International Journal of Grid Distribution Computing
Vol.8, No.1 (2015), pp.189-200
http://dx.doi.org/10.14257/ijgdc.2015.8.1.18

An Improved K-means Algorithm based on Mapreduce and Grid

Li Ma[1, 2, 3], Lei Gu[1, 2], Bo Li [1, 4], Yue Ma[5] and 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*
[5] *School of Mathematics and Statistics, Nanjing University of Information Science & Technology, Nanjing 210044*

Abstract

The traditional K-means clustering algorithm is difficult to initialize the number of clusters K, and the initial cluster centers are selected randomly, this makes the clustering results very unstable. Meanwhile, algorithms are susceptible to noise points. To solve the problems, the traditional K-means algorithm is improved. The improved method is divided into the same grid in space, according to the size of the data point property value and assigns it to the corresponding grid. And count the number of data points in each grid. Selecting M(M>K) grids, comprising the maximum number of data points, and calculate the central point. These M central points as input data, and then to determine the k value based on the clustering results. In the M points, find K points farthest from each other and those K center points as the initial cluster center of K-means clustering algorithm. At the same time, the maximum value in M must be included in K. If the number of data in the grid less than the threshold, then these points will be considered as noise points and be removed. In order to make the improved algorithm can adapt to handle large data. We will parallel the improved k-mean algorithm and combined with the MapReduce framework. Theoretical analysis and experimental results show that the improved algorithm compared to the traditional K-means clustering algorithm has high quality results, less iteration and has good stability. Parallelized algorithm has a very high efficiency in data processing, and has good scalability and speedup.

Keywords: *Cluster analysis, K-means, Grid, DBSCAN, MapReduce*

1. Introduction

Data mining is the process of automatically discovering useful information in large data repositories. Data mining techniques are deployed to scour large databases in order to find novel and useful patterns that might otherwise remain unknown. Data mining is one of the hot research areas in computer science [1].

Clustering analysis is a very important branch in data mining. Cluster analysis based on the data objects and their relationships, grouping the data objects. The goal is the same group object is similar to each other, while objects in different groups are different. If the greater similarity within the group, the greater difference between the two groups, then the clustering effect is better [2-4].

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. In various clustering

ISSN: 2005-4262 IJGDC

methods, K-means clustering method is extensively used [4-6]. It's a kind of partitioning clustering methods. K-means clustering algorithm is a typical distance-based clustering algorithm, using distance as similarity evaluation. K-means algorithm is simple and fast, with a more intuitive geometric meaning. It has been widely used in pattern recognition, image processing and computer vision. At the same time, the satisfactory results are obtained [7].

However, this algorithm is also some problems. For example, k value cannot be determined, the initial cluster centers are randomly selected and the presence of noise points, lack of big data processing capabilities, etc. In order to improve the K-means algorithm, the researchers propose a variety of solutions. There are typical Moh. DB. Al Daoud and Stuart A. Roberts proposed density estimation method based on the division [8]. Kaufman [9] proposed by local density of data points to estimate the initial cluster center. In order to avoid the center of a number of initial classes belong to the same category J.T. Tou et al. [10] methods are proposed to limit the initial class center distance. These approaches improve the performance of the K-means algorithm to a certain extent, but their improvement can only solve one of the problems mentioned above. In the cluster accuracy, cluster quality and the ability to process big data remains to be improved further.

In this article, presents a method to improve the K-means algorithm. Try to use the grid and the DBSCAN algorithm to improve the efficiency of K-means. First, we have to grasp the distribution of data points through the grid. Then calculate k values by DBSCAN algorithm. Finally, select the k initial cluster centers by the grid method and remove noise points. This makes the clustering results more accurate, algorithm has better stability. To overcome the problem of big data processing, we will parallelize the improved k-means algorithm, and combining it with MapReduce [11-13] framework.

2. Relevant Methods

2.1 K-means Clustering Algorithm

The basic idea of K-means algorithm [14-16]: Randomly selected K objects, each object as a cluster center. For each remaining object according to its distance from the center of each cluster, assign it to the nearest cluster. And then calculate the average of each cluster. This process is repeated until the objective function converges. Here the objective function is the sum of the distance of each point to its center.

K-means algorithm process:

Input: clustering number k, data sets.

Output: K cluster.

Algorithm steps:

Step 1: Randomly selected k objects from a data set, each object as an initial cluster center.

Step 2: The other data object is assigned to the nearest cluster.

