SCIENCE CHINA
Information Sciences

• RESEARCH PAPERS •

July 2010 Vol. 53 No. 7: 1345–1357
doi: 10.1007/s11432-010-3112-z

An unsupervised grid-based approach for clustering analysis

YUE ShiHong[1]*, WANG JeenShing[2], TAO Gao[1] & WANG HuaXiang[1]

[1]*School of Electrical Engineering and Automation, Tianjin Key Laboratory of Process Measurement and Control, Tianjin University, Tianjin* 300072*, China;*
[2]*Department of Electrical Engineering, National Cheng Kung University, Tainan* 701*, China*

Received February 17, 2009; accepted December 5, 2009

Abstract In recent years, the growing volume of data in numerous clustering tasks has greatly boosted the existing clustering algorithms in dealing with very large datasets. The K-means has been one of the most popular clustering algorithms because of its simplicity and easiness in application, but its efficiency and effectiveness for large datasets are often unacceptable. In contrast to the K-means algorithm, most existing grid-clustering algorithms have linear time and space complexities and thus can perform well for large datasets. In this paper, we propose a grid-based partitional algorithm to overcome the drawbacks of the K-means clustering algorithm. This new algorithm is based on two major concepts: 1) maximizing the average density of a group of grids instead of minimizing the minimal square error which is applied in the K-means algorithm, and 2) using grid-clustering algorithms to thoroughly reformulate the object-driven assigning in the K-means algorithm into a new grid-driven assigning. Consequently, our proposed algorithm obtains an average speed-up about 10–100 times faster and produces better partitions than those by the K-means algorithm. Also, compared with the K-means algorithm, our proposed algorithm has ability to partition any dataset when the number of clusters is unknown. The effectiveness of our proposed algorithm has been demonstrated through successfully clustering datasets with different features in comparison with the other three typical clustering algorithms besides the K-means algorithm.

Keywords clustering, grid-based algorithm, K-means algorithm

Citation Yue S H, Wang J S, Tao G, et al. An unsupervised grid-based approach for clustering analysis. Sci China Inf Sci, 2010, 53: 1345–1357, doi: 10.1007/s11432-010-3112-z

1 Introduction

Clustering analysis has been recognized as a primary data mining tool for knowledge discovery in numerous application fields such as pattern recognition, image processing, and market research. However, because of fast technological progress, the amount of data stored in the database increases rapidly. Although various algorithms have been proposed to address this problem, only few of them show preferable efficiency for large datasets. And the problem of memory availability arises when using a clustering algorithm to deal with very large datasets. Inefficient usage of the limited CPU processing time and memory space can degrade the behavior of a clustering algorithm considerably. The K-means algorithm proposed by MacQueen is a well accepted clustering method for various uses. However, the K-means algorithm is

*Corresponding author (email: shyue1999@tju.edu.cn)

difficult to be applied to very large datasets due to inefficient CPU processing time large memory needs, low efficiency for noisy data, unknown number of clusters, initialization dependence, etc. [1–5].

In order to overcome the above problems, many clustering algorithms have been proposed in the literature. For example, the feature weighting K-means algorithm (FWKA) proposed by Huang [6] integrates many merits of conventional partitional clustering methods and has been successfully applied to text classification [7, 8]. Recently, the general K-means algorithm proposed by Yu [9] presents a unified framework for most variants of the K-means algorithm. Commonly speaking, these algorithms do not aim at clustering very large datasets.

Different from the K-means algorithm, some clustering algorithms start with partitioning the data space into a set of hypercubes (we call them grids, hereinafter, for convenience). Therefore, we term these algorithms grid-based algorithms. Most of the grid-based clustering algorithms have linear time and space complexities and have ability to cluster very large datasets. The main problem of the grid-based algorithms is how to choose the optimal grid parameters as an independent data organization. In fact, in order to efficiently find all concerned clusters, the grid-based algorithms have to solve the following three problems: 1) how to select grids or generate a group of grids for data assignments, 2) how to combine these collected grids to find concerned clusters, and 3) how to extract a robust data partitioning from these combined grids. Any issue discussed above may significantly degrade the performances of a grid-clustering algorithm. So far, these problems remain unsolved.

A remarkable representative of the grid-based algorithms is the DENCLUE (DENsity-based CLUstEring) developed by Hinneburg et a1. [10]. The DENCLUE, which takes into consideration both grid partition and density, has a firm mathematical foundation, and is able to generalize other clustering methods such as CURE [4], Shift Grid [11], CLIQUE [12], K-means, etc. Recently, we have proposed a general grid-clustering approach (GGCA) [13]. The resulting grids partitioning the data space are clustered via a topological neighbor search, with each grid characterized by not only the populated object number but also the neighborhood information [14–16]. The grid-clustering method has proven as a valuable tool for analyzing the structural information of very large datasets. Some ideas of the proposed grid-clustering method were adopted in this study. In [17] we presented a new fuzzy cover-based clustering (FCC) algorithm. In the FCC algorithm, the concept of the cover is employed to build up the backbones of the final clusters. The involved two ideas, namely cover and average density maximization, in FCC are useful to our results in this paper.

2 Grid-based K-means algorithm for clustering

2.1 Key ideas

In this paper we propose a grid-based K-means (G-K-MEANS) algorithm for clustering. The key ideas of the G-K-MEANS algorithm are to take advantage of the high efficiency in the class of grid-based algorithmsOur considerations include:

1) Maximizing the average density of a group of grids to represent the objective functions in place of minimizing the minimal square error in the K-means algorithm. Such reformulation bridges the K-means algorithm to the grid-clustering algorithm, and thus leads to a better match to dataset.

2) Replacing the computation of pairs (center, objects) in the K-means algorithm by grid-based partitions. This makes a significant reduction in time requirement. In fact, clustering a group of grids that contain all data objects rather than all data objects in the data space is the main reason why the class of grid-based algorithms can deal with very large datasets with linear computational complexity.

3) Applying grid movement and grid updating to estimate all cluster centers. Each cluster will be assigned a grid at least to guarantee that any cluster can be found. All grids that are assigned to the same cluster move to the same cluster center and overlap. Consequently, only the most high-density grid in the same cluster is kept and the other grids are removed. This guarantees that each cluster is finally represented only by a grid. Consequently, the number of clusters and all cluster centers are efficiently determined.

