Information Sciences 245 (2013) 168–180

Contents lists available at SciVerse ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Fast global *k*-means clustering based on local geometrical information

Liang Bai [a], Jiye Liang [a,*], Chao Sui [b], Chuangyin Dang [c]

[a] School of Computer and Information Technology, Shanxi University, Taiyuan, 030006 Shanxi, China
[b] Shanxi Academy of Agricultural Sciences, Taiyuan, 030006 Shanxi, China
[c] Department of System Engineering and Engineering Management, City University of Hong Kong, Hong Kong

ARTICLE INFO

Article history:
Received 4 January 2012
Received in revised form 8 May 2013
Accepted 18 May 2013
Available online 28 May 2013

Keywords:
Cluster analysis
Optimization
Global *k*-means clustering
Local geometrical information
Computational complexity

ABSTRACT

The fast global *k*-means (FGKM) clustering algorithm is one of the most effective approaches for resolving the local convergence of the *k*-means clustering algorithm. Numerical experiments show that it can effectively determine a global or near global minimizer of the cost function. However, the FGKM algorithm needs a large amount of computational time or storage space when handling large data sets. To overcome this deficiency, a more efficient FGKM algorithm, namely FGKM+A, is developed in this paper. In the development, we first apply local geometrical information to describe approximately the set of objects represented by a candidate cluster center. On the basis of the approximate description, we then propose an acceleration mechanism for the production of new cluster centers. As a result of the acceleration, the FGKM+A algorithm not only yields the same clustering results as that of the FGKM algorithm but also requires less computational time and fewer distance calculations than the FGKM algorithm and its existing modifications. The efficiency of the FGKM+A algorithm is further confirmed by experimental studies on several UCI data sets.

1. Introduction

Clustering is an important problem in statistical multivariate analysis, data mining and machine learning [12]. The goal of clustering is to group a set of objects into clusters so that the objects in the same cluster are highly similar but remarkably dissimilar with objects in other clusters [20]. To tackle this problem, various types of clustering algorithms have been developed in the literature (e.g., [14] and references therein). Among them, the *k*-means clustering algorithm [17] is one of the most efficient clustering algorithms for large-scale spherical data sets. It has extensive applications in such domains as financial fraud, medical diagnosis, image processing, information retrieval, and bioinformatics.

The *k*-means clustering algorithm uses the alternating minimization method to solve a nonconvex optimization problem in finding cluster solutions [14]. However, the obtained clustering results guarantee local optimization solutions only [20]. To solve this problem, several techniques have been developed based on different global search methods, such as simulated annealing [6,11], genetic algorithms [3,15,19,23], colony optimization [2,27], particle swarm optimization [1,16], stochastic optimization [9], and black hole algorithm [13]. Among these methods, the fast global *k*-means clustering (FGKM) algorithm proposed by Likas et al. [22] is a very effective search approach, which uses the incremental learning technique to solve the local minimum problem. The numerical experiment results [4] have shown that the FGKM algorithm can determine a global

* Corresponding author. Tel.: +86 0351-7018176.
E-mail addresses: sxbailiang@hotmail.com (L. Bai), ljy@sxu.edu.cn (J. Liang), suichao@outlook.com (C. Sui), mecdang@cityu.edu.hk (C. Dang).

0020-0255/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ins.2013.05.023

or near global minimizer of the k-means objective function. Nevertheless, the FGKM algorithm requires calculating the distances between any two data objects in each iteration. For a small data set, we can use $O(n^2)$ memory space to store distances and avoid repeated computations, where n is the number of data objects. However, for a large data set, storing these distances is unfeasible. For instance, if a data set has $n = 10^6$ objects, storing the distances between all objects (assuming double precision storage) requires 8 TB of memory, which is unavailable on a general purpose machine [7].

