Density Based Initial Center Optimization Algorithm

Shengli Sun
School of Software and Microelectronics
Peking University
Beijing, China
e-mail: slsun@ss.pku.edu.cn

Zhigao Zheng
School of Software and Microelectronics
Peking University
Beijing, China
e-mail: zhengzhigao@pku.edu.cn

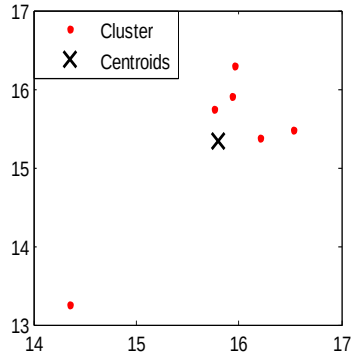
Yu Zhang
School of Software and Microelectronics
Peking University
Beijing, China

***Abstract*—K-Means is the most popular clustering algorithm with the convergence to one of numerous local minima, which results in much sensitivity to initial representatives and noise point because of its distance based judgment. Density-based clustering algorithm is not sensitive to outliers and noise data, but it's difficult to present the high dimensional data as well as the changes of the data density. Grid-based method is fast, but may reduce the quality and accuracy of the cluster. However, this paper proposes a novel density based initial center optimization algorithm (DBICO) to choose the initial center, which by means of the local optimality and sensitivity of density-based clustering algorithm and grid-based method. The core idea is to divide the dataset into several cubes and merge some cubes, delete the noise points according to the density, calculate the initial center point and then clustering the dataset. Doing this can reduce the number of iterations, and avoid the disadvantages of the K-Means algorithm results differ due to the different initial points. Theoretic analysis and experimental demonstrations show that the algorithms this paper proposed outperforms existing algorithms in clustering quality, and it was proved fruitful applications in the logistics.**

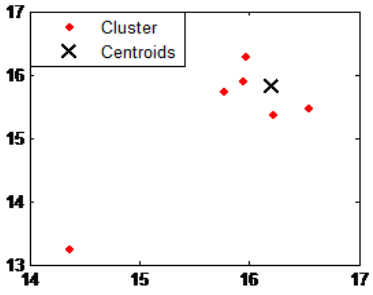
Keywords- partitio; merge cube; K-Means; Initial center points; density

I. INTRODUCTION

We often need to analyze the vehicle trajectory information for warehouse location decision and optimize logistics system. Doing this we need to use data mining knowledge to analyze vehicle trajectory information to find the real vehicle trajectory, based on this information we can compute a reasonable set of logistics warehouse which the cost and time is most provinces of the goods from the warehouse to the point-of-sale. We all know that vehicle trajectory is continuous so that the algorithms which based on the density scanned can't be used. This kind algorithms may classify all the locus point into a cluster and can't find the actual cluster center (the potential location of the warehouse). Since the K-means has the advantages of simplicity and fast convergence [1]. Especially for the numerical data, it can easily reflect the clustering geometric and statistical significance, at the same time it is can easily find the datasets with round shape. These features make the K-Means algorithm has important applications in the logistics information mining, but at the same time the original K-Means algorithm also has some drawbacks: 1. Requires user to input the value of k, but in practice is difficult to determine due to lack of experience; 2. The position of cluster centers is sensitive to the noise point, shown in Fig.1; 3. sensitive to the initial cluster centers, different initial cluster centers may have different results, shown in Fig. 2, and the result of different initial point is shown in Table 1. Inappropriate k or initial centers may cause resources waste and may cause some more serious negative impact to the entire system in the peak hours.