2.2 Algorithm description

Let $A = \{A_1, A_2, \ldots, A_d\}$ be a set of domains under a metric space. The input S is a set of d-dimensional data objects $x_1, x_2, \ldots, x_n$. Let GRID=$A_1 \times A_2 \times \cdots \times A_d$ be the minimum bounding grid that contains all data objects. Hereinafter, the sign $|\cdot|$ indicates the number of objects related to the set in the bracket (e.g., $|S| = n$). The steps of the proposed algorithm are summarized as follows:

1) Successively bisect GRID in the following ways:

• The first (initial) round of bisecting: Bisect an edge of GRID into two halves. Accordingly, the GRID is bisected into two new volume-equal grids, denoted by G^{11} and G^{12}.

• The jth round of bisecting: At the $(j-1)$th round, bisect every grid into two volume-equal new grids and denote all 2^j obtained grids at the jth round of bisecting by G^{j1}, G^{j2}, $G^{j3}, \ldots,$ and G^{j2^j}.

2) Find an optimal grid size. All generated nonempty grids in the jth round of bisecting are ordered into three grid number-equal sets: $D(t, j), t = 1, 2, 3$, satisfying the requirement that the object number of any grid in $D(t, j)$ is larger than that of any other grid in $D(t+1, j), t = 1, 2$. The round corresponding to optimal grid size, say J, is determined by minimizing the ratio of $|D(3, j)|$ and $|D(1, j)|$; that is,

$$J = \text{argmin}_j\{|D(3, j)|/|D(1, j)|\}, \tag{1}$$

where $|D(3, j)|/|D(1, j)|$ is called a partitioning index.

3) Stop bisecting process when the optimum of (1) is determined.

To illustrate the G-K-MEANS algorithm, some definitions are introduced as follows.

Definition 1. Mean center and geometric center. The mean center of a cluster is the arithmetic average of all object vectors in the cluster while the geometric center of a cluster is the center of a minimal hypersphere that encloses all objects in this cluster.

Usually, the geometric center and the mean center of any cluster are different but are very close to each other if the cluster is symmetric (e.g., sphere(ellipsoid)-shaped).

Definition 2. Online grid and offline grid. Let J be an integer and S be a set of 2^J grids. A grid is called an online grid if the grid is one of top K high-density grids in S; any grid that is not an online grid in S is called an offline grid.

In the above bisecting process, we order the obtained 2^J grids and assign them into two sets $X_1 = \{G_1, G_2, \ldots, G_K\}$ and $X_2 = \{G_{K+1}, G_{K+2}, \ldots, G_{2^J}\}$ such that

$$|G_1| \geqslant |G_2| \geqslant \cdots \geqslant |G_K| \geqslant |G_{K+1}| \geqslant |G_{K+2}| \geqslant \cdots \geqslant |G_{2^J}|. \tag{2}$$

Thus X_1 is the set of online grids and X_2 is the set of offline grids.

In light of the basic idea that maximizes the average density of a group of grids, we propose the objective function of the G-K-MEANS algorithm in Table 1. To compare the differences between the G-K-MEANS algorithm and the K-means algorithm, we illustrate their basic steps in one-to-one way as follows.

In Table 1 the belongingness of any object x_i to the j-cluster/grid is shown by a Boolean vector such that u_{ij}=1 if x_i belongs to the cluster/grid, $i = 1, 2, \ldots, n$; otherwise to 0. Equation $(G_i - G_1 - \cdots - G_{i-1})$ indicates the set of objects in G_i not in G_1, $G_2, \ldots,$ or G_{i-1}. The volume V_i of ith grid is the product of all its edge sizes, for $i = 1, 2, \ldots, K$.

Definition 3. Regular cluster. A cluster is called a regular cluster if the geometric center and the mean center of the cluster are identical.

In the sense of geometric presentation, the concept of regular cluster generalizes not only the most used sphere-shaped clusters, but also center-symmetrical clusters such as ellipse-shaped cluster, hyperplane-shaped clusters, etc. Assuming that any cluster is regular, the basic steps of the G-K-MEANS algorithm are illustrated as follows.

• Objective function: The objective functions in the above two algorithms satisfy

$$\sum_{i=1}^{K}\sum_{j=1}^{n} u_{ij}d_{ij} \to \min \quad \mapsto \quad \sum_{i=1}^{K} |G_i - G_1 - \cdots - G_{i-1}|/V_i \to \max. \tag{3}$$

Table 1 The proposed G-K-MEANS algorithm

Algorithm: K-means	**Algorithm:** G-K-MEANS
Objective function:	**Objective function:**
Min $\sum_{i=1}^{K}\sum_{j=1}^{n} u_{ij}d_{ij}$, s.t. $\sum_{i=1}^{K} u_{ij}=1$	Max $\sum_{i=1}^{K} \lvert G_i - G_1 - G_2 - \cdots - G_{i-1}\rvert / V_i$
Input: The number of clusters K and a database containing n objects.	**Input:** The number of clusters K and a database containing n objects.
Output: A set of K clusters that minimizes the above objective function.	**Output:** A set of K clusters that maximizes the above objective function.
Method:	**Method:**
1. Arbitrarily choose K objects as the initial cluster centers, $v_1, v_2, \ldots, v_K$.	1. Choose K geometric centers of online grid as the initial mean centers, $v_1, v_2, \ldots, v_K$.
2. Repeat.	2. Repeat.
3. (Re)assign each object to the most similar cluster based on similar measure.	3. (Re)assign objects to G_i that is centralized on v_i, for $i = 1, 2, \ldots, K$.
4. Update cluster centers by $v_i = \sum_{j=1}^{n} u_{ij}x_j/u_{ij},\ i = 1, 2, \ldots, K$.	4. Update cluster centers by $v_i = \sum_{j=1}^{\lvert G_i\rvert} u_{ij}x_j/u_{ij},\ i = 1, 2, \ldots, K$.
5. Stop if a convergence criterion is met; otherwise, go to step 2.	5. Go to step 6 if a convergence criterion is met; otherwise, go to step 2.
	6. Stop if there are no overlapping grids in X_1; otherwise, add the offline grid with highest density into X_1; go to step 2.

In (3), the objective function of the K-means algorithm is reformulated by a grid-based form in the G-K-MEANS algorithm. Consequently, the minimization of the objective function in the K-means algorithm is replaced by the maximization of the average density of a group of online grids in X_1, G_i, $i = 1, 2, \ldots, K$. Thus the objective function in the K-means algorithm is directly data object-based while the one in the G-K-MEANS algorithm is grid-based.