To make the FGKM algorithm more effective, a modified global k-means (MGKM) algorithm was proposed in [4]. The algorithm minimizes an auxiliary objective function to determine new cluster centers. Compared with the FGKM algorithm, the MGKM algorithm can obtain a slightly better result but with a longer computational time [18]. Bagirov et al. [5] suggested a new version of the MGKM algorithm to reduce the computational time of the clustering process and obtain an approximation result. Likas et al. [22] proposed the kd-tree to speed up the generation of new cluster centers in the FGKM algorithm. Unfortunately, the kd-tree-based algorithm is unsuitable for data sets with high dimensions, given that its computational complexity grows exponentially with the data dimensions [18,10]. Lai and Huand [18] presented a fast search algorithm by using projection and inequality to reduce the number of distance calculations in determining new cluster centers, which is called the MFGKM algorithm. This algorithm can expedite the FGKM algorithm while retaining its effectiveness. The projection and inequality in the MFGKM algorithm is based on global geometrical information, whereas clusters tend to exist in the local geometrical spaces of real data sets. In addition, local geometrical information can provide us with a smaller search space than global geometrical information. These facts lead us to the development of a more efficient FGKM algorithm that makes use of the local geometrical information in the expedition of the search process of the FGKM algorithm while retaining the same clustering results.

The rest of this paper is organized as follows. Section 2 reviews the k-means and FGKM algorithms. Section 3 presents a more efficient FGKM algorithm (i.e., FGKM+A). Section 4 analyzes the space and computational complexity of the proposed algorithm. Section 5 illustrates the effectiveness of the proposed algorithm on real data sets. Finally, Section 6 concludes the paper with some remarks.

2. The *k*-means and FGKM algorithms

Let $U = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ be a set of n objects. Object $\mathbf{x}_i = \{x_{i1}, x_{i2}, \ldots, x_{im}\}$ is characterized by a set of m attributes (variables). The k-means algorithm [17] searches for a partition of U into k clusters that minimizes the objective function F with unknown variables W and V:

$$F(W,V) = \sum_{l=1}^{k}\sum_{i=1}^{n} w_{li}\|\mathbf{x}_i - \mathbf{v}_l\|^2, \tag{1}$$

subject to

$$w_{li} \in \{0,1\}, \sum_{l=1}^{k} w_{li} = 1, 0 < \sum_{i=1}^{n} w_{li} < n, \quad 1 \leqslant l \leqslant k, \quad 1 \leqslant i \leqslant n. \tag{2}$$

where

- $W = [w_{li}]$ is a k-by-n $\{0,1\}$ matrix, w_{li} is a binary variable, and indicates whether object $\mathbf{x}_i$ belongs to the lth cluster, $w_{li} = 1$ if x_i belongs to the lth cluster and 0 otherwise;
- $V = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k]$ and $\mathbf{v}_l = [v_{l1}, v_{l2}, \ldots, v_{lm}]$ is the lth cluster center with m attributes;
- $\|\mathbf{x}_i - \mathbf{v}_l\|^2 = \sum_{j=1}^{m}(x_{ij} - v_{lj})^2$ is Euclidean distance between the object $\mathbf{x}_i$ and the lth cluster center $\mathbf{v}_l$.

The minimization of F in (1) with the constraints in (2) forms a class of constrained nonlinear optimization problems whose solutions are unknown. The usual method toward the optimization of F in (1) is to use partial optimization for V and W. In this method, we first fix V and find necessary conditions on W to minimize F. Thereafter, we fix W and minimize F with respect to V. The above optimization problem can be solved by iteratively solving the following two minimization problems.

- Problem P_1: Fix $V = \widehat{V}$, solve the reduced problem $F(W, \widehat{V})$;
- Problem P_2: Fix $W = \widehat{W}$, solve the reduced problem $F(\widehat{W}, V)$.

Problem P_1 is solved by

$$w_{li} = \begin{cases} 1, & \text{if} \quad \|\mathbf{x}_i - \hat{\mathbf{v}}_l\|^2 \leqslant \|\mathbf{x}_i - \hat{\mathbf{v}}_h\|^2, 1 \leqslant h \leqslant k, \\ 0, & \text{otherwise}. \end{cases} \tag{3}$$

for $1 \leqslant i \leqslant n$, $1 \leqslant l \leqslant k$.

Problem P_2 is solved by

$$v_{lj} = \frac{\sum_{i=1}^{n} \hat{w}_{li} x_{ij}}{\sum_{i=1}^{n} \hat{w}_{li}}, \tag{4}$$

for $1 \leqslant l \leqslant k$, $1 \leqslant j \leqslant m$.