(a) Results of K-Means algorithms



(b) Real results of clustering

Figure 1. The effect of K-Means brought by noise data

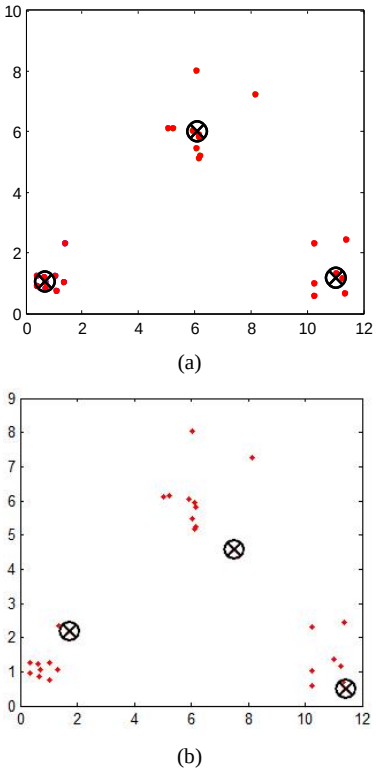


Figure 2. The results of different initial point for K-means

TABLE I. THE SUM DISTANCE OF K-MEANS CENTERS AND ALL POINTS WHICH IN THE CLUSTER

Run times	center1	center2	center3
1	98.49	74.38	22.69
2	22.69	22.69	76.92
3	22.69	22.69	22.69

To solve the problem of traditional K-means algorithm this paper proposed a divide density based initial point optimization selection method according to the density scanning method. The method this paper proposed integrated many advantages of clustering algorithms, especially the local optimum characteristics of K-means and anti-noise characteristics of density based methods. The possible location of the warehouse can be easily found based on these characteristics. Theoretical analysis and experimental results show that, density based optimization initial center method has the following characteristics:

1. Draw lessons from density-based clustering method which is not sensitive to outliers and noise data;

2. Combined partitioning technique to quickly determine the value of k, which make the algorithm does not need too much prior knowledge, at the same time the concept of fuzzy ratio was introduced, do this can reducing the sensitivity of parameter threshold;

3. Overcomes low quality and accuracy shortcomings of the mesh-based method.

According to the actual test, the algorithm can accurately and quickly find the location of the warehouse in logistics system, and do this can reduce shipping and time cost bring by inappropriate warehouse. Section 2 describes the related work; section 3 introduced the details of density-based method and the principle; density based initial center optimization algorithm (DBICO) and the formal definition are all in section 4; section 5 is the simulated experiment, and the conclusion is at last.

II. RELATED WORK

In order to overcome the deficiencies of the original K-Means algorithm, different scholars put forward a series variant algorithms from different perspectives. Huang proposed a K-Means-based automated variable weighting algorithm, which improved the variable selection problem [2]. Dhillon improved the performance by adjust iteration process to recalculate the centers [3].Zhang adjust the iterative optimization process according to the weights [4]. A new similarity measure function was proposed by Yang Fengzhao to overcome the shortcomings of the traditional distance function for high-dimensional space [5]. Sarafis applied the genetic algorithm to K-Means objective function based on this work he proposed a new clustering algorithm [6].An immunization based K-means clustering algorithm was proposed in reference [7] to overcome the local minima and sensitive initialization value problem of traditional algorithm, and the new algorithm can convergence faster.

A high efficiency and accuracy clustering algorithm was proposed in reference [8], which based on the sub-cluster weighted connected graph. The method merged the sub-cluster according to the connectivity. Reference [9] use the density-based method and a more effective pruning method in micro-clustering stage which also can improve the accuracy and efficiency. Kaufman [10] proposed a heuristic method, estimated the local density of the data points as the initial value of the sample. A initializing the spectrum of K-Means method was proposed in reference [11] and a density-sensitive similarity measure method was proposed in reference [12] based on this method the author introduced the spectral clustering into density-sensitive spectral clustering algorithm. Unlike traditional clustering algorithm, class clusters are defined as dense connectivity sub-regional. This method can discover clusters of arbitrary shape according to the density (number of instances per unit area), at the same time the method can also have the ability of immunity outliers and noise data; a grid structure was used in the grid-based clustering algorithm, the algorithm clustered the data based on the data block which divided by the value space [13]. The grid-based clustering algorithm is often combined with other methods, particularly the density-based clustering approach. As the vehicle trajectory is a continuous dataset so we cannot use the density-based and gird-based method in logistics warehouse location system, otherwise, the whole track is gathered into a cluster and lose the meaning of the data mining.

