations. For example, the data size increases, algorithm requires large memory support and larger I/O consumption. Serial DBSCAN algorithm cannot handle large data, once the data quantity is too large, algorithm is slow, and even can't normal operation.

In order to solve the problem that the original DBSCAN algorithm will occupy a lot of memory, this paper proposes a DBSCAN clustering algorithm based on grid, namely G-DBSCAN. It attempts to use the grid to reduce the amount of data processed DBSCAN algorithm, while removing noise points, thereby reducing the memory footprint, improve efficiency. To solve the problem that G-DBSCAN algorithm for large data processing is slow, we parallelized the G-DBSCAN algorithm, and combining it with Map Reduce [12, 13] framework, and get MRG-DBSCAN algorithm.

2. Methods

2.1. DBSCAN Clustering Algorithm

DBSCAN algorithm is a simple, effective density-based clustering algorithm, this algorithm uses the class of density connectivity quickly discover arbitrary shape clusters. The central idea of the algorithm: For a class of each object, the object within its given radius contains not less than a specified minimum number [14, 15].

Algorithm Definitions:

Definition1. Core object: For a point P, take it as the center, taking Eps as the radius of a circle, the circle contains at least MinPts points, then the given object is the core object.

Definition2. Boundary object: In the Eps neighborhood of core objects, but do not meet the conditions of the core object, then these objects called boundary objects.

Definition3. Directly density-reachable: If P belongs to the Eps neighborhood of Q, and P is the core point, then object P is directly density-reachable from object Q

Definition4. Density-reachable: Object P is directly density-reachable from object Q, When there is a series of point P1, P2, P3……Pn=P started by Q, meet object Pi+1 is directly density-reachable from object Q, and Q is the core point, then object P is density-reachable from object Q.

Definition5. Density-connected: There exists a point O, if object P is density-reachable from object O, at the same time, object Q is density-reachable from object O, and then object P is density-connected to object Q.

Definition6. Noise points: P is neither core object nor boundary object, then it is a noise points.

The Basic Concept of DBSCAN:

Each point in the density-connected set is density-reachable. Choose any point P, if P is not classified, check that whether the P is the core point. If the point is the core point, find all points, they are directly density-reachable from object P. Form a new cluster with these points, assign an ID to each cluster. If P is a boundary object, then continue to access the next data point. Continue this process until all points have been processed. Finally, no ID points as noise points [16].

2.2. The Advantages and Disadvantages of DBSCAN

Advantages:

Algorithm can find clusters of arbitrary shapes and sizes, automatically determine the number of clusters, isolated noise points, high efficiency and one scan can complete the clustering.

Disadvantages:

In the process of clustering, DBSCAN once found the core object, then this core object as the center outward expansion, this process will continue to increase core objects, unprocessed objects are retained in memory. If a large cluster exists in the database, it will require a lot of memory to store the core object information.

When the data density is not uneven, the quality of clustering is very poor. Input parameters sensitive. Parameter Eps, MinPts difficult to determine.

2.3. G-DBSCAN Clustering Algorithm

When understanding the G-DBSCAN algorithm, we need to know some relevant definitions.

Definition1. Grid size: grid side length defined according to the actual situation

Definition2. Noise-point threshold: the data grid is below the threshold, it will be treated as noise points. Threshold is generally designated by the artificial.

Definition3. Data center of the grid X:

$$X= \left(\sum_{i=1}^{n} x_i\right)/n \qquad (1)$$

Here, n is the number of data points in each grid.

Definition4. Distance calculation formula D:

$$D=\sqrt{\sum_{i=0}^{n} (x_i - x)^2} \qquad (2)$$

Here, n is the number of data attribute values.

Improved Basic Idea:

The average time complexity of DBSCAN algorithm is O (nlogn) (n is the number of data contained in the database). Most of the clustering process time is used in data query. In fact, DBSCAN clustering algorithm is a continuous process of data query. Therefore, if reduce the number of search data, we can reduce the memory footprint, improve the speed of clustering. Here, from the view of reducing the number of initial input data, we give a fast density-based clustering algorithm.

Although DBSCAN algorithm itself can remove noise points, it will also occupy memory space when judging the noise points. This also led to the processing speed of DBSCAN algorithm is slow. In order to improve the processing speed, in view of the above problem, we improved DBSCAN clustering algorithm.