Step 3: The average of all the data in each cluster as new center of this cluster.

Step 4: Repeat steps 2 and 3 until the objective function converges.

2.2 K-means Algorithm Advantages and Disadvantages:

Advantage:

The algorithm is simple, for large-scale data mining with high efficiency and scalability. Its complexity is O (n), near-linear time complexity. When the clustering of data is dense (convex), and the difference between the clusters of data is large, the effect of K-means clustering algorithm is better.

Disadvantages:

(1) The algorithm for convex class treatment effect is very good, but not easy to find a class of non-convex.
(2) Treatment effect of the algorithm depends on k worthy choice.
(3) The input sequence to the initial clustering centers and sample sensitive. For different initial cluster centers and sample input sequence, clustering results will be very different.
(4) The algorithm only when numeric data can be applied, is not available when it comes to the classification attribute data sets.
(5) The initial clustering center is random.
(6) The effect of the algorithm is much affected by noise.

2.3 Improved method of K-means algorithm

When understanding the improved K-means 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:

$$X=(\sum_{i=1}^{n} X_i)/n \quad (1)$$

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

Definition4. Distance calculation formula D:

$$D=\sqrt{\sum_{i=0}^{n}(Xi-X)^2} \quad (2)$$

Here, n is the number of data attribute values.

Definition5. Objective function J:

$$J=\sum_{i=1}^{k}\sum_{x_\lambda \in classi}\sqrt{\sum_{j=1}^{q}(X_{\lambda,j}-X_{i,j})^2} \quad (3)$$

Where i is the number of classes, λ is the number of data in each class, j is the number of data attribute values.

Definition6. Accurate rate W:

$$W=\frac{M}{N}*100\% \quad (4)$$

M is the number of data that can be assigned to the correct class. N is the total number of data.

Improvement Ideas:

The traditional K-means algorithm uses a randomly selected point as the initial centers. This causes instability clustering. Clustering result is affected by noise points, leading to inaccurate results. In order to ensure the K-means clustering algorithm stability and convergence, we will reduce the noise point number. For initial centers are randomly selected, the presence of noise points and k values of uncertainty, K-means clustering algorithm to improve.

(1) Remove the Noise Point

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, so as to ensure the accuracy of clustering.

(2) Determine the Value of k

In the first step, data has been assigned to the respective grid, each grid computing the number of data and the data center. The rest of the grid data center will be used as input data of DBSCAN algorithm [17, 18], which generates cluster k value.

(3) Determine the Initial Centers

After the k value is determined, Choose M grids, their maximum number of data. In the M center points, find K points farthest from each other and those K center points as the initial cluster center of K-means clustering algorithm. At the same time, the maximum value in M must be included in K.

The improved K-means algorithm flow:

Input: Data sets, grid size, noise point threshold

Output: K cluster.

Algorithm steps:

Step 1: Determine the size of the grid according to the actual situation, according to the size of the data attribute values to assign data to the grid.

Step 2: Calculate each grid data quantity and data center.

Step 3: According to the noise threshold to remove noise point.

Step 4: The rest of the grid center point as input data of DBSCAN to obtain K value.

Step 5: After the k value is determined, Choose M grids (M>K), their maximum number of data. In the M center points, find K points farthest from each other and those K center points as the initial cluster center of K-means clustering algorithm.

Step 6: According to the initial center point, distributes the data to the nearest class.

Step 7: Calculating the average of all the data in each class as the new center.

Step 8: Repeat steps 6 and 7 until the objective function converges.

Figure 1 is an improved K-means algorithm flowchart.

