International Journal of Database Theory and Application
Vol.9, No.8 (2016), pp.79-88
http://dx.doi.org/10.14257/ijdta.2016.9.8.09

Research on K-MEANS Clustering Algorithm Based on HADOOP

Feng Hu

Qiongtai Teachers College, Haikou 570100, china,
272800588@qq.com

Abstract

This paper proposes an improved clustering algorithm on the basis of the characteristics of sampling and density. The initial k value and initial center are determined by sampling and density, and parallel improvement is based on the HADOOP platform. Through the experiment, the improved K-Means algorithm has good parallelism.

Keywords*: Hadoop, MapReduce, K-Means, Collaborative Filtering*

1. Introduction

With development of Internet, traditional data mining algorithm was not able to adapt to the discovery of massive information [1-2]. Here we improved traditional data mining algorithm with most recent cloud computing technology, increasing the parallelized processing capability of it through Hadoop platform. Clustering K-means algorithm was improved as well [3-4]. For k value and initial central point that it relies on, we propose the improved K-means algorithm based on sampling and density and do parallelized improvement to make it run on Hadoop platform. The improvement was made with features of Hadoop platform about the dependence of K-means algorithm on initial k value and initial central point. Before K-means algorithm clustering, Hadoop is used to take sample of initial data; then clustering is made after initial central points are determined with neighborhood density [5].

2. Idea of K-Means Algorithm

K-means algorithm [6] is a clustering analysis method. It divides n samples into k clusters, with higher similarity intra objects and lower similarity between cluster and cluster.

User decides the number k of cluster and chooses randomly k points as initial point, with each initial point as a cluster; then partition other points of samples to the nearest cluster by distance formula or other similarity calculation formula; then calculate the mean value of all objects in the cluster and use it as new central point. Repeat iterations till the objective function converges. K-means algorithm is characteristic of estimating whether sampling points are distributed to the closest clustering center during each iteration. If it's wrong distribution, it needs to adjust to relative clustering center; if it's distributed correctly, it requires no adjustment.

2.1. Procedure of the Algorithm

K-means algorithm adopts classifying criteria *e.g.* distance formula, classifying data to k clusters, with higher similarity intra-clusters and lower similarity between clusters. The main steps are shown in algorithm 1:

ISSN: 2005-4270 IJDTA

Algorithm1 K-Means main algorithm

Input: number k of clusters, initial central points and data waiting for division;
Output: members of k clusters
(1) Choose randomly k objects as cluster center;
(2) Calculate distance between the other object and clustering center; divide the object to relative clustering center;
(3) Figure out clustering center based on objects of each cluster;
(4) Judge if clustering center changes and the number of iteration below threshold value; if changes, return to step (2);
(5) Judge if the number of iteration below threshold value; if yes, output k groups of members; otherwise, the output of clusters fails;
(6) The program execution ends.

Here to an example to explain the K-Means clustering process. Here in order to explain the convenience, the choice is the two-dimensional data. Data as shown in Table1:

Table 1. K-Means Experimental Data

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11
x	1	2	2	3	8	8	8	10	15	16	16
y	10	11	11	12	4	2	5	5	15	16	12