1) Remove the Noise Point

We can first use the grid method to remove part of the noise points. The data points according to their attribute values assigned to the corresponding grid. Count the number of data points in each grid and calculate its data center. In the grid data number is less than a threshold point as noise points were removed from the dataset, thereby reducing the noise points in the data set.

2) Reduce the Memory Footprint

In the first step, the data has been assigned to each grid, and counts the number of data in each grid and its data center point. After removing noise points, the rest of the grid data center will be used as input data of DBSCAN. When get the clustering results, assign each grid point to the class that contains the center of the grid. Because the data is reduced, so the DBSCAN algorithm for data processing time is reduced.

Figure 1 is the schematic diagram of the G-DBSCAN algorithm. The black point is the input data, the blue point is noise, and the red point is the center of the points in each grid. The red points instead of black spots as the algorithm input data.

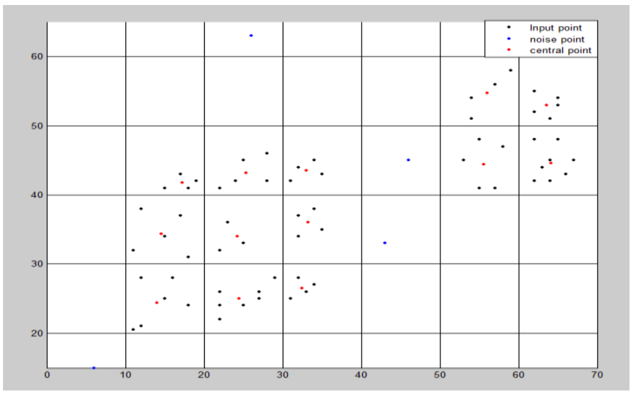


Figure 1. G-DBSCAN Algorithm Schematics

Figure 2 is an improved DBSCAN algorithm flowchart, namely G-DBSCAN algorithm flowchart.

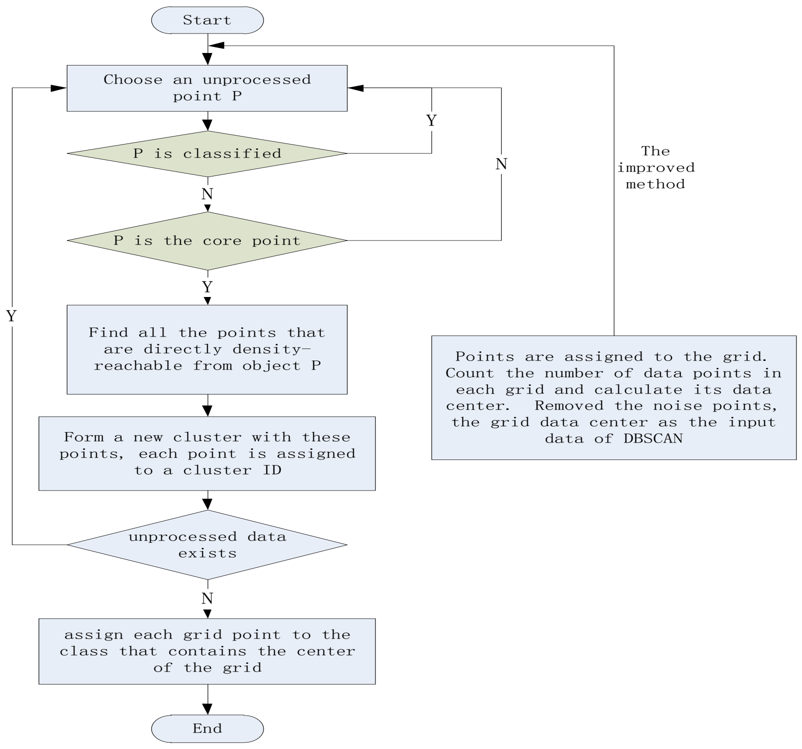

Start
Choose an unprocessed point P
P is classified
Y
N
P is the core point
N
Y
Find all the points that are directly density-reachable from object P
Form a new cluster with these points, each point is assigned to a cluster ID
unprocessed data exists
Y
N
assign each grid point to the class that contains the center of the grid
End
The improved method
Points are assigned to the grid. Count the number of data points in each grid and calculate its data center. Removed the noise points, the grid data center as the input data of DBSCAN