DBSCAN is a typical low-dimensional space-oriented data density-based clustering algorithm, the key point is that the density of local neighborhood decide the connectivity. The DBSCAN algorithm provide a searching method which based on the density threshold parameter, so the result was identified and algorithm itself is not responsible for the results. All these

characters made the DBSCAN algorithm is sensitive to the parameter values, cannot characterize the intrinsic clustering structure of the data set, and the vast majority of sample points gathered in a very small number of clusters, and some other such shortcomings.

The OPTICS (ordering points to identify the clustering structure) [14] algorithm improved some shortcomings of DBSCAN.

III. DENSITY-BASED PARTITION METHOD

In the gird-based method we divided the data space into finite cells, so all the processing objects are a single unite. Doing this can gain quickly processing speed, at the same time as the method is far away from the target database, so it may reduce the quality and accuracy of the clusters [13]. The proposed method considered the density of the target data in the database, overcome the shortcomings of mesh-based approach's low quality and accuracy and learnt the advantages of the density-based method. The existing density estimation method was proposed based on the assumption that the data obey Gaussian mixture distribution, based on these we can consider that the dense regions of inputted data contains natural clustering, so we can locate the initial center according to the density of the data. The limitation of this method is only good for the convex structure data but not any complex shape data structure.

This paper proposed a density-based partition method to overcome the shortcomings of the Gaussian function. At first we divided the sample database into a number of a cube and determine the center of each cube, then merge the cubes according to the density similarity and re-calculate the new center of the merged cube. Based on these we use the calculated center as the initial points of the K-Means algorithm. This will not only overcome the uncertain results drawbacks which brought by the different starting point of K-Means algorithm, at the same time overcame the drawbacks of the DBSCAN algorithm.

A. Basic Definitions

Definition 1 (Cube) A sample database D=$\{x_1,x_2,\ldots,x_n\}$ was made up by m cubes, a cube was a subset of the sample database. $\forall i, j, cube_i \cap cube_j = \varnothing$, and $cube_1 \cup cube_2 \cup ... \cup cube_m = D$。

Definition 2 (Partition) The processing of divide the sample database into *m* cubes.

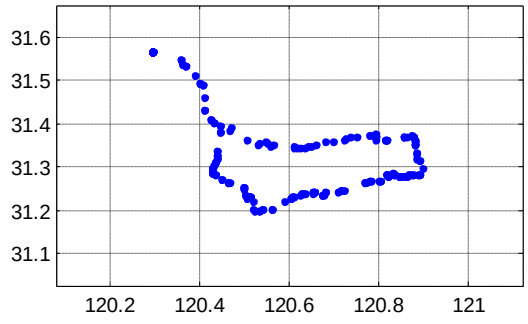


Figure 3. A partition for a sample database D

Definition 3 (Density of Cube) The density of the cube was described as the comparison of the number of sample points and the area (volume) of the cube. Noted as $den(cube_i)$

$$den(cube_i) = \frac{count(cube_i)}{V(cube_i)}$$

Where $count(cube_i)$ was the points of $cube_i$, $V(cube_i)$ was the area (in two-dimensional space) or volume (in multi-dimensional space) of $cube_i$.

Definition 4 (Center of the cube) The point which have the minimum sum distance from the others in the *cube*, noted as *cubecenter*. If the points of the *cube* is $s, \forall x_i, x_j \in cube$, then:

$$subcenter = \min \sum_{j=1}^{s} \sqrt{(x_i - x_j)^T (x_i - x_j)}$$

Definition 5 (Fuzzy ratio) Note the new cube as *cube** which made by the boundary of the center points of two adjacent cubes $cube_i, cube_j$, the fuzzy ratio was the comparison of the density of *cube** and the quadratic of product of $cube_i$ and $cube_j$, noted as $fuzzy(cube_i, cube_j)$.

$$fuzzy(cube_i, cube_j) = \frac{den(cube*)}{\sqrt{den(cube_i) * den(cube_j)}}$$

B. Density-based Partition Method

The basic idea of partition method is chosen 2^d (*d* is the dimension of the sample database) boundary points to decide the boundary of the sample database. Since the clusters is no more than $\sqrt{n}$ (*n* is the number of points in sample database *D*), so we can divide the sample database into $\sqrt{n}$ *cubes* according to this property, according to definition 5we know that the bigger fuzzy ratio the higher similarity the two *cubes*.