The space diagram is shown in Figure 1

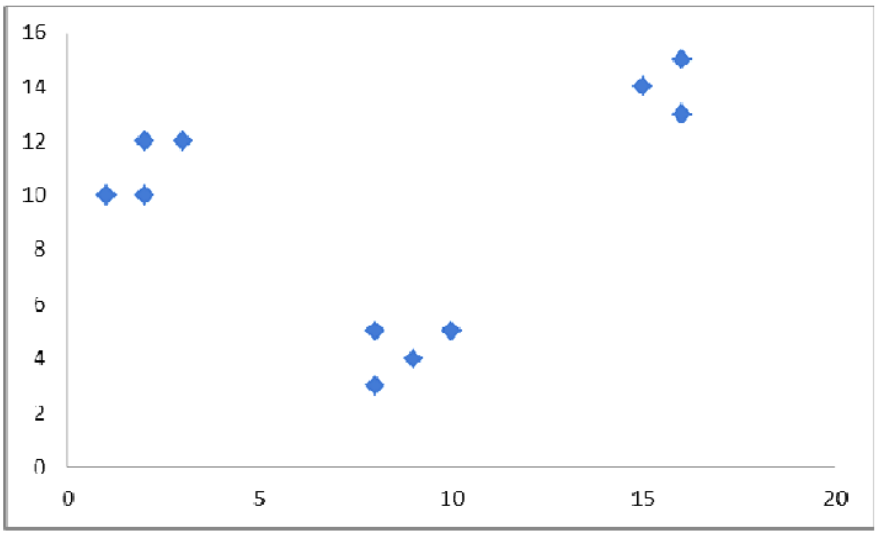

Figure 1. Data Space Graph

Set the initial K=3, and selection of the initial center point for x1, x2, x3. According to Euclidean formula for the first iteration, the clustering results as shown in Figure 2.

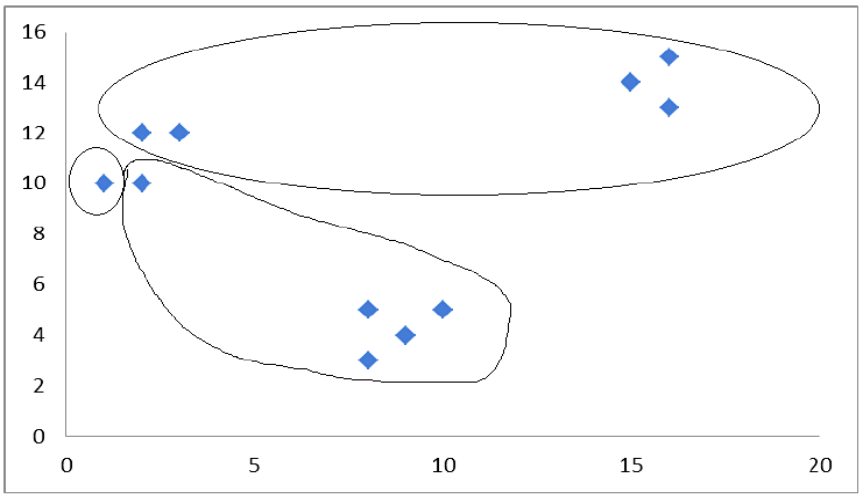

Figure 2. First Clustering Results

After many clustering, the final result is shown in Figure3. I can be seen from the graph, the clustering results accord with the reality condition

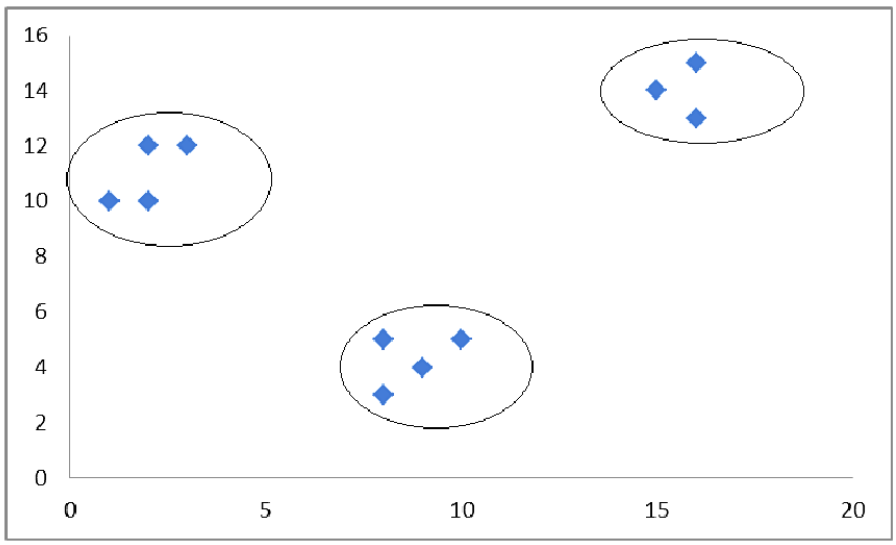

Figure 3. Clustering Result

2.2. Shortcomings of the Algorithm

K-means algorithm has merits of simple and easy implementation. The complexity of it is $O(tkn)$, where t is number of iteration; n is summation of classifying data; k is number of groups. Generally kn, tn, the complexity of the algorithm approximates $O(n)$.