Figure 2. G-DBSCAN Algorithm Flowchart

The following shows the basic framework of G-DBSCAN clustering algorithm and its main pseudo code.

```
G-DBSCAN (Data, Eps, MinPts, Noise threshold, Grid size)
For each point p in Data DO
IF p. attribute>Grid.Coordinate&& p. attribute<Grid.Coordinate+ Grid.size THEN
p.ID=Grid.ID
END IF
END FOR
For each point p in Data DO
        Grid.CentralPoint = Sum (p, Grid.ID)/N   // Find the center of each grid, N is
    the number of each grid
END FOR
     IF Grid.number< Noise threshold THEN // Grid.number is the number of each
  grid
Delete (Grid.points) // Delete noise points,
Data1=Data - NoisePoints.number
END IF
DBSCAN (Grid.CentralPoint, Eps, MinPts)   //  The results with DBSCAN
IF p.ID= Grid.CentralPoint.ID THEN // p and Grid.CentralPoint in the same grid
P.ID= cluster.ID  // p assigned to the cluster which ctains the Grid.CentralPoint
END IF
```

2.4. G-DBSCAN Algorithm Parallelization (MRG-DBSCAN)

2.4.1. MapReduce Working Mechanism

MapReduce is a distributed data processing model to run on large clusters. Mapreudce and HDFS (Hadoop Distributed File System) is the two core of Hadoop distributed system. HDFS is responsible for managing files. MapReduce is responsible for processing data. MapReduce is divided into two stages: Map phase and the Reduce phase, and each stage with key-value as input and output. Hadoop input data are divided into small data blocks, create a Map task for each data block. The data will be entered in the form of key-value, Map function processing ends, MapReduce framework will be sorted the Map output data by key, and then enter into Reduce task. Those data with the same key will be sent to the same Reduce. Reduce task will merge those data with the same key into one data. So, algorithm parallelization is realized Map and Reduce function. Figure 3 depicts the MapReduce working mechanism.

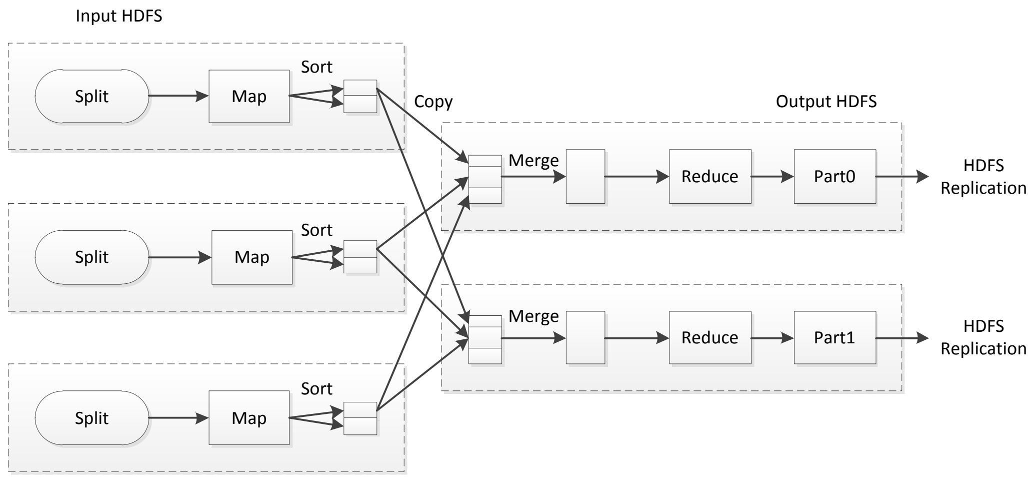


Figure 3. Map Reduce Data Flow Diagram

2.4.2. MRG-DBSCAN Algorithm

G-DBSCAN algorithm is divided into two parts: Center points generate parallelization and DBSCAN algorithm parallelization.

1) Center points generate parallelization

Map function design

The Map function's task is to divide the grid, and the data according to the attribute value assigned to the grid. Each point is converted into <key, value>pairs, where key is the starting offset of the point, value is the attribute values of the point. The output is <Grid ID, attribute values>.