• Object assigning principle: Let each cluster be surrounded by a separate minimal hypersphere. When an object falls in a hypersphere, the object usually is closer to the geometric center of the hypersphere than geometric centers of any other clusters. In the K-means algorithm, the object is assigned to the cluster based on the hyperspheres. Consequently, the object assigning principle is to iteratively find K minimal hyperspheres that enclose K clusters such that the geometric centers of these hyperspheres can be well determined. Contrarily, the G-K-MEANS algorithm iteratively applies K online grids that are centralized on v_1, $v_2, \ldots, v_K$ in place of K hyperspheres in the K-means algorithm. Namely an object is assigned to the ith cluster if the object uniquely falls into the grid G_i that stands for the cluster, for $i = 1, \ldots, K$. Finally the G-K-MEANS algorithm assigns any remaining object that is not in any grid to their closest center.

• Center updating: any new cluster center in the G-K-MEANS algorithm is locally computed by these objects limited to the grid that stands for the cluster. In contrast, the mean center in the K-means algorithm may be affected by any object no matter how far it is from the associated mean center.

• Online grid updating and stop criterion: If the distance between two mean centers of any two online grids is less than a threshold ε, the two online grids are considered as belonging to the same cluster and as the same online grid. The low-density one in the two online grids is removed from X_1 and the most high-density offline grid in X_2 is added into X_1 to proceed in the next iteration. In the G-K-MEANS algorithm, the iteration is terminated if the difference in the objective function values between two consecutive iterations is smaller than the threshold ε.

2.3 Algorithm analysis

In this section the G-K-MEANS is analyzed based on the basic steps, objective function, object assigning, efficiency in iterations, and converegence.

We use a dataset with three normally distributed clusters A, B, and C to illustrate the basic steps of the G-K-MEANS algorithm in Table 1 (see Figure 1).

In the example, sixteen initially partitioned nonempty grids numbered 1–16 are obtained according to (1). These grids can be partitioned into online grid and offline grid by (2) (see Figure 1(a)). Initially, G_6, G_7, and G_9 are three online grids while the other grids are offline grids, but none of the online grids is assigned to cluster B. First, the geometric centers of the three online grids are chosen as the mean centers (see step 1 in the G-K-MEANS algorithm). An object is assigned to one of the three online grids if it is within the grid (see step 3). Iteratively (step 2) all objects are reassigned into the associated grid (step 3) after the updates of the online grid center (step 4). Moreover, the mean centers of G_6 and G_7 move to the same cluster center of cluster A (step 4) and then are combined (step 6). Consequently, the G-K-MEANS algorithm chooses the offline grid G_{11} with highest density into the set of online grids (step 6) (see Figure 1(b)). Iteratively (step 2) G_{11} and G_7 get overlapped (steps 3 and 4). Similarly, once G_{10} is chosen as an online grid (step 6), it iteratively moves to overlap with G_7 (steps 2, 3 and 4). Finally, an online grid G_{16} is assigned to cluster B. The three cluster centers of cluster A, cluster B, and cluster C are accurately determined (step 5). This example demonstrates that the grid-moving process can obtain the optimal solutions of the G-K-MEANS algorithm and that given two online grids are assigned to the same regular cluster, the two grids must move to the same cluster center and overlap. Meanwhile, any cluster can be assigned to at least one online grid.

The G-K-MEANS algorithm originates from the K-means algorithm that mainly is designed for hypersphere-shaped clusters. For a set of K hypersphere-shaped clusters, if K mean centers suggested by the K-means algorithm accurately coincide with K geometric centers, the K minimal hyperspheres centered on the K mean centers can accurately partition all data objects. However, the K-means algorithm usually cannot attain its optimum at K geometric centers due to the limitations of its objective function, essentially for density-diverse, size-diverse and noisy data-contained clusters [3, 5, 9, 16, 17]. We further examine the objective function in the K-means algorithm by the following two noisy data-contained and object distribution-diverse datasets (see Figure 1).

The three cluster-contained dataset contains two tangential clusters and a separate cluster (see Figure 2(a)). It is clear that the three minimal circles centered on the related geometric centers can perfectly partition all objects into three parts (clusters) except those noisy objects. However, the three mean centers suggested by the K-means algorithm are very different from these geometric centers. Hence, the K-means algorithm must lead to numerous incorrect partitioned objects. The difference between the geometric center and mean center can also be observed in the second dataset (see Figure 2(b)), where the cluster A has evenly-distributed data objects while the cluster B consists of two half-circle, and the number of objects in a half-circle is twice that of the other. Indeed, the objective functions of the K-means algorithm attain their global minimums in the two datasets, respectively, but these minimums by no means lead to these geometric centers. In fact, in each dataset when we take these geometric centers into the objective function, the values of the objective function are larger than its global minimum (Table 2).

This demonstrates that the objective function in the K-means algorithm is not a good optimization criterion. In contrast, the average density of the group of minimal circles centered on the geometric centers always is their minimum in terms of the second equation in (3). Consequently, a reasonable and general assumption is to use the average density of K minimal hyperspheres in place of the objective function in the K-means algorithm. In the FCC algorithm [17], we have demonstrated that when the K minimal covers (e.g., hyperspheres, grids, etc.) centralized on K geometric centers enclose K regular clusters the average density of the K covers must be globally maximal. However, the determination of K minimal hyperspheres has to involve a very expensive distance computation across all objects, and hardly leads to time and space complexity reduction. Hence, in the G-K-MEANS algorithm we apply K grids for this purpose. Consequently, to maximize the equation $\sum_{i=1}^{K} |G_i - G_1 - \cdots - G_{i-1}|/V_i$ actually is to maximize the average density of all grids centralized on $v_1, v_2, \ldots, v_K$ individually. The grid-based average density in the G-K-MEANS algorithm is computed much more easily than the hypersphere-based objective function in the K-means algorithm. Besides, a cluster actually is a local description rather than a global one [3, 5]. In the G-K-MEANS algorithm each grid corresponds to a cluster coinciding with the notation. In contrast, the K-means algorithm globally assigns all objects to the K current mean centers

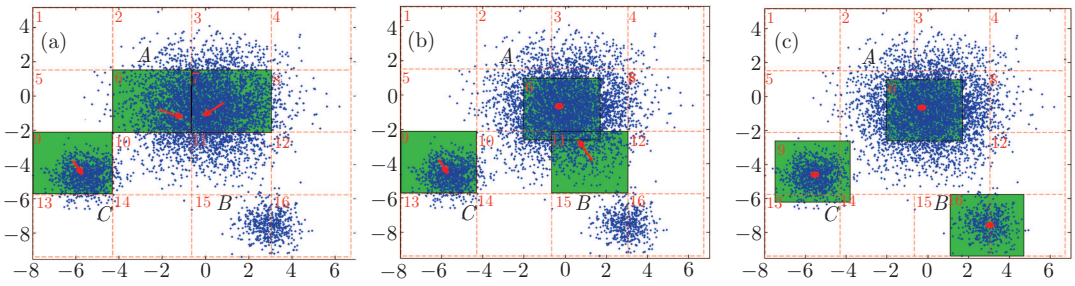

Figure 1 The iterative and updating processes of three online grids. Each red arrow indicates the moving direction of a grid, while each red symbol "·" in the figures refers to the final position that a grid moves to. (a) Initial partitions of data space and three online grids; (b) online grid G_7 that is overlapped with G_6 is removed and G_{11} becomes an online grid; (c) each cluster finally is uniquely assigned to an online grid that stands for the corresponding cluster, respectively.