1 K-Means algorithm depends on the K value set

The value of k of clustering result set is often given by user, which decides clustering result. So a reasonable value of k is necessary. The value of k can be determined by experience or observed from data space. Intuitively from data space graph in Figure 1, k=3; however that's only applicable for very few data; for massive dataset, it's not possible to observe visually. About that weakness, the former made lots of improvement. In [7-8], distance cost function was presented as a function to verify the effectiveness of k value. Theoretically it's proved $k_{\max} < \sqrt{n}$.

2 K-means reliant on initial clustering center

In the primitive K-means algorithm, initial clustering center is chosen randomly. The final clustering result depends on the selection of in initial central point, which leads to instability of the result. Lai Yuxia suggested the combination of K-means algorithm and genetic algorithm [9], overcoming the dependence of K-means on initial central points. Lei Xiaofeng developed a K-means CAN algorithm [10]. The algorithm recalled K-means algorithm several times as to get several groups of clustering results; then construct weighted connected graph with those results and merge intersections as per connectivity. Zhang Zhongping utilized breadth-first search to determine initial central point [11].

3 Sensitivity of isolated points

Isolated points can degrade the quality of K-means clustering results because K-means uses mean value as central point, easily susceptible to the extreme value. K-medoids algorithm uses the point in the most central part of cluster as central point rather than mean value as reference point. The fundamental conception is still to distribute points to the most similar central point. PAM is one of methods for implementing k-medoids algorithm. In the beginning, k initial points are randomly determined; use repetitively non-central points to replace initial central point till the best focal point is generated. Ma Shuai stated a clustering algorithm based on reference point and density [12]. The algorithm reflected data's spatial geometric features by reference point and is not sensitive to isolated points. Chen Enhong reduced the effect of isolated points with representative points [13].

4 Scalability

Through analysis above, the complexity of K-means is approximate to $O(n)$. But in the face of huge data volume, the times of computation increases and calculation of similarity becomes time-consuming. Hence in whatever cases, it's essential to do parallel computation. Lv Yiqing [14] put forward parallel K-means algorithm based on message passing interface. Wang Hui [15] proposed a K-means algorithm parallelizing in cluster environment. The two methods demonstrated very good accelerated speed.

4. Improved K-Means Algorithm Based on Sampling and Density

We propose the improved K-means algorithm based on sampling and density (STKMeans) according to the analysis of defects of K-means algorithm. Through sampling and density to determine initial k value and central point, the drawback can be eliminated that k value and initial central point need assigning at the initial stage. Implement the improved K-means in the MapReduce; enhance the scalability of K-means algorithm by its capability to process data concurrently with Hadoop. In the end, the algorithm proves better scalability in the experimental process.

3.1. Concept

Definition 1 (neighborhood of a point): for any point P in the space, the area constituted with radius and P as center of circle is called the domain of point P.

Definition 2 (density): for any point in the space, the number of points in the area of P is called density of P.

3.2. Parallelized Improvement

STKMeans algorithm includes four parts:

(1)Take multiple samples of enormous data

(2)Use density to find out central point of sampling data

(3)Determine global central point

(4)Utilize K-means algorithm to cluster data

Multiple samples are acquired from massive data to generate samples which can reflect the formation of tremendous data. With sampling data, we can calculate the distance

between data points and decide the domain to which data belongs and determine the central point of samples as per the density of neighborhood, thus to decide the global central point of initial data based on sampling central point. After that, the disadvantage can be avoided that initial K-means algorithm relies on initial central point. After the point is determined, data can be clustered with K-means algorithm. From the introduction of K-means algorithm in the above part, in the case of massive data, it's too time-consuming to calculate the distance between object and clustering center. The time grows along with increasing data. Hence we move STKMeans algorithm to Hadoop platform, using it to deal with the most time-consuming computing operation with its parallel calculating ability.