Map function pseudo code

```
Input: Data sets, Grid size d
Output: <Grid ID, value>
FOR each point p in DB DO
Grid ID=value/d;
Output< Grid ID, values>
END FOR
```

Reduce function design

Enter all < Grid ID, attribute values >, if the point has the same Grid ID, adding their attribute values, and record the number of those data points. The sum of attribute values divided by the number of data points is the center point of each grid. Compare the number of data in each grid, if the number of data point in the grid is less than the threshold value, then removed those points from the data set.

Reduce function pseudo code

```
Input: <Grid ID, value>, Isolated points threshold a
Output: <Grid ID, center point value>
FOR each < Grid ID, value> DO
TotalValue+=value;   // The sum of all value
Num++;          // Calculating the number of data points in each grid
END FOR
IF Num>a THEN
AvgValue =TotalValue/Num;   // Computing Center Point
Output< Grid ID, (AvgValue, Num)>
END IF
```

2) DBSCAN parallelization

Map function design

Map function main task is to identify the type of data points, and data partitions. Data types include core points and border points. Enter < Grid ID, (AvgValue, Num)>, data point in accordance with AvgValue size into various partitions. Each partition overlap, overlap length 2Eps. The overlapping area is used to merge clusters.

Map function pseudo code

```
Input: < Grid ID, (AvgValue, Num)>   X,  Eps (X is the length of partition)
Output: <P, ( (AvgValue, Num))>  (P is the partition ID)
FOR each < Grid ID, (AvgValue, Num)> DO
P= AvgValue/X;        // P is the partition ID
Q= AvgValue%X;      // whether the data point in the overlap region
END FOR
IF (Q>0&&Q<Eps)    // the data point in the previous overlap region
Output1<P-1, (AvgValue, Num)>
Output2<P, (AvgValue, Num)>
```

```
ELSE IF (Q>(X-Eps)&&Q<X)   // the data point in another overlap region
Output1<P, (AvgValue, Num)>
Output2<P+1, (AvgValue, Num)>
ELSE                    // Point is not in the overlap region
Output<P, (AvgValue, Num)>
```

Reduce function design

DBSCAN clustering performed on each partition. After clustering is completed, merging clusters. Merging rules: If the data point P in the overlapping area and it is the core point, then combined the cluster contains the core point p. If p is not a core point, be assigned to the nearest cluster. Final, output <Cluster ID, value>.

Map function pseudo code

```
Input：<P, (AvgValue, Num)>, Eps, Minpits
Output：<Cluster ID,  AvgValue >
FOR each <P, (AvgValue, Num)> DO
List ds.add(AvgValue);
END FOR
DBSCAN (ds);
Merge cluster;
END IF
```

3. Experimental Analysis

In this paper, the traditional DBSCAN algorithm and G-DBSCAN algorithm are compared. Test data sets taken from UCI database. UCI is a specialized database for testing machine learning, data mining algorithms. The data in the library have a certain classification, so it can be used to test the quality of clustering. In the UCI database, we selected Iris, Wine, Glass and Indian data sets to test, data set information as shown in Table 1. Where Number is the number of data, Attributes is the number of data attribute. In order to verify the improved algorithm, the distribution of test data remains unchanged.

In this paper, the original DBSCAN algorithm and the G-DBSCAN algorithm experiments were carried out four times. The noise threshold of G-DBSCAN algorithm is 1, that is, when the number of data in the grid is 1, we think it is a noise point, and removing it from the dataset.

Table 1. Dataset Information

Datasets	Number	Attributes
Iris	150	4
Wine	178	13
Glass	214	10
Indian	768	8

Table 2 is a comparison of the experimental results, where Number is the number of data processing, Memory is memory footprint and Time is the time taken to achieve clustering.

Table 2. Experimental Results Comparison

Datasets	DBSCAN			Grid-based DBSCAN		
	Number	Memory	Time	Number	Memory	Time
Iris	150	1.625696 MB	46ms	27	0.975408 MB	10ms

Wine	178	2.276008 MB	114ms	51	1.300592 MB	27ms
Glass	214	2.279256 MB	133ms	16	0.975432 MB	11ms
India n	768	10.170304 MB	368ms	58	1.300616 MB	44ms

As can be seen from Table 2, the original DBSCAN algorithm has a lot of input data, takes up more memory and long running time. The G-DBSCAN algorithm is to divide some similar points to the same grid, after removing noise points, the rest of the grid data center will be used as input data of DBSCAN. When get the clustering results, assign each grid point to the class that contains the center of the grid and achieved the purpose of reducing the amount of input data. Data show that G-DBSCAN algorithm greatly reduces the memory footprint and program running time.