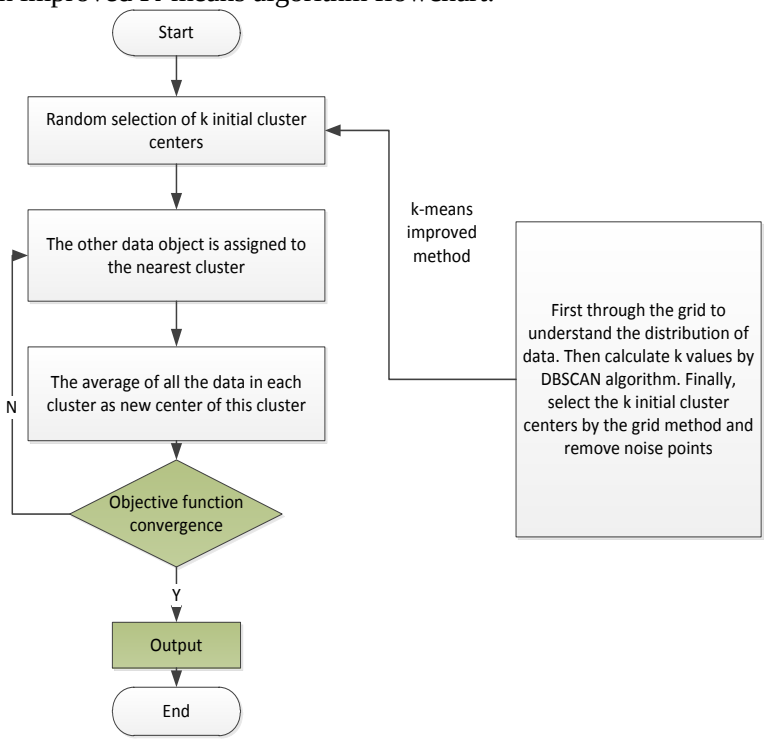


Figure 1. Algorithm Flow Chart

2.4 Improved k-means Algorithm Parallelization

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 2 depicts the MapReduce working mechanism.

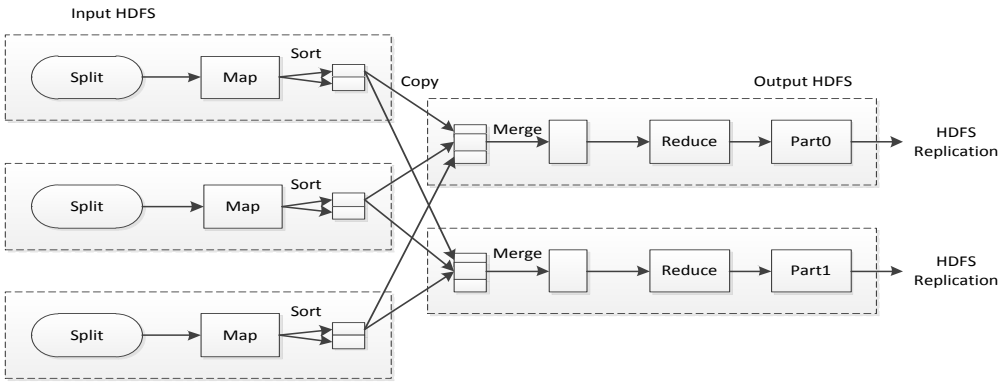


Figure 2. MapReduce Data Flow Diagram

2.4.2 Algorithm Parallelization

Parallelized algorithm is divided into two parts: one is Initial center points generate parallelization, and the other is k-means algorithm parallelization.

1) Initial Center Points Generate Parallelization

Map Function Design

Map function is to assign each data to the corresponding grid and mark the grid ID. 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 side length 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 key, 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, select the data points up to k grid and record its center point, while the number of data points in the grid is less than the threshold value are removed 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 List MaxNum.size<k    // k is the number of clusters
    List MaxNum.add(Num);   //List MaxNum is a list
ELSE IF Num>min(MaxNum)
    List MaxNum.remove(min(MaxNum));
    List MaxNum.add(Num);
END IF
IF Num>a THEN
    AvgValue =TotalValue/Num;   // Computing Center Point
    Output< Grid ID, (AvgValue, Num)>
END IF
```

2) K-means Algorithm Parallelization

Map Function Design

First, enter all the data and cluster centers. The input data is converted into key-value pairs, where key is the starting offset of the point and value is the attribute values of the point. Calculate the distance between data point and those k center points, find the nearest center point and record the ID as key2. Output is <key2, value>.

Map Function Pseudo Code

```
Input: the processed data sets NewDB
Output: <key2, value>
FOR each point p in NewDB DO
    Dis= min((value-centerpoint)* (value-centerpoint) )     // find the nearest center
point
    Output<key2, value>    // key2 is the ID of  the nearest center point
END FOR
```

Reduce Function Design

Points with the same key2, and added its values as TotalValue, at the same time, calculate the number of those points and get Num. TotalValue divided by Num to get a new cluster centers. Updated cluster centers and beginning the next iteration. When the center point does not change, stop iteration, algorithm terminates.

Reduce Function Pseudo Code