1. Determine Data Central Point Based on Sampling and Density

To determine the central point through sampling and density can be respectively carried out in a serial manner, which is impossible to big samples and multiple samples because it's too time-consuming. Besides, no connection exists to determine central point by samples. The speed of obtaining central point can be enhanced through optimization by Hadoop's capability to process a large quantity of data.

Hadoop platform distributes samples to different execution nodes. Each execution node invokes custom-setting Map function to calculate candidate points which generate samples; then perform Reduce operation of generated candidate points. According to the idea, design SampleMap class, SampleReduce class.

SampleMap class is the actual implementation of Map operation. The default input of Map operation is $< key, value >$, in which Key value is the offset of current row against initial line. Value is coordinate information of node x. In Map operation, we calculate the distance between point x and candidate point; if all distance is bigger than r, use point x as new candidate point; otherwise, add the information of point x to candidate points whose distance between x is smaller than r; then ultimately output candidate point $< key^{'}, value^{'} >$. The main steps are shown in algorithm 2:

Algorithm 2 SampleMap class main algorithm

Input: offset Key of initial row; node x's coordinate information value

Output: identifier $key^{'}$ of candidate central point; $value^{'}$ of candidate central point

(1) Calculate the distance between node x and each candidate point;
(2) If the distance is smaller than radius, accumulate each coordinate of point x to candidate point and their candidate will increase by 1;
(3) If all distance is bigger than radius, regard point x as new candidate central point.
(4) After all candidate central points are generated, construct character strings to represent each coordinate of candidate points, with their hash value as $key^{'}$; character string including the sum of each dimensional coordinate in the area of candidate central points and density are used as $value^{'}$.
(5) Output $< key^{'}, value^{'} >$
(6) Algorithm end

SampleReduce class is the actual implementation of Reduce operation. The default input of Reduce operation is $< key, V >$; at this moment Key's value is identifier of candidate point; the value of V is the collection of intermediate values with the same Key. Based on the density setting value, Reduce function determines if new candidate points are qualified, *i.e.* bigger than predefined density value; then outputs qualified central points. The main steps are shown in algorithm 3:

Algorithm3 SampleReduce class main algorithm

Input: character string hash value key of candidate points; intermediate value V with the same key;

Output: identifier key' of candidate central points; $value'$ of candidate central points.

(1) Judge if density value of candidate central points is bigger than preset density value;

(2) If bigger, calculate new central point in the area; user identifier of new central point as key' and new central point as $value'$; output is $< key', value' >$ □;

(3) If smaller, abandon candidate central points.

(4) Algorithm end

2. Use K-Means to Generate Clustering

After central point is obtained through sampling and density, K-means algorithm divides data to relevant clusters. To divide clusters with K-means algorithm is time-consuming for the reason of calculation of the distance between data and central points and re-calculation of those points. Here we assign the distance calculation operation to each execution node of Hadoop platform, using execution nodes to calculate the distance between data point and central point and classify it to the cluster with the smallest distance. The re-calculation of central points is completed by Reduce operation. Re-calculate the central point of cluster at Reduce execution point. According to the idea, we need design KMeansMap class and KMeansReduce class.

KMeansMap class is the actual implementation of Map operation. At Map stage, calculate the distance between each data point and central point to get the shortest distance and allocate data point to the nearest central point. The main steps are shown in algorithm 4:

Algorithm 4 KMeansMap class main algorithm

Input: offset Key of initial row; node x's coordinate information value

Output: Group number key'; $value'$ of node x's coordinate information

(1) The first implementation of the need to read from the HDFS global center point, stored in the global variable space;

(2) Calculate the distance between the X and the global center point, find the minimum distance, determine the center of the point x;

(3) Index as the center point as group key', node x coordinates information as $value'$;