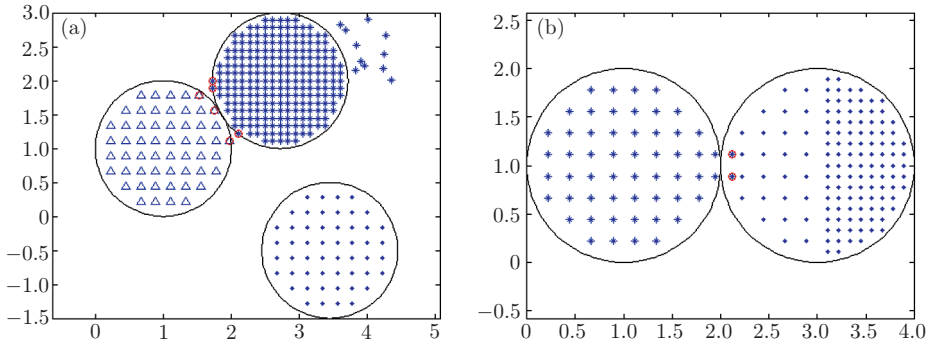

Figure 2 Two synthetic datasets with diverse characters. The points in each circle stand for the ones suggested by the K-means algorithm correctly, except that the objects colored in red are incorrectly partitioned by the K-means algorithm. (a) Three clusters in noisy objects (outside the three black circles); (b) two density-diverse clusters. Each black circle indicates the boundary of the minimal circles centered on the geometric center of the involved cluster and encloses all data.

Table 2 Objective function values in the K-means algorithm and the G-K-MEANS algorithm

Dataset	K-means		G-K-MEANS	
	Geometric centers	Mean centers	Geometric centers	Mean centers
Set 1	176.2	108.3	97.3	85.7
Set 2	96.9	80.4	93.1	90.6

at each iteration, no matter how far these objects depart from the K mean centers. This may lead to the fact that the K-means algorithm cannot assess these high-density areas that usually contain cluster prototypes.

The G-K-MEANS and the K-means algorithms employ the same optimization way for determining new centers (step 4) and thus their effects on optimization are the same. However, in the G-K-MEANS algorithm, the density of a grid will increase as the grid center tends to the cluster center For a regular cluster, if the geometric center of a grid and a cluster center are identical, the number of objects in the grid will be the largest among all grids related to the cluster. Otherwise, the grid will iteratively move to the cluster center in the direction from its geometric center to the mean center until the distance of the two centers is less than the threshold ε. This moving process will be completed in finite steps and thus all online grids independently converge to their individual cluster centers in finite steps. On the other hand, an irregular cluster can approximately consist of a number of regular clusters [3, 13, 17], and any grid will move to one of these regular clusters. Consequently, the convergence of the G-K-MEANS algorithm is guaranteed.

For n objects distributed in K clusters the computation of the G-K-MEANS algorithm mainly consists of three parts: 1) bisecting the data space for a number of rounds to a set of grids, 2) assigning n objects to K grids, and 3) computing all mean centers in iterations. The runtime of the second part is the

longest, which is similar to the K-means algorithm. Thus we compare the runtime of the second part of the K-means and the G-K-MEANS algorithms. The G-K-MEANS algorithm assigns n objects to K grids by data object inquiring $c \times n$ times. The K-means algorithm assigns n objects to K centers by distance computations which require $c \times n$ times computations followed by ordering these distances to find the nearest center of each object. In addition, according to the computational complexity conversion theory [12], the Euclidian distance computation of a pair of objects (vectors) in d-dimensional data space consists of about $1000d$ basic computational operations, while the inquiry of a pair of objects consists of 10 d basic computational operations. Therefore, the runtime of one iteration in the K-means algorithm is about 10–100 times more than that of the G-K-MEANS algorithm. It is apparent that G-K-MEANS algorithm has the great advantage of computation time. Thus the time complexity of the G-K-MEANS algorithm is $O(ctn)$ after t iterations while that of the K-means algorithm is $O(100ctn)$ at least. Furthermore, the G-K-MEANS algorithm does not need the computation to order all distances to find the nearest center of each object in the K-means algorithm. This further reduces its computational load. The G-K-MEANS algorithm and the K-means algorithm store n objects into memory for estimating each mean center and finding the nearest center for every object. Consequently, their maximal space complexities are $O(n)$.

3 Experiments

In this study, all the results are calculated by a desktop computer with 3.2 GHz CPU, 512 MB of RAM and Windows XP professional version 2002. We apply the G-K-MEANS algorithm to cluster three real high-dimensional datasets and six synthetic low-dimensional datasets of diverse cluster characters (e.g., density, size, and shape), and compare their clustering results with those suggested by the K-means, the DENCLUE, the GGCA, and the FWKA algorithms.