```
Input: <key2, value>

Output: < Centerpoint, value>

FOR each point p<key2, value> DO

    TotalValue+=value;
```


Num++; // Calculating the number of points having the same key2

END FOR

Centerpoint= TotalValue/Num; // Updated cluster centers

Output< Centerpoint, value>

3. Experimental Analysis

In this paper, in order to verify the validity of the improved K-means clustering algorithm, the traditional K-means algorithm and improved K-means algorithm were compared. Test data is UCI database Iris data set. UCI is a public database for testing machine learning, data mining algorithms. The data have a certain classification. Therefore, the accuracy rate can be used to visually represent the quality of clustering.

The most commonly used to detect algorithm quality data set is Iris. Iris data set which contains 150 data, each data contains four attributes. In order to verify the improved algorithm, the distribution of test data remains unchanged.

In this paper, the original K-means algorithm and improved K-means algorithm experiments were carried out six times, where J is the objective function and W is accuracy. Specific results are shown in Table 1 and Table 2.

Table 1 is the relative parameters of the improved K-means algorithm, and some part of the clustering results. It can be seen from the table that the grid size is set to 0.67, the noise threshold is 1. K value calculated by the DBSCAN algorithm is 3, the number of noise points is 26.

Table 1. Improved K-means Algorithm Data Values (data: Iris)

number of data	Grid size	Noise threshold	K	number of noise points
150	0.67	1	3	26

As can be seen from Table 2, the initial center of the original K-means algorithm is randomly selected, so six experimental results are different and the results of volatile. Specific performance for the results of the objective function J is uncertain. The improved K-means algorithm to determine the initial centers due to the grid method, so clustering results unchanged. Because of the removal of the 26 noise, so the value of the objective function is relatively small, but the accuracy of clustering has been greatly improved. In 6 experiments, the accuracy of the original K-means algorithm for a maximum of 90%, at least 81.3%, the accuracy results of 6 tests did not exceed the improved algorithm. And the DBSCAN algorithm can also be a good way to determine the K value. The experimental results show that the proposed algorithm is very good to determine the value of K, successfully solved the problem that the original k - means algorithm is sensitive to the initial center and affected by noise.

Table 2 Comparison of Objective Function (data: Iris)

N	K-means algorithm			Improved K-means algorithm		
	Initial centers	J	W	Initial centers	J	W
1	(5.0,2.3,3.3,1.0) (6.1,3.0,4.9,1.8) (6.4,2.7,5.3,1.9)	106.18175	84%	(5.0625,3.55 1.55, 0.275) (6.30,2.93,4.97 ,1.76) (6.825, 3.15 ,5.675 ,2.2625)	77.94432	91.33%
2	(4.6,3.1,1.5,0.2) (6.2,2.9,4.3,1.3) (4.9,2.5,4.5,1.7)	99.100876	90%	(5.0625,3.55 1.55, 0.275) (6.30,2.93,4.97 ,1.76) (6.825, 3.15 ,5.675 ,2.2625)	77.94432	91.33%
3	(4.9,2.5,4.5,1.7) (6.3,2.5,4.9,1.5)	105.07447	87.3%	(5.0625,3.55 1.55, 0.275) (6.30,2.93,4.97 ,1.76)	77.94432	91.33%

	(6.1,2.6,5.6,1.4)			(6.825, 3.15 ,5.675 ,2.2625)		
4	(6.3,3.4,5.6,2.4) (6.3,2.9,5.6,1.8) (6.1,2.6,5.6,1.4)	117.07054	82.6%	(5.0625,3.55 1.55, 0.275) (6.30,2.93,4.97 ,1.76) (6.825, 3.15 ,5.675 ,2.2625)	77.94432	91.33 %