(4) output $< key', value' >$

(5) Algorithm end

The KMeansReduce class gets the data points for each cluster to calculate the center of each group. The main steps are shown in algorithm 5:

Algorithm 5 KMeansReduce class main algorithm

Input: Group index belongs to the group of nodes

Output: Group index as key', the new center point as $value'$

(1) The sum of nodes is the same as the nodes of the same group of index, and the average value of each dimension is used as the new center point;

(2) Group index as key', the new center point as $value'$

(3) output $< key', value' >$

(4) Algorithm end

5. Experiment Design and Discussion

In the experiment, we compare the running situation between K-Means parallel algorithm and STKMeans algorithm with several groups of data. We analyzed experimental results from the aspect of clustering result, convergence time and accelerated speed. The testing data contains two parts. Testing data are collected from Iris data by Edgar Anderson. The convergence time and speed-up ratio are tested with artificial data: D0 (including 100000 data), D1 (including 300000 records), D2 (including 700000 records), D3 (including 1000000 records), D4 (including 1400000 records).

4.1. Clustering Analysis

The iris data set is a Canadian Anderson research on the Jasper peninsula of iris Geographic variation data [16], which contains 150 samples. In the 150 samples, including three kinds of iris, respectively is mountain iris setosa, iris versicolor and iris virginica. Each sample has four properties, respectively, the length and width of the calyx and petals, so the sample can be represented by a matrix of 150*4.

Select the iris as a test of the original K-Means algorithm and STKMeans algorithm in the data set. These 150 samples have been determined to be divided into three categories, and have a clear clustering center, the central position of the points, respectively (5.538,1274,5.452,1.036), (4.005,2.428,2.454,1.254), (4.836,1.78,3.27,2.328).

4.2. Running Time

Running time is used to judge the execution speed of the algorithm. Judged from the algorithm itself, K-means method consumes time largely on data grouping; while STKMeans' time is spent on these two parts: generating central points and data grouping. But in the case of massive raw data, STKMeans algorithm spends much time on data packet. To prove that the algorithm is more time-consuming, Hadoop node communication consuming time is ignored and also the error of time taken by different nodes in implementing identical data, measuring the execution speed of the algorithm by according to its iterations.

Analyzed from the algorithm, the execution time of concurrent K-means algorithm and STKMeans algorithm depends on initial central point, under the same circumstances. In the experiment, the number of Hadoop platform node is set 4 as to test respectively dataset D0, D1, D2, D3, D4 in Figure4. The iteration of both STMeans algorithm and parallel K-means algorithm increases together with growing data volume. However, STKMeans algorithm iterates less often than K-means algorithm because the former determines initial points based on sampling and density, more targeted than random choice of points, so it converges more rapidly.

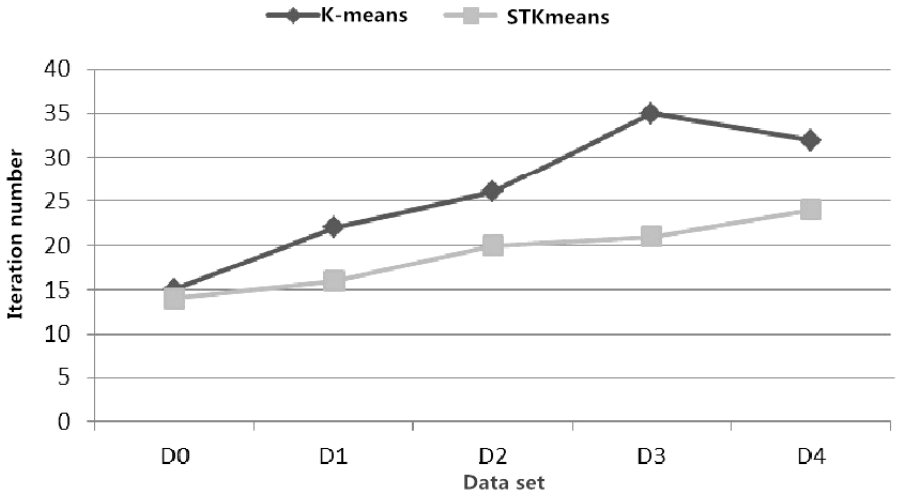

Figure 4. Algorithm Running Times

4.3. Acceleration Rate

Acceleration rate means the ratio of task execution time between the single processor and multiprocessor. It's used to evaluate the parallelized performance and effect of the program. The formula is defined as follows