After that we conducted MRG-DBSCAN algorithm experiment. To test the effect of different number of nodes on algorithm performance, we choose 20000, 40000, 80000, 160000 and 320000 data for testing. Due to the limited experimental conditions, we only have four nodes, one master node, and three slave nodes. Figure 4 is the test result of each node.

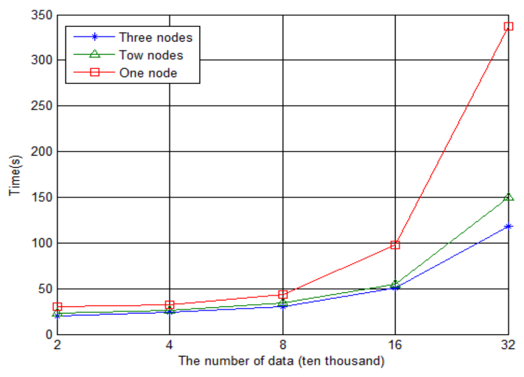


Figure 4. Node Test Chart

During the experiment the algorithm is stable. As shown in Figure 4, the node increases the reduction in running time. This shows that the MRG-DBSCAN algorithm has faster processing speed than the G-DBSCAN algorithm, and with the increase of the node, the algorithm processing data faster.

Usually, there are three kinds of methods to evaluate parallel algorithm, here we use the speedup. Speedup evaluation method is to keep the data unchanged, increasing the number of computers. Figure 5 is the speedup of MRG-DBSCAN algorithm.

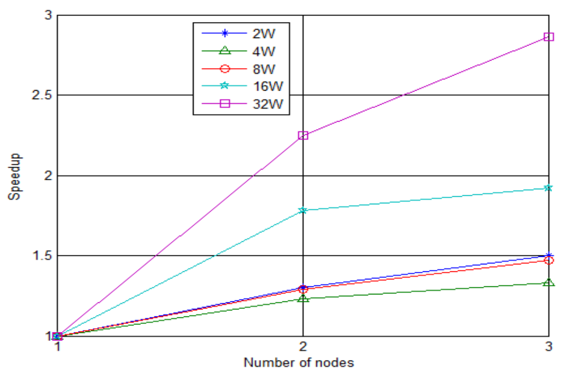


Figure 5. Speedup Chart

As can be seen from Figure 5, increasing the number of computers, Speedup maintains linear growth. MRG-DBSCAN algorithm has good speed up.

4. Conclusion

Now, spatial database is large in scale, and contains a large amount of information. The main task of data mining is to find useful information from the complex spatial database. Clustering analysis has been a hot research topic in data mining. Clustering is found similar data sets from large amounts of data. In this paper, through the analysis of the DBSCAN clustering algorithm, in view of its weaknesses, using the grid method to extend its performance so that it can effectively deal with large-scale spatial database. The G-DBSCAN algorithm parallelization, called MRG-DBSCAN algorithm to achieve large data processing. Experimental results show that the algorithm is feasible and effective. With the increasing scale of spatial database, data information becomes more and more complex. So in many applications it is difficult to select the appropriate clustering parameters. Therefore, the development of adaptive clustering algorithm will become an important part of our future research.

Acknowledgements

This paper is a revised and expanded version of a paper entitled "G-DBSCAN: an Improved DBSCAN Clustering Method Based on Grid" presented at FGCN 2014, Hainan China, December 20-23, 2014. This work is supported by the NSFC (61402234, 61103142). It was also supported by the Industrial Strategic Technology Development Program (10041740) funded by the Ministry of Trade, Industry and Energy (MOTIE) Korea.