5	(7.9,3.8,6.4,2.0) (4.8,3.4,1.6,0.2) (4.5,2.3,1.3,0.3)	118.28588	81.3%	(5.0625,3.55 1.55, 0.275) (6.30,2.93,4.97 ,1.76) (6.825, 3.15 ,5.675 ,2.2625)	77.94432	91.33 %
6	(6.3,3.4,5.6,2.4) (5.5,2.4,3.8,1.1) (6.4,2.8,5.6,2.1)	113.4145	82%	(5.0625,3.55 1.55, 0.275) (6.30,2.93,4.97 ,1.76) (6.825, 3.15 ,5.675 ,2.2625)	77.94432	91.33 %

After that we conducted parallel algorithm experiment. To test the effect of different number of nodes on algorithm performance, the data set is increased while increasing the number of nodes. This experiment, we choose 5000, 10000, 20000, 40000 data for testing. Due to the limited experimental conditions, we only have four nodes, one master node, and three slave nodes.

Table 3 shows the test results for each node. T1, T2, T3 respectively for one node, tow nodes, three nodes average time of iteration.

Table 3 Data Processing Time

Data set (10^5)	Iterations	T1	T2	T3
0.5	2	51sec	43sec	35sec
1	8	59sec	49sec	41sec
2	4	1mins 41sec	1mins 20sec	1mins 6sec
4	8	5mins 26sec	3mins 56sec	2mins 5sec

Figure 3 is the test result of each node, figure 3 is a graphical representation of the table 3.

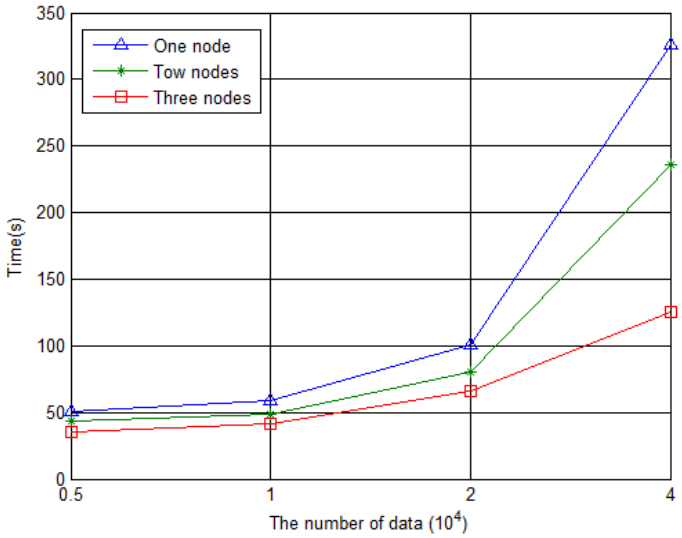


Figure 3. Node Test Chart

As shown in figure 3, when the number of nodes and process data size is proportional to the growth, MapReduce processing of data remained on the same level. Parallel algorithms exhibit good scalability.

Speedup is used to measure the performance of parallel algorithms. Equation 3 is speedup calculation formula.

(5)

T1 is the computation time on a computer, Tp is the computation time on p computers. Figure 4 is the speedup of the parallel algorithm.

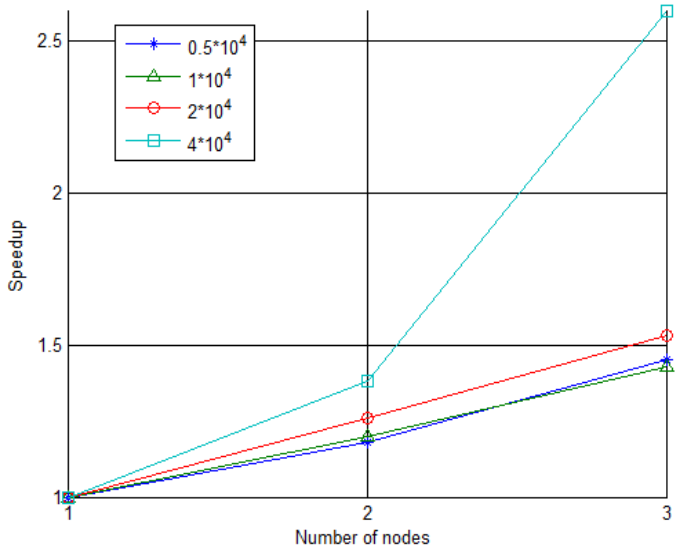


Figure 4. Speedup Chart

4. Conclusion

Selection of the initial center point, determined to remove noise points and the number of clusters will influence the clustering results of K-means algorithm. This article will address these three issues. By assigning the data to the corresponding grid, to achieve the determination of initial center and noise removal, and through the DBSCAN algorithm to determine the K value, thus the traditional K-means algorithm has been improved. Compared with the traditional K-means algorithm, the improved algorithm successfully determined the initial clustering centers, which solves the problem of the K-means initial center point sensitive. Remove the noise points, so as to improve the clustering accuracy, and to determine the number of clusters. Through the improved algorithm parallelization, achieved large data processing. Both algorithms improve results is satisfactory and K-means algorithm performance has been greatly improved.

Acknowledgements

This paper is a revised and expanded version of a paper entitled “An improved grid-based k-means clustering algorithm” presented at FGCN 2014, Hainan China, December 20-23, 2014. This work is supported by the NSFC (61402234, 61103142). It was also supported by Industrial Strategic Technology Development Program (10041740) funded by the Ministry of Trade, Industry and Energy (MOTIE) Korea.

References

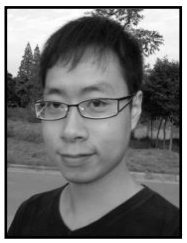
[1] P. N. Tan, M. Steinbach and V. Kumar, “Introduction To Data Mining”, **(2005)**.
[2] A. K. Jain, M. N. Murty and P. J. Flynn, “Data clustering: A review”, ACM Comput. **(1999)**, pp. 264–323.