$$S_n = \frac{T_1}{T_n} \tag{1}$$

In the experiment, we tested execution time of dataset D0, D1, D2, D3 and D4 on different numbers of nodes. Figure 5 shows the acceleration rate of K-means algorithm. Figure 6 shows that of STKMeans algorithm. From Figure 6, we can see that STKMeans algorithm and K-means method both have good acceleration rate on the Hadoop platform, which becomes bigger with increasing data size. But with increasing number of nodes, the algorithm's acceleration rate tends to grow steadily, because with more nodes, inter-node communication consumption increases, causing that the increment of acceleration rate becomes gentler.

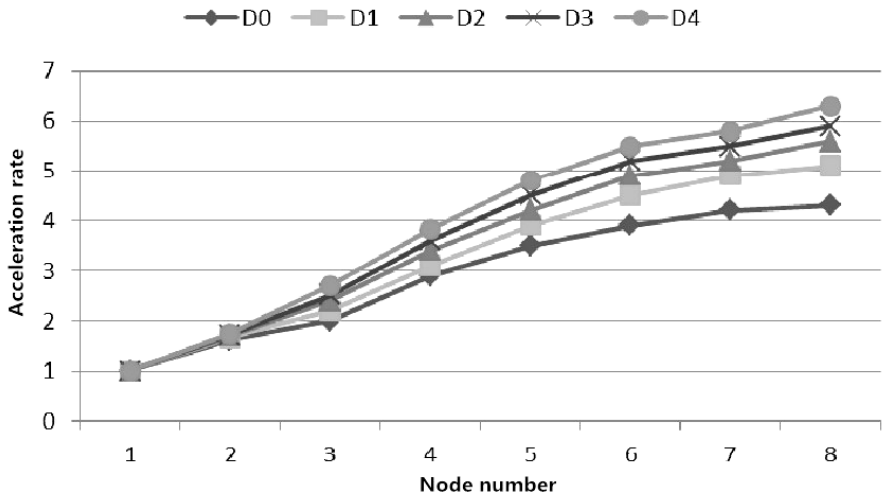

Figure 5. Acceleration Rate of Parallel K-Means Algorithm

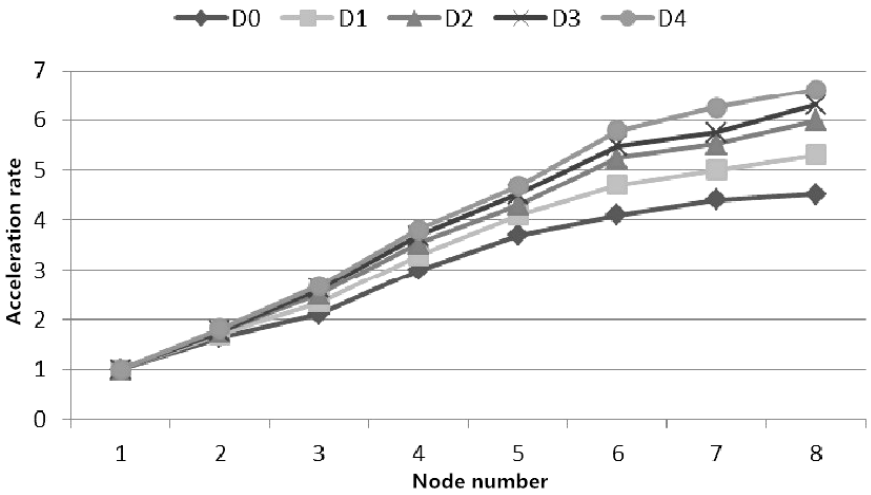

Figure 6. Acceleration Rate of STKMeans Algorithm

6. Conclusion

Starting with the K-Means algorithm, the clustering results are dependent on the K value and the initial center point of the defect. An improved clustering algorithm STKMeans based on sampling, density and Hadoop is proposed. STKMeans algorithm retains the advantages of the original K-Means algorithm, through the density to select the initial center point, the algorithm is not dependent on the K value and the initial center point. Finally, the STKMeans algorithm has better convergence and speedup by using different data sets to test the algorithm.