Prove: Define the similar ratio of two cubes according the definition

$$sim(cube_i, cube_j) = \frac{den(cube_i)}{den(cube_j)}$$

The similar ratio closer to 1 the more similar of two cubes. So we need to prove $sim(cube_1, cube_2) > sim(cube_2, cube_4)$ according to $fuzzy(cube_1, cube_2) > fuzzy(cube_2, cube_4)$. Letbe $sim(cube_1, cube_2) = a, sim(cube_2, cube_4) = b$, so we need to prove *a*>*b*。

As $fuzzy(cube_1, cube_2) > fuzzy(cube_3, cube_4)$ so

$$\frac{den(cube_1) + den(cube_2)}{\sqrt{den(cube_1) * den(cube_2)}} > \frac{den(cube_3) + den(cube_4)}{\sqrt{den(cube_3) * den(cube_4)}}$$

That is:

$$\frac{count(cube_1) + count(cube_2)}{\sqrt{count(cube_1) * count(cube_2)}} > \frac{count(cube_3) + count(cube_4)}{\sqrt{count(cube_3) * count(cube_4)}}$$

So $b(a+1) > \sqrt{ab}(b+1)$

So $a > b$.

Reference [14] researched that how should *k* set in the K-Nearest Neighbor classification algorithm, the author give us a

rule: $k \approx n^{3/8}$. Refer to this conclusion we merged the cubes which have higher fuzzy ratio, until $m \approx n^{3/8}$ 。

IV. Density Based Initial Center Optimization Algorithm

A. DBICO Algorithm

According to the Density Based Partition Method we divided the sample database into *m* cubes, where $m \approx \sqrt{n}$. Then we descend all the cubes according to the density, according to the results of reference [14], we consider the last $\sqrt{n} - n^{3/8}$ *cubes* maybe the noise area. So we to merge the *m* cubes according to the similarity until the left cubes approaching to $n^{3/8}$. At the same time we need to mark the nonadjacent cubes as the noise are and put it into the noise queue and the others into non-noise queue.

The basic idea of the algorithm is chosen the cube's center as the initial center which take maximum density, then remove the cube from the non-noise queue, and do the same thing to get the second center, loop to perform these steps until the non-noise queue was empty. So we can get *k* initial centers. According to these centers divide the data into different cubes to make the objective function

$$E = \sum_{i=1}^{k} \sum_{j=1}^{n_j} d(x_j, c_i)$$

Minimize, to make the generated clusters as compact and independent as possible.

The following is the description of the algorithm.

DBICO Algorithm
Input The database which has n points
Output *k* initial centers
Step1 Divided the database into *m* cubes according to definition 2;
Step2 Merge the cubes according to definition 5;
Step3 Order the cubes according to the density of the cube and mark the last $\sqrt{n} - n^{3/8}$ *cubes* as noise area, then decide the *k* initial centers;
Step4 Delete the noise area, and output the *k* initial centers.
Step5 End.

Figure 4. DBICO Algorithm

The DBICO algorithm use the partition method to calculate the initial center make the algorithm without much prior knowledge. Algorithm learnt the density-based method to exclusion some noise area to make the algorithm not sensitive to outliers and noise data.