[3] R. O. Duda, P. E. Hart and D. G. Stork, “Pattern Classification”, **(2001)**.
[4] S. Lloyd, “Least Squares Quantization in PCM”, Information Theory, **(1982)**, pp. 129-137
[5] S. N. Sulaiman and N. A. M. Isa, “Adaptive Fuzzy-K-means Clustering Algorithm for Image Segmentation” Consumer Electronics, **(2010)**, pp. 2661 - 2668
[6] J. B. Macqueen, “Some Methods of Classification and Analysis of Multivariate Observations”, Proc. Fifth Berkeley Symp. Math, Statistics and Probability, **(1967)**, pp. 281-297.
[7] C. Boutsidis and M. Magdon-Ismail, “Deterministic Feature Selection for K-means Clustering”, Information Theory, **(2013)**, pp. 6099 – 6110.
[8] M. B. Daoud and S. A. Roberts, “New methods for the initializat ion of clusters”, Pattern Recognition Letters, **(2001)**, pp. 451- 455.

[9] L. Kaufman and P. J. Rousseeuw, "Finding groups in data: an introduction to cluster analysis", **(1990)**.
[10] J. He, M. Lan, C. L. Tan, "Initialization of cluster refinement algorithms: a review and comparative study".
[11] A. Kumar, M. Kiran and B. R. Prathap, "Verification and validation of MapReduce program model for parallel K-means algorithm on Hadoop cluster", Computing, Communications and Networking Technologies (ICCCNT), **(2013)**, pp. 1 – 8.
[12] Q. Liao, F. Yang and J. M. Zhao, "An improved parallel K-means clustering algorithm with MapReduce, Communication Technology (ICCT)".
[13] R. Bhavani, G. S. Sadasivam and R. Kumaran, "A novel parallel hybrid K-means-DE-ACO clustering approach for genomic clustering using MapReduce", Information and Communication Technologies, **(2011)**, pp. 132 - 137
[14] S. Yu, L. C. Tranchevent and X. H. Liu, "Optimized Data Fusion for Kernel K-means Clustering", Pattern Analysis and Machine Intelligence, **(2012)**, pp. 1031 – 1039.
[15] Y. M. Yu and R. C. Lo, "Recognition of various tactile stimuli using independent component analysis and K-means", **(2010)**, pp. 630 - 639
[16] T. W. Chen, C. H. Sun, H. H. Su, S. Y. Chien, D. Deguchi, I. Ide and H. Murase, "Power-Efficient Hardware Architecture of K-means Clustering With Bayesian Information Criterion Processor for Multimedia Processing Applications", Emerging and Selected Topics in Circuits and Systems, **(2011)**, pp. 357 - 368
[17] M. Tekbir, S. Albayrak, "Recursive-Partitioned DBSCAN", Signal Processing and Communications Applications Conference, **(2010)**.
[18] T. Ali, S. Asghar and N. A. Sajid, "Critical analysis of DBSCAN variations", Information and Emerging Technologies, **(2010)**, pp. 1 – 6.

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 candidate 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 CMA Research Centre for Strategic Development.

Jin Wang. Dr. Jin Wang received the B.S. and M.S. degree in the Electronical Engineering from Nanjing University of Posts and Telecommunications, China in 2002 and 2005, respectively. He received Ph.D. degree in the Ubiquitous Computing laboratory from the Computer Engineering Department of Kyung Hee University Korea in 2010. Now, he is a professor in the Computer and Software Institute, Nanjing University of Information Science and technology. His research interests mainly include routing method and algorithm design, performance evaluation and optimization for wireless ad hoc and sensor networks. He is a member of the IEEE and ACM.