This process is formalized in the k-means algorithm [17]:

The k-means algorithm

Step 1. Choose an initial point set $V^{(1)} \in R^{mk}$. Determine $W^{(1)}$ such that $F(W, V^{(1)})$ is minimized. Set $t = 1$.
Step 2. Determine $V^{(t+1)}$ such that $F(W^{(t)}, V^{(t+1)})$ is minimized. If $F(W^{(t)}, V^{(t+1)}) = F(W^{(t)}, V^{(t)})$, then stop; otherwise goto Step 3.
Step 3. Determine $W^{(t+1)}$ such that $F(W^{(t+1)}, V^{(t+1)})$ is minimized. If $F(W^{(t+1)}, V^{(t+1)}) = F(W^{(t)}, V^{(t+1)})$, then stop; otherwise set $t = t + 1$ and goto Step 2.

Since the time complexity of the algorithm is $O(nkmt)$, it can efficiently cluster large data sets. However, the obtained clustering results guarantee local minimum solutions only. Thus, the performance of the algorithm heavily depends on initial cluster centers.

The global k-means clustering algorithm introduced by [22] constitutes a deterministic global optimization method that is independent of any initial parameter values and employs the k-means algorithm as a local search procedure. Instead of randomly selecting initial values for all cluster centers as is the case with most global clustering algorithms, the proposed technique proceeds in an incremental manner to add optimally a new cluster center at each stage.

According to (3), we can obtain W and minimize $F(W, \widehat{V})$ when $\widehat{V}$ is given. Therefore, the objective function (1) can be rewritten as follows:

$$\mathbb{F}(V) = \min_{W} F(W, V) = \sum_{i=1}^{n} \min_{\mathbf{v}_l \in V} \|\mathbf{x}_i - \mathbf{v}_l\|^2. \tag{5}$$

The global k-means algorithm is briefly described as follows [22]:

The global k-means algorithm

Step 1. Compute $V_1^* = \{\mathbf{v}_1\}$ from the data set U, where $\mathbf{v}_1 = \sum_{i=1}^{n} \mathbf{x}_i / n$ and n is the number of objects in U. Set $h = 1$.
Step 2. Set $h = h + 1$. If $h > k$, then stop.
Step 3. For each object $\mathbf{x}_i \in U$, apply the k-means algorithm with the initial set of cluster centers $V_{h-1}^* \cup \{\mathbf{x}_i\}$ and obtain the resulting set of cluster centers $V_h(i) = \{\mathbf{v}_1(i), \mathbf{v}_2(i), \ldots, \mathbf{v}_h(i)\}$.
Step 4. Set $V_h^* = V_h(r)$ which satisfies

$$\mathbb{F}(V_h(r)) = \min_{i=1}^{n} \mathbb{F}(V_h(i)),$$

and goto Step 2.

When handling large data sets, the global k-means algorithm is inefficient, since it has a time complexity of $O(n^2mk^2t)$. Therefore, several modified algorithms have been proposed to reduce the computational load.

Likas et al. [22] proposed a FGKM algorithm, which is described as follows:

The FGKM algorithm

Step 1. Compute $V_1^* = \{\mathbf{v}_1\}$ from the data set U, where $\mathbf{v}_1 = \sum_{i=1}^{n} \mathbf{x}_i / n$ and n is the number of objects in U. Set $h = 1$.
Step 2. Set $h = h + 1$. If $h > k$, then stop.
Step 3. For each object $\mathbf{x}_i \in U$, compute

$$b_h^i = \sum_{j=1}^{n} \max\left(0, d_{h-1}^j - \|\mathbf{x}_j - \mathbf{x}_i\|^2\right)$$

where $d_{h-1}^j = \min_{\mathbf{v}_l \in V_{h-1}^*} \|\mathbf{v}_l - \mathbf{x}_j\|^2$.
Step 4. Set $V = V_{h-1}^* \cup \{\mathbf{x}_q\}$ which satisfies

$$b_h^q = \max_{i=1}^{n} b_h^i.$$

Step 5. Apply the k-means algorithm with the initial set of cluster centers V, save the resulting set of cluster centers into V_h^* and compute d_h^i for each $\mathbf{x}_j \in U$. Goto Step 2.

Compared with the global k-means clustering algorithm, the FGKM algorithm does not execute the k-means algorithm for each data object in Step 3. Instead, the FGKM computes an upper bound $\mathbb{F}(V_h(i)) \leqslant \mathbb{F}(V_{h-1}^*) - b_h^i$ making the time complexity $O(n^2mk + nmk^2t)$. The numerical experiment results in [4] have shown that the FGKM algorithm can determine a global or near global minimizer of the objective function.