3.1 Sythetic datasets

Each cluster in the six synthetic datasets is generated by a "*randn*()" in the Matlab®toolbox. Thus each cluster is regular, and is centralized on the center of related "*randn*()" function. As a result, after labeling the above "*randn*()" functions, the correct cluster label of any data is just the label of the "*randn*()" function that generates these data. These original cluster labels in the above three datasets are not given in any clustering process, but they are used to examine the accuracy of algorithms after the clustering process is completed. The above six synthetic datasets are indicated by Set 1–Set 6. All the above datasets consist of two-dimensional data except that Set 6 consists of three-dimensional data. Consequently, the six artificial datasets all is low-dimensional but contains the most encountered characters in many datasets. Set 1 contains 6000 data in three clusters. One is a high-density cluster with 4000 data and the other two are low-density clusters with 1000 data each (see Figure 3(a)). These clusters are density-diverse and two of them are slightly overlapped. Set 2 contains 6000 data in three ellipse-shaped clusters, but is added with 1000 noisy data in random bivariate normal distributions with a large variance. Thus Set 2 is used to simulate a noisy dataset (see Figure 3(b)), and is applied to test the robustness of a clustering algorithm. Set 3 contains 6000 data with three size-diverse clusters, where the larger cluster has a diameter twice that of the two smaller clusters (see Figure 3(c)). In Set 4, there are 20000 data in 20 sphere-shaped and partially overlapping clusters (see Figure 3(d)). We use this dataset to test the runtime and space needs. Set 5 contains 366 data in five low-density ellipsoidal clusters and one high-density line-like cluster (see Figure 3(e)). The data in the six clusters are all evenly distributed. Set 6 consists of two clusters with 200 three-dimensional data in three mutually vertical coordinates. One cluster is distributed in the plane parallel to xoy plane and the other to yoz plane (see Figure 3(f)). Hence, the two clusters are distributed in two diverse subspaces of partial attributes respectivley.

3.2 Five real datasets

We have evaluated the efficiency of the G-K-MEANS algorithm by five real datasets from the UCI machine learning repository, *Iris*, *Satimage*, *Cancer*, *Letter*, and *Texture*. This three-cluster dataset

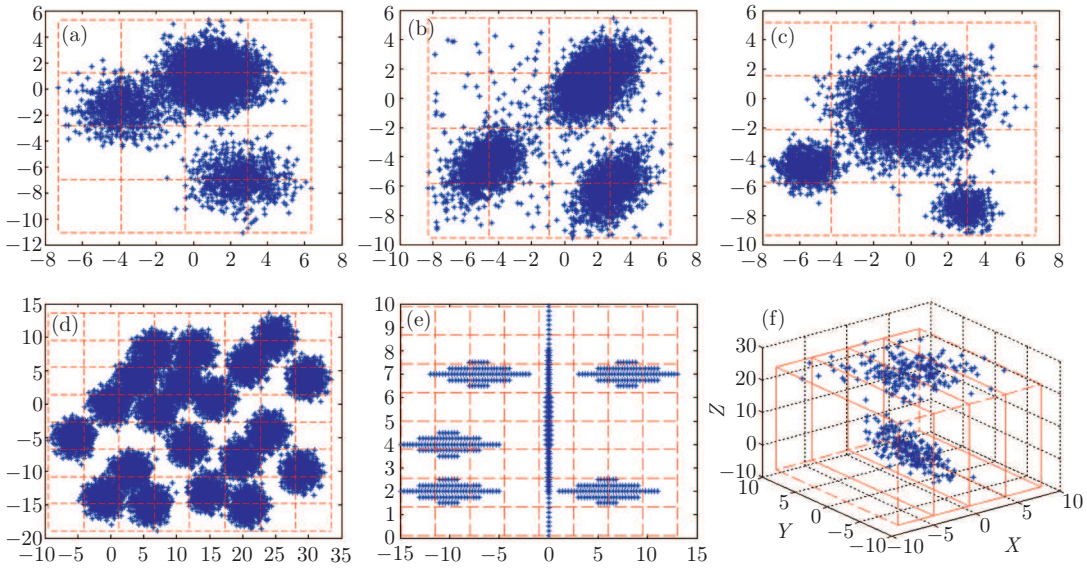

Figure 3 Synthetic datasets with diverse characters and grids in these datasets at the first iteration. (a) Density-diverse clusters; (b) clusters with noisy data; (c) size-diverse clusters; (d) low-dimensional and large-size clusters; (e) six arbitrary-shaped clusters; (f) two clusters in different subspaces.

Iris contains 150 data with 4 attributes. Each cluster has 50 data equally. Two clusters are overlapped and one cluster is separated from those two clusters. In the past decades, the dataset has been frequently applied to evaluate the effectiveness of clustering algorithms [16].

The *Satimage* dataset consists of data from earth resource satellite-generated multi-spectral imaging scan (MSS), and each frame in MSS contains 4 different frequency components of the digital image in the same scene. Each datum in the *Satimage* dataset is a 36-dimensional data sample. The *Satimage* dataset, denoted by *Satimage*, contains 6435 data samples that belong to 6 different clusters.

The *Cancer* dataset contains 699 instances that originally belong to two hyperplane-shaped clusters named "Benign" and "Malignant", and each instance has ten attributes plus the cluster attribute. In the *Cancer* dataset the two clusters are regular. There are 16 missing instances in the original dataset, which we removed.

The *Letter* dataset contains twenty-six clusters with 20000 records and 16 attributes. Hence, the *Letter* dataset is a representative of high-dimensional and very large datasets. However, the *Letter* dataset itself is inappropriate evaluate a clustering algorithm since there are no clear boundaries among different clusters in it [13]. Hence, we modify the *Letter* dataset by deleting any two records whose cluster labels are different and mutual distance is less than a small threshold 0.02. The modified dataset is denoted by *M-Letter* that contains 18360 records.

The *Texturedataset* consists of 4000 patterns in a 19-dimensional feature space, and represents an image with 4 distinct textures (clusters). This is a difficult data set due to the overlap between nonlinear-seperated clusters.

The original clustering labels are not used for the clustering, instead, the labels are used to evaluate the accuracy of the algorithm.

3.3 Applied algorithms

We cluster the above datasets by six algorithms: K-means, G-K-MEANS, DENCLUE, SHIFT, FWKA and GGCA, where the GGCA algorithm, which was proposed by us previously, is the representative of the grid-based clustering approaches [13]. *Apriori* number of clusters for each dataset is suggested to all the above algorithms and a commonly acceptable error 10^{-5} is used to measure the difference between the result of the previous iteration and that of the current iteration. To determine the optimal grid size for a given dataset, the G-K-MEANS algorithm first finds its global minimum from (1) after implementing a

number of rounds of bisecting, and then searches for the corresponding round J along with the minimum. The values of $D(3,j)/D(1,j)$ in each dataset are shown in Figure 4. The number of initial grids in the G-K-MEANS algorithm is 2^T subject to $K < 2^T < 2^J$, where J is determined by (1). Thus we use 16, 16, 16, 64, 64, and 4 initial grids for Set 1–Set 6, respectively. In the DENCLUE algorithm we partition the data space and obtain the set of grids whose number is forty times more than the number of clusters in the corresponding dataset. To minimize the impacts from various initial setting, the best result from all possible initial settings is chosen for both K-mean and FWKA algorithms.