B. Analysis the Complexity of the Algorithm

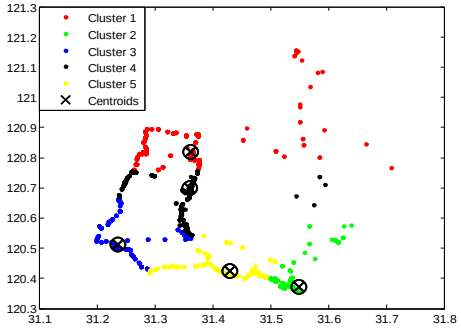
The efficiency of K-means is quite high K-Means algorithm efficiency is quite high, its computational complexity is *O* (*nkdt*), where *n* is the number of objects of all samples, *k* is the numbers of clusters, *d* is the dimension of the sample data, *t* is the iterative times. Usually *k*<<*n* and *t*<<*n*. The time complexity of database partition, cube merge, noise area excluding are constant in the proposed algorithm, and all subsequent calculations are based on pre-divided, cube merger, eliminate noise district, which does not involve any high costs of space. So the time complexity of the DBICO algorithm is $O(\sqrt{n})$, and the division process can be dealt with under online. At the same time the time complexity of K-Means is *O* (*nkdt*)+ $O(\sqrt{n})$, *n* is the number of objects of all samples, *k* is the numbers of clusters, *d* is the dimension of the sample data, *t* is the iterative times. Since the algorithm divide the database firstly and then merged the cubes so the value of *k* is relatively stable, compared with ordinary K-Means algorithm, the *t* value of DBICO is smaller.

But in terms of time complexity the DBICO and K-Means are in the same order of magnitude, the advantage is not obvious. But the pre-treatment of DBICO can effectively overcome the drawbacks of K-Means, difficult to decide the value of k and the center point offset which due to the lack of prior knowledge .

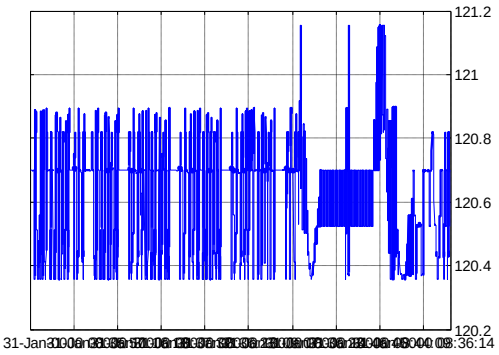
V. Simulation Experiment

In this section we use experiment to study the performance and effectiveness of the proposed DBICO algorithm. The data used in the experiment include the real GPS track data of a logistic company and a simulation dataset for clustering algorithm, the GPS data of the logistic company has 3613 sample points, and the simulation data has 16497 sample points. The hardware environment of experiment is Intel (R) Core (TM) i5-25200 four-core 64-bit 2.5GHz CPU and 4GB memory, the software environment is Windows 7-64bit (Professional) operating system, all code are written by Java (64bit JDK) and Matlab2012.

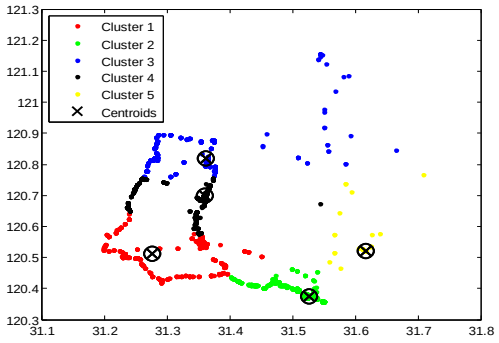
Fig. 5 to Fig. 6 shows the results two sets of data after database dividing, cube merger, and eliminate noise area and the results of the optimized K-Means algorithm. In each figure, (a) is the results of DBICO algorithm, (b) is the corresponding density difference sequence curve, (c) is the results of the K-Means clustering algorithm.