3. The FGKM+A algorithm

In the FGKM algorithm, computing b_h^i for each $\mathbf{x}_i$ in U is an important step, given that b_h^i is necessary to determine which object in U will be the initial center of a new cluster. However, the process is extremely time consuming. Here, $b_h^i = \sum_{\mathbf{x}_j \in P_h(\mathbf{x}_i)} \left(d_h^j - \|\mathbf{x}_j - \mathbf{x}_i\|^2 \right)$, where $P_h(\mathbf{x}_i) = \{\mathbf{x}_j | \|\mathbf{x}_j - \mathbf{x}_i\|^2 \leqslant d_{h-1}^j, \mathbf{x}_j \in U\}$, for $1 \leqslant i \leqslant n$. The FGKM algorithm needs to identify a set $P_h(\mathbf{x}_i)$ from U by calculating the distance between each object $\mathbf{x}_j$ in U and $\mathbf{x}_i$. The entire process needs n^2 distance calculations. In this section, we propose an acceleration mechanism to reduce the computing cost of the process.

The mechanism will enhance the efficiency of the FGKM algorithm in the following two aspects:

- Instead of directly computing $\|\mathbf{x}_j - \mathbf{x}_i\|^2$ for each object $\mathbf{x}_j \in U$, we will compute its estimated value and determine whether $\mathbf{x}_j$ belongs to $P_h(\mathbf{x}_i)$.
- Given that the FGKM algorithm requires only the determination of the object $\mathbf{x}_i$ with the maximum b_h^i, computing the exact b_h^i value for each $\mathbf{x}_i$, $1 \leqslant i \leqslant n$ is unnecessary. We will first compute an upper bound $\hat{b}_h^i$ of b_h^i. If $\hat{b}_h^i < \max_{i=1}^n b_h^i$, it is concluded that $\mathbf{x}_i$ is impossible to be the new initial center; otherwise, we will compute the exact b_h^i.

We will introduce how to take advantage of the local geometrical information of objects to build the acceleration mechanism. We first pre-process the given data set U into k' subsets such that data objects close to each other in space are likely to be placed in the same subset, where $n \gg k' > k$ (Fig. 1). The formal description is as follows: Let $\mathbf{S} = \{S_1, S_2, \ldots, S_{k'}\}$, where $S_l \subset U$, $S_l \cap S_q = \varnothing$, and $\bigcup_{l=1}^{k'} S_l = U$, for $1 \leqslant l \neq q \leqslant k'$; $\mathbf{C} = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_{k'}\}$, where $\mathbf{c}_l = \sum_{\mathbf{x} \in S_l} \mathbf{x} / |S_l|$ is the center of S_l; $\mathbf{R} = \{r_1, r_2, \ldots, r_{k'}\}$, where $r_l = \max_{\mathbf{x} \in S_l} \|\mathbf{x} - \mathbf{c}_l\|^2$ is the radius of S_l for $1 \leqslant i \leqslant k'$.

The pre-processing is required to not produce such clusters in which most of objects are from the same class but quickly obtain relatively uniform clusters. Since the k-means algorithm has an approximate linear time complexity for the number of objects and tends to partition the data set into clusters with relatively uniform sizes [8,26], we suggest applying the algorithm with k' randomly selected initial centers to quickly produce k' small clusters.

After obtaining the partition $\mathbf{S}$, we can rewrite $b_h^i = \sum_{l=1}^{k'} \psi_h^l(\mathbf{x}_i)$ where $\psi_h^l(\mathbf{x}_i) = \sum_{\mathbf{x}_j \in S_l} \max\{0, d_{h-1}^j - \|\mathbf{x}_i - \mathbf{x}_j\|^2\}$, and $P_h(\mathbf{x}_i) = \bigcup_{l=1}^{k'} B_h^l(\mathbf{x}_i)$ where $B_h^l(\mathbf{x}_i) = \{\mathbf{x}_j | \|\mathbf{x}_i - \mathbf{x}_j\|^2 \leqslant d_{h-1}^j, \mathbf{x}_j \in S_l\}$ (Fig. 2). When determining whether the objects in S_l belong to $B_h^l(\mathbf{x}_i)$, we do not directly compute the distance between $\mathbf{x}_i$ and each $\mathbf{x}_j$ in S_l. Considering

$$\left| \|\mathbf{x}_i - \mathbf{c}_l\|^2 - \|\mathbf{x}_j - \mathbf{c}_l\|^2 \right| \leqslant \|\mathbf{x}_i - \mathbf{x}_j\|^2 \leqslant \|\mathbf{x}_i - \mathbf{c}_l\|^2 + \|\mathbf{x}_j - \mathbf{c}_l\|^2 \tag{6}$$

for each $\mathbf{x}_j$ in S_l, we use $\|\mathbf{x}_i - \mathbf{c}_l\|^2$ and $\|\mathbf{x}_j - \mathbf{c}_l\|^2$ to estimate $\|\mathbf{x}_i - \mathbf{x}_j\|^2$ and obtain the following inequality rules to reduce several unnecessary operations when constructing $P_h(\mathbf{x}_i)$:

(a) If $|\|\mathbf{x}_i - \mathbf{c}_l\|^2 - \|\mathbf{x}_j - \mathbf{c}_l\|^2| \geqslant d_{h-1}^j$ and $\mathbf{x}_j \in S_l$, the object $\mathbf{x}_j$ does not belong to $B_h^l(\mathbf{x}_i)$.
(b) If $\|\mathbf{x}_i - \mathbf{c}_l\|^2 + \|\mathbf{x}_j - \mathbf{c}_l\|^2 \leqslant d_{h-1}^j$ and $\mathbf{x}_j \in S_l$, the object $\mathbf{x}_j$ belongs to $B_h^l(\mathbf{x}_i)$.

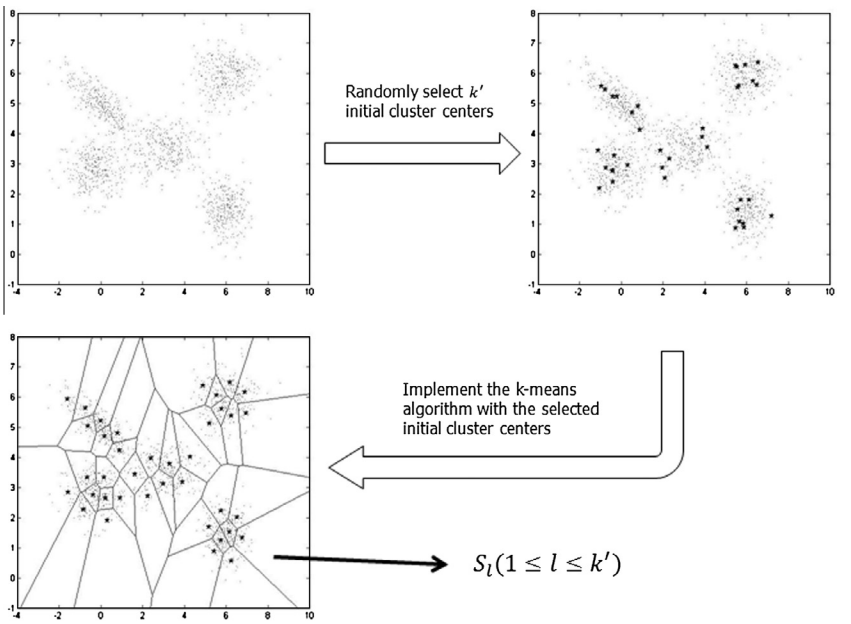

Fig. 1. Pre-processing of a data set.