In case of five real datasets such as *Iris*, *Satimage*, *Cancer*, *MLetter*, and *Texture*, we use (1) to get the optimal grid size (see Figure 4(b)). Consequently, the numbers of their initial grids are 4, 16, 16, 64 and 32, respectively. To efficiently cluster the high-dimensional data, the G-K-MEANS algorithm uniformly transforms each in the above 2^J grids to an edge-equal grid which has the same center as the original grid, and has edge width e, satisfying

$$e = \sqrt[d]{\lambda V/2^J}, \quad \text{for } i = 1, 2, \ldots, 2^J. \tag{4}$$

Here λ is a ratio of the volume of all nonempty grids after bisecting the GRID to a group of grids in which each contains only one object, and V is the volume of the GRID.

In all experiments, we coded the DENCLUE algorithm according to the basic steps available in [10]. The normalized attribute space is partitioned along each numerical dimension with width $\sigma = 0.02$ for the DENCLUE algorithm. The GGCA and SHIFT algorithms are implemented in house. We apply the K-means and the FWKA algorithms to implement the clustering analysis in the data mining workshop: AlphaMiner 2.0 [18].

3.4 Clustering results

The clustering results of the above algorithms are evaluated by four indices: accuracy, CPU possessing time (seconds), robustness, and initialization, where the accuracy is measured by the percentage of correctly-partitioned data in each dataset. For the G-K-MEANS algorithm, the CPU possessing time for list in each dataset includes the time of all rounds of bisecting for the optimal grid size and the iterative processing to seek for optimal solutions. All the clustering results are listed in Table 3, and the clustering results of Set 1–Set 6 are further shown in Figure 5.

We explain these results as follows.

(1) Accuracy. The G-K-MEANS algorithm is better than the other four algorithms in Set 1–Set 4, and *Satimage*, but is worse than those in Set 5, Set 6, *Cancer* and *M-Letter*. Figures 5(a)–5(f) show that the mean centers suggested by the K-means algorithm in Set 1–Set 6 partially deviate from their actual geometric centers due to very density-diverse and tangential clusters in Set 1, noisy data in Set 2, size-different clusters in Set 3, overlapped clusters in Set 4, shape-different clusters in Set 5, and attribute-different clusters in Set 6. Thus the K-means algorithm has the lowest accuracy among the above five algorithms. In contrast, the mean centers obtained by the G-K-MEANS algorithm coincide with the true geometric centers of the clusters in the datasets. Consequently, a majority of the clusters in Set 1–Set 4 are partitioned correctly. However, in Set 5, the G-K-MEANS algorithm is capable of finding six mean centers correctly, with some objects in the sixth clusters not satisfying the nearest neighbor assigning principle. Take Set 6 as an example. The grid centers in the G-K-MEANS algorithm gradually move to the two hyperplanes of clusters A and B (see Figure 5(f)), yet the results are not better than those of the GGCA and FWKA algorithms. In contrary, the DENCLUE and GGCA algorithms get the better clustering results for Set 5 and Set 6 than the other three algorithms, since they can identify arbitrarily shaped clusters. Specifically, the clustering results of the FWKA algorithm show that it performs well at locating clusters distributed in different subspaces.

The *Iris*, *Satimag* and *Cancer* all are approximately regular cluster-contained dataset but the *M-Letter* and the *Texture* are not. The difference leads to very different accuracies of the G-K-MEANS algorithm. In the *Satimag* each regular cluster is spherical, and the G-K-MEANS algorithm outperforms the other four algorithms. In the *Cancer* with regular but non-spherical cluster-contained datasets, the

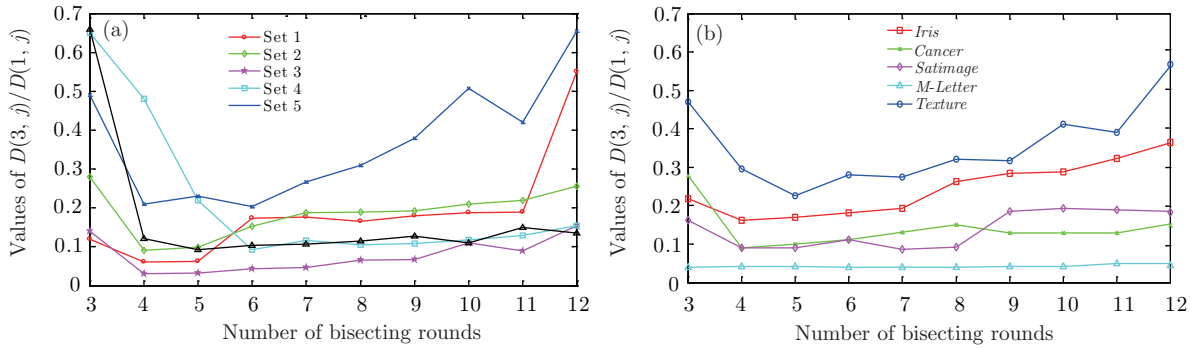

Figure 4 Values of $D(3,j)/D(1,j)$ in a number of bisecting rounds in (a) Set 1–Set 6, and (b) five real datasets.

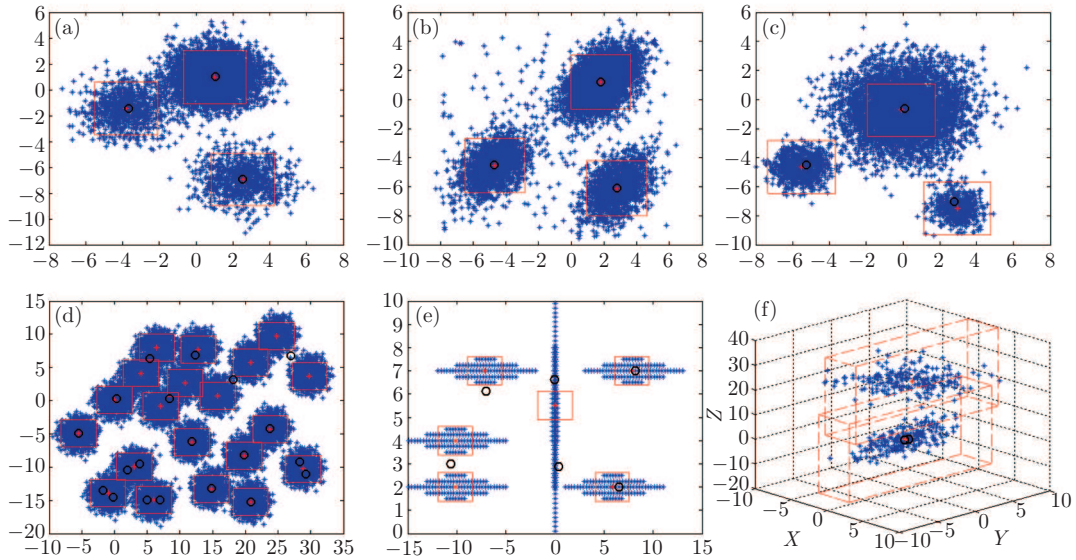

Figure 5 Clustering results of the six synthetic datasets. (a)–(f) correspond to Set 1–Set 6 respectively. Each red symbol "•" indicates a grid center by the G-K-MEANS algorithm, each black symbol "○" indicates a cluster center by the K-means algorithm, and each red square indicates an online grid (hypercube) when iterative stop condition is satisfied.