References

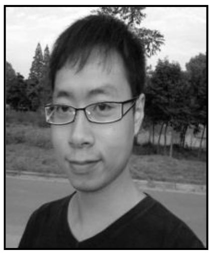
[1] C. Apte, "Data mining: an industrial research perspective", Computational Science & Engineering, vol.4, no.2, **(1997)**, pp. 6 – 9.
[2] L. Cao, H. Zhang, Y. Zhao, D. Luo and C. Zhang, "Combined Mining: Discovering Informative Knowledge in Complex Data, 2011,vol.41, no.3, pp.699-712.
[3] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay and C. A. Coello, "A Survey of Multi objective Evolutionary Algorithms for Data Mining: Part I, Evolutionary Computation, vol.18, no.1, **(2014)**, pp.4-19.
[4] X. Tang and P. Zhu, "Hierarchical Clustering Problems and Analysis of Fuzzy Proximity Relation on Granular Space, Fuzzy Systems, vol. 21, no.5, **(2013)**, pp. 814 - 824.
[5] O. Wu, W. Hu, S. J, Mingliang and Z. B. Li, "Efficient Clustering Aggregation Based on Data Fragments, Systems, Man, and Cybernetics", Part B: Cybernetics, vol. 42, no. 3, **(2012)**, pp. 913 – 926.
[6] J. Huang, H. Sun, Q. Song, H. Deng and J. Han, "Revealing Density-Based Clustering Structure from the Core-Connected Tree of a Network", Knowledge and Data Engineering, vol. 25, no.8, **(2013)**, pp. 1876 - 1889.
[7] Y. Chu, J. Huang, K. Chuang, D. Yang and M. Chen, "Density Conscious Subspace Clustering for High-Dimensional Data", Knowledge and Data Engineering, vol. 22, no. 1, **(2010)**, pp.16-30.
[8] J. Hou, E. Xu, W. Liu, Q. Xia and N. Qi, "A density-based enhancement to dominant setsclustering", Computer Vision, vol. 7, no. 5, **(2013)**, pp. 354-361.
[9] B. Jiang, J. Pei, Y. Tao and X. Lin, "Clustering Uncertain Data Based on Probability Distribution Similarity", Knowledge and Data Engineering, vol.25, no.4, **(2013)**, pp.751-763.
[10] M. Kryszkiewicz and P. Lasek, "TI-DBSCAN: Clustering with DBSCAN by Means of the Triangle Inequality", Rough Sets and Current Trends in Computing, vol. 6086, **(2010)**, pp.60-69.
[11] A. Smiti and Z. Elouedi Z, "DBSCAN-GM: An improved clustering method based on Gaussian Means and DBSCAN techniques", Intelligent Engineering Systems (INES), **(2012)**, pp.573 – 578.
[12] Y. He , H. Tan , W. Luo, H. Mao, D. M, S. Feng and J. Fan, "MR-DBSCAN: An Efficient Parallel Density-Based Clustering Algorithm Using MapReduce", Parallel and Distributed Systems (ICPADS), **(2011)**, pp.473 – 480.
[13] L. Lingjuan and X. Yang, "Research on Clustering Algorithm and Its Parallelization Strategy", Computational and Information Sciences (ICCIS), **(2011)**, pp.325 – 328.
[14] A. Thom and O. Kramer, "Acceleration of DBSCAN-Based Clustering with Reduced Neighborhood Evaluations", KI 2010: Advances in Artificial Intelligence, vol. 6359, **(2010)**, pp.195-202.

[15] A. Smiti and Z. Elouedi, "DBSCAN-GM: An improved clustering method based on Gaussian Means and DBSCAN techniques", Intelligent Engineering Systems, **(2012)**, pp.573-578.
[16] M. Huang and F. Bian, "A Grid and Density Based Fast Spatial Clustering Algorithm", Artificial Intelligence and Computational Intelligence, **(2009)** , pp. 260 – 263.

Authors

Li Ma, she received her B.S. degree in 1985 from the Chengdu Institute of Meteorology and her Ph. D degree in 2010 from Nanjing University of Information Science and Technology. She is a professor and tutor for graduates in Nanjing University of Information Science and Technology. Her main research interests include image processing, pattern recognition, and meteorological information processing and data assimilation.

Lei Gu, he obtained his B.S. degree in the Computer and Software Institute from Nanjing University of Information Science and technology, China in 2012. Now, he is working toward the M.S. degree in the Computer and Software Institute. His main research interests include Data mining and cloud computing.



Bo Li, he was born on May 12, 1987. Currently He is a master in Nanjing University of Information Science and Technology. He received his bachelor degree in Chongqing Normal University. His areas of interest are short-term wind power prediction, meteorological information processing and data assimilation. Now, he works at the CMA Research Centre for Strategic Development.

Shouyi Qiao, he was born on May 22, 1993. Currently He is a student in Nanjing University of Information Science and Technology.