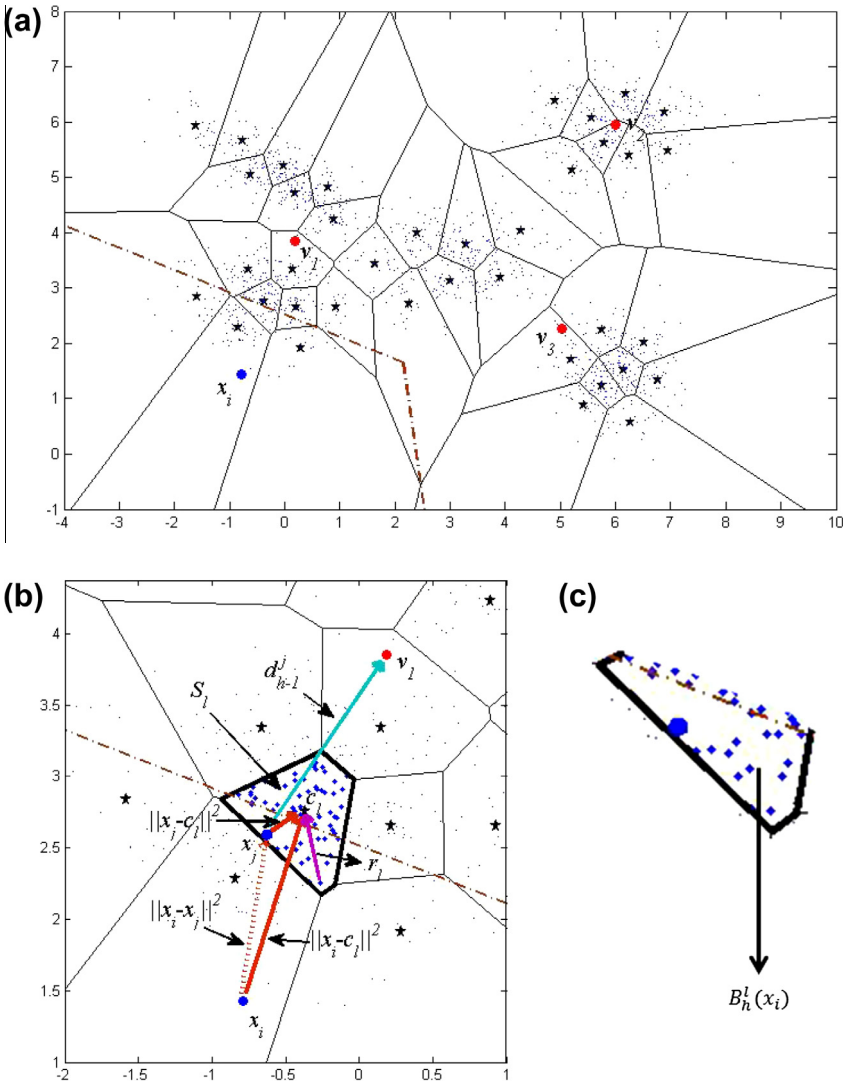

Fig. 2. (a) Object space that $\mathbf{x}_i$ can represent when $\mathbf{x}_i$ is the new cluster center. (b) $d^j_{h-1}, r_l, \|\mathbf{x}_i - \mathbf{c}_l\|^2, \|\mathbf{x}_j - \mathbf{c}_l\|^2$ and $\|\mathbf{x}_i - \mathbf{x}_j\|^2$ when giving S_l,$\mathbf{x}_i$, and $\mathbf{x}_j$. (c) $B^l_h(\mathbf{x}_i)$ in S_l.

(c) If $|\|\mathbf{x}_i - \mathbf{c}_l\|^2 - r_l| \geqslant \max_{\mathbf{x}_j \in S_l} d^j_{h-1}$, all the objects in S_l do not belong to $B^l_h(\mathbf{x}_i)$.

(d) If $\|\mathbf{x}_i + \mathbf{c}_l\|^2 + r_l \leqslant \min_{\mathbf{x}_j \in S_l} d^j_{h-1}$, all the objects in S_l belong to $B^l_h(\mathbf{x}_i)$.

Similar to rough set theory [21,24,25], we use these rules to build the upper and lower approximations for each $B^l_h(\mathbf{x}_i)$, namely, $\overline{B^l_h}(\mathbf{x}_i)$ and $\underline{B^l_h}(\mathbf{x}_i)$, which are described as follows:

$$\overline{B^l_h}(\mathbf{x}_i) = \left\{\mathbf{x}_j | |\|\mathbf{x}_i - \mathbf{c}_l\|^2 - \|\mathbf{x}_j - \mathbf{c}_l\|^2| \leqslant d^j_{h-1}, \mathbf{x}_j \in S_l\right\} \tag{7}$$

and

$$\underline{B^l_h}(\mathbf{x}_i) = \{\mathbf{x}_j |\ \|\mathbf{x}_i - \mathbf{c}_l\|^2 + \|\mathbf{x}_j - \mathbf{c}_l\|^2 \leqslant d^j_{h-1}, \mathbf{x}_j \in S_l\}. \tag{8}$$

We use the two sets to describe approximately $B^l_h(\mathbf{x}_i)$ (Fig. 3). $\underline{B^l_h}(\mathbf{x}_i)$ denotes a set including the objects that belong to $B^l_h(\mathbf{x}_i)$. $\overline{B^l_h}(\mathbf{x}_i)$ denotes a set including the objects that may belong to $B^l_h(\mathbf{x}_i)$. $S_l - \overline{B^l_h}(\mathbf{x}_i)$ denotes a set including the objects that do not belong to $B^l_h(\mathbf{x}_i)$. These sets have the following relation:

$$\underline{B^l_h}(\mathbf{x}_i) \subseteq B^l_h(\mathbf{x}_i) \subseteq \overline{B^l_h}(\mathbf{x}_i). \tag{9}$$