Table 3 Clustering results of the experimental datasets[a]

Algorithms	K-means	DENCLUE	FWKA	SHIFT	GGCA	G-K-MEANS
Datasets	$A(\%)/C(\text{s})$	$A(\%)/C(\text{s})$	$A(\%)/C(\text{s})$	$A(\%)/C(\text{s})$	$A(\%)/C(\text{s})$	$A(\%)/C(\text{s})$
Set 1	86.2/11.62	88.3/8.78	89.2/14.57	89.2/14.57	88.3/0.26	97.3 /0.15
Set 2	91.9/12.32	89.2/0.78	92.6/13.72	92.6/13.72	89.2/0.95	93.1 /0.19
Set 3	87.8/14.15	93.3/1.78	91.4/18.53	91.4/18.53	93.3/2.11	95.2 /0.26
Set 4	65.3/12.26	87.6/19.72	90.8/23.21	90.8/23.21	87.6/8.02	93.2 /0.38
Set 5	72.1/0.61	90.4/2.18	87.3/1.87	87.3/1.87	93.7 /3.88	88.7/0.87
Set 6	64.3/0.63	76.3/1.19	86.3/1.83	86.3/1.83	94.3 /2.33	80.3/0.27
Iris	90.1/0.11	93.4/1.02	92.7/0.13	90.3/0.12	96.2 /0.28	95.3/0.16
Satimage	44.3/18.23	66.3/12.17	58.7/13.53	58.7/13.53	70.2/10.64	73.5/6.46
Cancer	70.4/0.33	89.2/1.19	88.9/8.13	88.9/8.13	96.1 /4.19	74.5/1.46
M-Letter	34.3/114.63	87.9/83.41	76.3/172.3	76.3/172.3	92.9 /8.11	44.7/3.28
Texture	76.8/7.23	95.3 /10.21	85.4/11.15	88.7/14.13	94.2/5.64	86.5/6.46

a) For any clustering algorithm, "$A(\%)$" indicates accuracy (percentage), "$C(\text{s})$" CPU runtime (second). The underlined item is the best result in any corresponding row.

G-K-MEANS algorithm can find the cluster centers better than the other five algorithms except the GGCA. However, these well-determined cluster centers cannot lead to the highest accuracy since the

object assigning principle of the G-K-MEANS algorithm naturally regards all clusters as spherical ones. In the *M-Letter* and *Texture* with irregular and arbitrary-shaped clusters, the G-K-MEANS algorithm is worse than the DENCLUE, GGCA and FKWA algorithms, but still is better than the K-means algorithm.

The performance of the G-K-MEANS algorithm is summarized as follows. First, if all clusters in a given dataset are regular in sphere geometry, the G-K- MEANS algorithm has the best accuracy among the four algorithms without considering the dimensionality of the dataset. In this case, the G-K-MEANS algorithm outperforms the K-means algorithm significantly. Second, if all clusters are approximately regular but not spherical, the G-K-MEANS algorithm can find correct cluster centers, but partial objects that are against the nearest neighbor principle may be assigned to cluster centers/labels incorrectly. Finally, if some of the clusters in the dataset are irregular, the merit of the G-K-MEANS algorithm is to find the objects distributed in high-density areas of clusters. However, other objects that are not located in high-density area may be difficult to be partitioned correctly. Essentially, the G-K-MEANS algorithm cannot efficiently handle arbitrarily geometric shapes compared with the existing grid-clustering algorithms. For example, the accuracy of the G-K-MEANS algorithm is 86.5%. The accuracy of K-means algorithm with fixed k= 4 (i.e., four prototypes) is 768%. But the SHIFT, DENCLUE and GGCA have higher accuracies of 88.9%, 95.3% and 942%, respectively. Thus for arbitrary-shape clusters the existing grid-clustering algorithms with the necessary *apriori* knowledge may outperform the G-K-MEANS algorithm.

(2) CPU processing time. For the largesize datasets Set 4 and *M-Letter*, Table 3 shows that the time cost of the G-K-MEANS algorithm is less that of the other five algorithms In the *M-Letter* dataset, the runtime of the K-means algorithm for different initial conditions are almost 10–100 times more than that of the G-K-MEANS algorithm. However, for the smallsize datasets such as Set 5, Set 6 and *Cancer*, the CPU processing time of the G-K-MEANS algorithm is longer than those of the other four algorithms since a number of rounds of bisecting are performed to find the optimal grid size. In general, the runtime of the G-K-MEANS algorithm is directly proportional to the number of initial grids rather than that of data in a dataset.

(3) Initialization and the estimation of the number of clusters. Table 4 shows the clustering results with different initializations for the five algorithms. The result is measured by the differences between the worst and the best accuracies after applying 1) different initial cluster centers in the K-means and FWKA algorithm; 2) different parameter pairs of p and q in the GGCA; 3) different parameter settings of σ in the DENCLUE; 4) all combinations of the chosen dimensionalities in each round of bisecting in the G-K-MEANS algorithm.

Table 4 shows that the clustering results of the G-K-MEANS algorithm can be affected by different bisecting dimensions in each round of bisecting to some extent, but the affection is smaller than the K-means and the FWKA algorithms. In *Cancer* and *M-Letter*, the G-K-MEANS algorithm is inferior to the DENCLUE algorithm. However, the performance of G-K-MEANS is much improved when (4) is applied at each round of bisecting. In addition, the GGCA and SHIFT algorithms are impacted by different initial conditions mostly compared with the G-K-MEANS algorithm, and are not capable of resolving the common issue existing in the grid-based clustering algorithms.