(a) The result of DBICO Algorithm

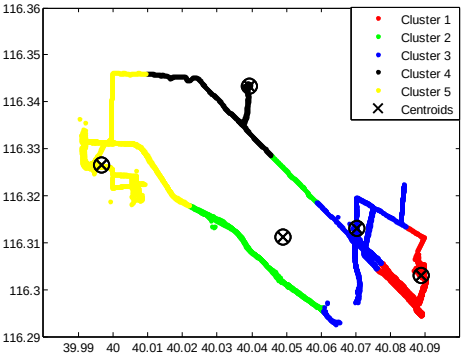


(b) Density Difference Sequence Curve

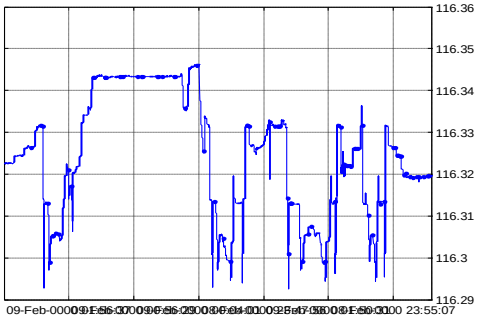


(c) The result of K-Means

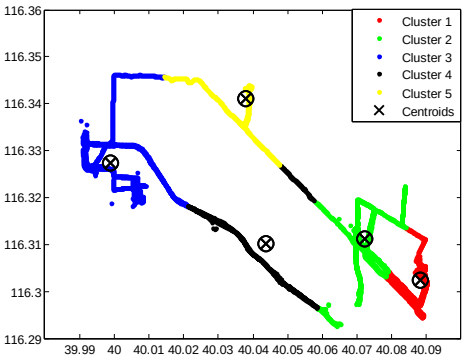
Figure 5. The Result of First Dataset



(a) The result of DBICO Algorithm



(b) Density Difference Sequence Curve



(c) The result of K-Means

Figure 6. The Result of Second Dataset

From the two sets of results we can conclude that the DBICO algorithm which this paper proposed can obtained high quality of clustering results without post-processing step. We first use the K-Means algorithm to do 100 times clustering with k equals 5, and then selected 5 set results, and calculated the average. Then use the DBICO algorithm to do the same thing compare the results with K-Means, the following table is a comparison of the results of K-means and DBICO algorith

TABLE II. THE CENTER POINTS OF K-MEANS AND DBICO FOR DATASET 1

Algorithm	Center1	Center2	Center3	Center4	Center5
K-Means	(31.35,120.84)	(31.23,120.51)	(31.54,120.41)	(31.35,120.70)	(31.37,120.50)
K-Means	(31.37,120.50)	(31.35,120.70)	(31.61,120.54)	(31.35,120.84)	(31.51,120.38)
K-Means	(31.23,120.52)	(31.37,120.49)	(31.35,120.84)	(31.35,120.70)	(31.54,120.41)
K-Means	(31.35,120.70)	(31.54,120.92)	(31.30,120.51)	(31.32,120.82)	(31.53,120.41)
K-Means	(31.35,120.74)	(31.61,120.54)	(31.53,120.37)	(31.42,120.44)	(31.28,120.53)
average	(31.33,120.66)	(31.42,120.63)	(31.46,120.53)	(31.36,120.70)	(31.44, 120.44)
DBICO	(31.35,120.61)	(31.42,120.61)	(31.44,120.53)	(31.38,120.70)	(31.44, 120.45)

TABLE III. THE CENTER POINTS OF K-MEANS AND DBICO FOR DATASET 2

Algorithm	Center1	Center2	Center3	Center4	Center5
K-Means	(40.09,116.30)	(40.07,116.31)	(40.00,116.33)	(40.04,116.31)	(40.04,116.34)
K-Means	(40.07,116.31)	(40.09,116.30)	(40.04,116.34)	(40.00,116.33)	(40.01,116.32)
K-Means	(39.99,116.33)	(40.04,116.34)	(40.08,116.31)	(40.05,116.30)	(40.01,116.32)
K-Means	(40.04,116.34)	(40.00,116.33)	(40.05,116.31)	(40.01,116.32)	(40.08,116.31)
K-Means	(40.04,116.33)	(40.08,116.31)	(39.99,116.33)	(40.01,116.35)	(40.01,116.32)
average	(40.04,116.32)	(40.05,116.32)	(40.03,116.32)	(40.03,116.32)	(40.03,116.32)
DBICO	(40.03,116.31)	(40.04,116.32)	(40.03,116.32)	(40.02,116.33)	(40.02,116.32)

According to Table 2 and 3 we can conclude that the centers calculated by DBICO algorithm was relatively stable than K-Means.

VI. CONCLUSION

This paper proposed a DBICO algorithm to select the initial center points of the K-Means. Theoretic analysis and experimental demonstrations show that the algorithms this paper proposed outperforms existing algorithms in clustering quality, can effectively overcoming the shortcomings of existing algorithms and it was proved fruitful applications in the logistics. Meanwhile, the concept of the fuzzy ratio and cube combined was introduced in this paper, which can effectively prevent the shortcomings which brought by not fixed center points of K-Means. The new algorithm also reduced iterations through cube merge. In addition, the offline pretreatment can usefully remove the noise data.