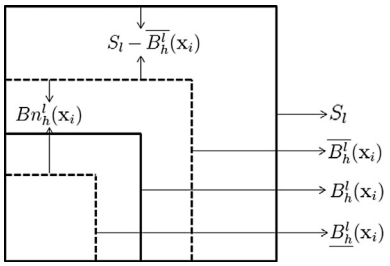

Fig. 3. Approximate description of $B_h^l(\mathbf{x}_i)$.

The boundary of $B_h^l(\mathbf{x}_i)$ is given as follows:

$$Bn_h^l(\mathbf{x}_i) = \overline{B_h^l}(\mathbf{x}_i) - \underline{B_h^l}(\mathbf{x}_i). \tag{10}$$

For each $\psi_h^l(\mathbf{x}_i)$, we obtain the following relation:

$$\begin{aligned}
\psi_h^l(\mathbf{x}_i) = & \sum_{\mathbf{x}_j \in \overline{B_h^l}(\mathbf{x}_i)} \max\left\{0, d_{h-1}^j - \|\mathbf{x}_i - \mathbf{x}_j\|^2\right\} \\
= & \sum_{\mathbf{x}_j \in \underline{B_h^l}(\mathbf{x}_i)} \left(d_{h-1}^j - \|\mathbf{x}_i - \mathbf{x}_j\|^2\right) + \sum_{\mathbf{x}_j \in Bn_h^l(\mathbf{x}_i)} \max\{0, d_{h-1}^j - \|\mathbf{x}_i - \mathbf{x}_j\|^2\} \\
= & \sum_{\mathbf{x}_j \in \underline{B_h^l}(\mathbf{x}_i)} d_{h-1}^j - \sum_{\mathbf{x}_j \in \underline{B_h^l}(\mathbf{x}_i)} \|\mathbf{x}_i - \mathbf{x}_j\|^2 + \sum_{\mathbf{x}_j \in Bn_h^l(\mathbf{x}_i)} \max\left\{0, d_{h-1}^j - \|\mathbf{x}_i - \mathbf{x}_j\|^2\right\} \\
= & \sum_{\mathbf{x}_j \in \underline{B_h^l}(\mathbf{x}_i)} d_{h-1}^j - \sum_{\mathbf{x}_j \in \underline{B_h^l}(\mathbf{x}_i)} |\underline{B_h^l}(\mathbf{x}_i)| \|\mathbf{x}_i\|^2 - \sum_{\mathbf{x}_j \in \underline{B_h^l}(\mathbf{x}_i)} \|\mathbf{x}_j\|^2 + 2\mathbf{x}_i \sum_{\mathbf{x}_j \in \underline{B_h^l}(\mathbf{x}_i)} \mathbf{x}_j \\
+ & \sum_{\mathbf{x}_j \in Bn_h^l(\mathbf{x}_i)} \max\left\{0, d_{h-1}^j - \|\mathbf{x}_i - \mathbf{x}_j\|^2\right\} \\
= & \sum_{\mathbf{x}_j \in \underline{B_h^l}(\mathbf{x}_i)} d_{h-1}^j - |\underline{B_h^l}(\mathbf{x}_i)| \left(\|\mathbf{x}_i - \bar{\mathbf{x}}_l\|^2 - \|\bar{\mathbf{x}}_l\|^2\right) - \sum_{\mathbf{x}_j \in \underline{B_h^l}(\mathbf{x}_i)} \|\mathbf{x}_j\|^2 \\
+ & \sum_{\mathbf{x}_j \in Bn_h^l(\mathbf{x}_i)} \max\left\{0, d_{h-1}^j - \|\mathbf{x}_i - \mathbf{x}_j\|^2\right\}
\end{aligned} \tag{11}$$