If the number of clusters is unknown *apriori*, the G-K-MEANS algorithm can use all initial grids as online girds to partition any given dataset. In this case, each cluster will be assigned a grid at least so that none of all clusters is missed. At the same time, all grids that are assigned to the same cluster move to the same center of the cluster, so that these grids overlap and only the most high-density grid in them is kept and the other grids are removed. Consequently, the clustering accuracies of all datasets by the G-K-MEANS algorithm are comparable to those shown in Table 3. Since (1) can determine the number of initial grids, the clustering does not require any user-determined parameter. However, to use all initial grids as online grids we have to increase the CPU possessing time in every dataset, as shown in the final column in Table 4. This will decrease the efficiency of the G-K-MEANS algorithm in the CUP processing time to some extent.

Table 4 Additional accuracy under diverse initializations and extra CPU processing time for the number of clusters[a)]

Algorithm	K-means	FWKA	DENCLUE	GGCA	SHIFT	G-K-MEANS
Datasets	$A(\%)$	$A(\%)$	$A(\%)$	$A(\%)$	$A(\%)$	$A(\%)/C(\%)$
Set 1	10.8	8.4	12.1	10.7	13.8	1.4 /112.7
Set 2	10.2	6.7	8.2	11.3	13.2	5.2 /341.2
Set 3	9.4	10.8	8.8	61	93	2.8 /234.2
Set 4	11.3	9.2	14.2	17.2	181	6.2 /145.6
Set 5	11.8	15.7	12.1	10.1	13.5	4.1 /238.4
Set 6	19.2	8.2	11.3	134	122	6.9 /278.3
Satimage	15.8	15.1	7.1	8.1	7.3	5.1 /137.2
Cancer	12.2	10.6	8.2	19.2	17.7	11.3/763.2
M-Letter	19.4	15.3	11.3	22.3	20.6	15.3/549.8

a) For any clustering algorithm, the symbol "$A(\%)$" indicates the additional accuracy (percentage) of any dataset between the best and the worst results, and "$C(\%)$" the extra runtime (percentage) of the G-K-MEANS algorithm to find the unknown number of clusters. The underlined item is the best result in any corresponding row.

4 Conclusions

We focus on the integration of the grid-based mechanism to the K-means algorithm and thus develop a grid-based K-means (G-K-MEANS) algorithm. The G-K-MEANS algorithm tends to find cluster centers in high-density areas while the K-means cannot guarantee this point. Therefore, the cluster quality should be better from a density viewpoint. Moreover, the G-K-MEANS algorithm assigning objects to grids needs much less computation time than existing algorithms such as the K-means algorithm, and hierarchical algorithm of large datasets [18, 19] which requires computing the Euclidean distances. Thus the G-K-MEANS algorithm is more efficient. Besides, the G-K-MEANS algorithm can provide a fast and efficient initial partition for the data space. This is very helpful for further cooperation with other clustering algorithms such as PCM (probabilistic clustering method) [20], etc. Compared to the existing grid-based clustering algorithms, the G-K-MEANS algorithm is not involved in the three main problems regarding the determination of grid parameters mentioned in section 1. Our study reveals that the objective function of the K-means algorithm is not the best optimal criterion for a given pattern set, while the one of the G-K-MEANS algorithm is a better candidate. Due to these promising performances, we believe that the G-K-MEANS algorithm is quite reasonable and applicable to solving very large datasets.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant Nos. 60772080, 60572065, 50974095), and the Tianjin Science Foundation (Grant No. 08JCYBJC13800).

References

1 MacQueen J B. Some methods for classification and analysis of multivariate observations. In: The 5th Berkeley Symposium on Mathematical and Probability. Berkeley, 1967, 1: 281–297
2 Jenssen R, Erdogmus D, Hild K, et al. Information cut for clustering using a gradient decent approach. Patt Recogn, 2007, 40: 796–806
3 Xu R, Wunsch D. Survey of clustering algorithm. IEEE Trans Neur Netw, 2005, 16: 645–678
4 Guha S, Rastogi R, Shim K. CURE: An efficient clustering algorithm for large databases. Inf Syst, 2001, 26: 35–58
5 Pedrycz W. Fuzzy clustering with a knowledge-based guidance. Patt Recogn Lett, 2004, 25: 469–480
6 Huang Z, Ng M K, Rong H. Automated variable weighting in k-means type clustering. IEEE Trans Patt Anal Mach Intell, 2005, 27: 657–668
7 Jing L, Gao Y, Wu G, et al. Feature weighting k-means algorithm for large-scale documents clustering. In: Proc. 1st China Classification Conf. Beijing, China, 2005, 1: 85–90

8 Fabrizio S. Machine learning in automated text categorization. ACM Comput Surv, 2002, 34: 1–47
9 Yu J. General C-means clustering model. IEEE Trans Patt Anal Mach Intell, 2003, 25: 1197–1211
10 Hinneburg A, Keim D A. An efficient approach to clustering in large multimedia databases with noise. In: Proc Int Conf Knowl Disc Data Mining, New York, 1998. 58–65
11 Eden W, Ma M, Tommy W S. A new shifting grid clustering algorithm. Patt Recogn, 2004, 37: 503–514
12 Agrawal R, Gehrke J, Gunopulos D. Automatic subspace clustering of high dimensional data. Data Mining. Knowl Discov, 2005, 11: 5–33
13 Yue S, Wei M, Wang J. A general grid-based approach to clustering. Patt Recogn Lett, 2008, 29: 1372–1384
14 Ordonez C, Omiecinski E. Efficient disk-based K-means clustering for relational databases. IEEE Trans Knowl Data Eng, 2004, 16: 909–921
15 Chiang J, Yin Z. Unsupervised minor prototype detection using an adaptive population partitioning algorithm. Patt Recogn, 2007, 40: 3132–3145
16 Parizeau M, Lee S W. A fuzzy-syntactic approach to allograph modeling for cursive script recognition. IEEE Trans Patt Anal Mach Intell, 1995, 17: 702–712
17 Chiang H, Yue S, Yin Z. A new fuzzy cover approach to clustering. IEEE Trans Fuzzy Syst, 2004, 12: 199–208
18 AlphaMiner2.0: http://bi.hitsz.edu.cn/alphaminer/index.htm
19 Zalik K. An efficient k-means clustering algorithm. Patt Recogn Lett, 2008, 29: 1385–1391
20 Krishnapuran R, Keller J M. A possiblistic c-means algorithm. IEEE Trans Fuzzy Syst, 1993, 2: 100–112