ACKNOWLEDGMENT

This work is supported by: The Natural Science Foundation of Jiangsu Province under Grant No. BK2010139

Project funds: The Natural Science Foundation of Jiangsu Province under Grant No. BK2010139.SUN Shengli,was born in 1979. He was an associate professor of School of Software and Microelectronics of Peking University. He was a CCF member, membership ID E20-00 11046M. His research interests include Services Computing, data management and data mining etc. Corresponding author ZHENG Zhigao, was born in 1988. He is a member of China Computer Federation (CCF, membership ID E200026001G), Association for Computing Machinery (ACM, membership ID 7105878) and International Association of Computer Science and Information Technology (IACSIT, membership ID 80344484). His research interests include data mining, services computing, cloud computing & big data, and cyber physical system, etc.

REFERENCES

[1] Huang Z. Extensions to the K-means algorithm for clustering large data sets with categorical values. Data Mining and Knowledge Discovery, 1998, 2(3): 283 ~ 304

[2] Huang J Z, Ng M K, Rong Hongqiang, et al. Automated Variable Weighting in K-Means Type Clustering. IEEE Tans on Pattern Analysis and Machine Intelligence, 2005, 27(5): 657-668

[3] Celebi M., Kingravi Hassan, Vela Patricio A. A comparative study of efficient initialization methods for the k-means clustering algorithm. Expert Systems with Applications. 2013, 40(1): 200-210

[4] Zhang B. Generalized K-Harmonic Means: Dynamic Weighting of Data in Unsupervised Learning. Proceeding of the 1st SLAM International Conference on Data Mining. Chicago, USA, 2001: 1-13

[5] Yang Fengzhao, Zhu Yangyong. An Efficient Method for Similarity Search on Quantitative Transaction Data. JOURNAL OF COMPUTER RESEARCH AND DEVELOPMENT, 2004,41(2): 361-368

[6] Sarafis I, Zalzala A M S, Trinder P W. A Genetic Rule-Based Data Clustering Toolkit. Proceeding of the Congress on Evolutionary Computation. Honolulu, USA, 2002:1238-1243

[7] Xing Xiaoshuai, Pan Jin, Jiao Licheng. A Novel K-means Clustering Based on the Immune Programming Algorithm. CHINESE JOURNAL OF COMPUTERS. 2003, 26(5): 1-6

[8] LEI Xiao-Feng, XIE Kun-Qing, LIN Fan and XIA Zheng-Yi. An Efficient Clustering Algorithm Based on Local Optimality of K-Means. JOURNAL OF SOFTWARE. 2008, 19(7): 1683-1692

[9] Ma J, Perkins S. Time-Series Novelty Detection Using One-Class Support Vector Machines. Proceeding of International Joint Conference on Neural Networks, USA, 2003,Ⅲ:1741-1745

[10] Kaufman L, Roussecuw P J. Finding Groups in Data: An Introduction to Cluster Analysis. New York, USA: John Wiley & Sons, 1990

[11] QIAN Xian, HUANG Xuan-Jing and WU Li-De. A Spectral Method of K-means Initialization. ACTA AUTOMATICA SINICA, 2007,33(4): 342-346

[12] WANG Ling, BO Lie-feng, JIAO Li-cheng. Density-Sensitive Spectral Clustering. ACTA AUTOMATICA SINICA, 2007, 35(8): 1577-1581

[13] SUN Ji-Gui, LIU Jie and ZHAO Lian-Yu. Clustering Algorithms Research. JOURNAL OF SOFTWARE. 2008, 19(1):48−61

[14] Ankerst M, Breuning M, Kriegel HP, and Sander J. OPTICS: Ordering points to identify the clustering structure. Proceeding of the ACM SIGMOD Int'1 Conf. on Management of Data. Philadelphia; ACM Press, 1999. 49-60.