References

[1] L. Shenghui, H. Tao and T. Yanna, "Research based on K-means clustering algorithm", Computer technology and development, vol. 7, **(2011)**, pp. 54-57.
[2] W. Qian, W. Cheng, F. Zhenyuan and Y. Jinfeng, "Overview of research on K-means clustering algorithm", Electronic design engineering, vol. 7, **(2012)**, pp. 21-24.
[3] W. Wenpeng, C. Guangping and H. Jun, "An improved K-means algorithm for initial clustering center selection", Small microcomputer system, vol. 6, **(2012)**, pp. 1320-1323.
[4] X. Xiuhua and L. Taoshen, "An improved K-means based PSO clustering algorithm", Computer technology and development, vol. 2, **(2014)**, pp. 34-38.
[5] C. Zhengqi, C. Yongchun and S. Yabin, "An improved artificial bee colony clustering algorithm based on K-means", Computer application, vol. 1, **(2014)**, pp. 204-207
[6] J. M. Queen, "Some methods for classification and analysis of multivariate observation", Proceeding 5th Berkeley Symp. Math. Statist. Prob, vol. 1, no. 28, **(1967)**, pp. 1-297.
[7] Y. Shanlin. L. Yongsen, H. Xiaoxuan and P. Ruoyu, "K-means algorithm of K value optimization problem", Systems engineering theory & practice, vol. 10, no. 2, **(2006)**, pp. 97-102.
[8] T. Senping and W. Wenliang, "Automatic acquisition of K-Means clustering parameters K algorithm", Computer engineering and design, vol. 32, no. 1, **(2011)**, pp. 274-277.
[9] L. Yuxia, L. Jianping and Y. Guoxing, "Analysis of K means cluster analysis based on genetic algorithm", Process, vol. 34, no. 20, **(2008)**, pp. 200-203.
[10] X. F. Lei, K. Q. Xie, F. Lin and Z. Y. Xia, "An eminent clustering algorithm based on local optimality of K-Means", Journal of Software, vol. 19, no. 7, **(2008)**, pp. 1683-1692.
[11] Z. Z. Ping, W. A. Jie and C. L. Ping, "Method for initializing K-Means clustering algorithm based on breadth first search", Computer Engineering and Applications, vol. 44, no. 27, **(2008)**, pp. 159-161.
[12] S. Ma, T. J. Wang, S. W. Tang, D. Q. Yang and J. Gao, "A fast clustering algorithm based on reference and density", Journal of Software, vol. 14, no. 6, **(2003)**, pp. 1089-1095.
[13] C. Enhong, S. Wang, Y. Ning and W. Xufa, "A design and Realization of efficient clustering algorithm with representative points", Pattern recognition and artificial intelligence, vol. 14, no. 4, **(2011)**, pp. 416-422.
[14] L. Yiqing and L. Jinxian, "Based on MPI parallel PSO combined with K-means clustering algorithm", the application of computer, construction control, vol. 31, no. 2, **(2011)**.

[15] W. Hui, Z. Wang and M. Fan, "Parallel of K-Means clustering algorithm based on cluster environment", Journal of Henan University of Science and Technology: Natural Science Edition, vol. 29, no. 4, **(2008)**, pp. 42-46.
[16] E. Anderson, "The irises of the Gaspé Peninsula", Bulletin of the American Iris Society, vol. 59, **(1935)**, pp. 2-5.

Author

Feng Hu, He received his B.S degree from Hainan Normal University and received his M.S degree from Chongqing University. He is a lecturer at Qiongtai Teachers College. His research interests include computer application.