where $\bar{\mathbf{x}}_l = \frac{\sum_{\mathbf{x}_j \in \underline{B_h^l}(\mathbf{x}_i)} \mathbf{x}_j}{|\underline{B_h^l}(\mathbf{x}_i)|}$ is the mean of the objects in $\underline{B_h^l}(\mathbf{x}_i)$. By computing $\psi_h^l(\mathbf{x}_i)$ according to (11), we can derive the following observations: (1) $|S_l| - |\overline{B_h^l}(\mathbf{x}_i)|$ objects in S_l can be directly rejected to construct $B_h^l(\mathbf{x}_i)$; (2) if we save $\|\mathbf{x}_i\|^2$ for each object $\mathbf{x}_i \in U$ before the FGKM algorithm is implemented, which only needs $O(n)$ spaces, we only need to compute $\|\bar{\mathbf{x}}_l\|^2$ and $\|\mathbf{x}_i - \bar{\mathbf{x}}_l\|^2$, instead of computing $\|\mathbf{x}_i - \mathbf{x}_j\|^2$ for each $\mathbf{x}_j \in \underline{B_h^l}(\mathbf{x}_i)$; (3) $Bn_h^l(\mathbf{x}_i)$ is an uncertain region. We need to compute $\|\mathbf{x}_i - \mathbf{x}_j\|^2$ for each $\mathbf{x}_j$ in $Bn_h^l(\mathbf{x}_i)$ to determine whether $\mathbf{x}_j$ belongs to $B_h^l(\mathbf{x}_i)$, thus indicating that a smaller $|Bn_h^l(\mathbf{x}_i)|$ value corresponds to a lower number of distances that are unnecessarily computed. When $Bn_h^l(\mathbf{x}_i)$ is empty, cases where all objects in S_l are certain are identified:

(1) If rule (c) is satisfied, then $\overline{B_h^l}(\mathbf{x}_i) = \emptyset$. In this case, we can directly reject all the objects in S_l and set $\psi_h^l(\mathbf{x}_i) = 0$.
(2) If rule (d) is satisfied, then $\underline{B_h^l}(\mathbf{x}_i) = S_l$. In this case, we can directly compute

$$\psi_h^l(\mathbf{x}_i) = \sum_{\mathbf{x}_j \in S_l} d_{h-1}^j - |S_l| \left(\|\mathbf{x}_i - \mathbf{c}_l\|^2 - \|\mathbf{c}_l\|^2\right) - \sum_{\mathbf{x}_j \in S_l} \|\mathbf{x}_j\|^2. \tag{12}$$

On the basis of the above analyses, we can reduce lots of unnecessary distance calculations by $\underline{B_h^l}(\mathbf{x}_i)$ and $\overline{B_h^l}(\mathbf{x}_i)$ for $1 \leqslant l \leqslant k'$ while computing the exact b_h^i.

We will further reduce the computational complexity. According to Step 4 of the FGKM algorithm, only the object $\mathbf{x}_i$ with the maximum b_h^i should be identified, indicating that computing the exact b_h^i value for each $\mathbf{x}_i$, $1 \leqslant i \leqslant n$, is unnecessary. Therefore, we will first use the approximate description of $B_h^l(\mathbf{x}_i)$ by $\underline{B_h^l}(\mathbf{x}_i)$ and $\overline{B_h^l}(\mathbf{x}_i)$ to calculate the upper bound $\hat{b}_h^i$ of b_h^i for $1 \leqslant i \leqslant n$, which is defined as follows:

$$\hat{b}_h^i = \sum_{l=1}^{k'} \hat{\psi}_h^l(\mathbf{x}_i), \tag{13}$$

where

$$\hat{\psi}_h^l(\mathbf{x}_i) = \begin{cases} 0 & \text{if } \overline{B_h^l}(\mathbf{x}_i) = \emptyset, \\ |S_l|\left(\|\mathbf{x}_i - \mathbf{c}_l\|^2 - \|\mathbf{c}_l\|^2\right) + \sum_{\mathbf{x}_j \in S_l} \|\mathbf{x}_j\|^2, & \text{if } \underline{B_h^l}(\mathbf{x}_i) = S_l, \\ \sum_{\mathbf{x}_j \in \overline{B_h^l}(\mathbf{x}_i)} \left(d_{h-1}^j - \left|\|\mathbf{x}_i - \mathbf{c}_l\|^2 - \|\mathbf{x}_j - \mathbf{c}_l\|^2\right|\right), & \text{otherwise}. \end{cases} \tag{14}$$

Given that $\psi_h^l(\mathbf{x}_i) \leqslant \hat{\psi}_h^l(\mathbf{x}_i)$ for $1 \leqslant l \leqslant k'$, we have $b_h^i \leqslant \hat{b}_h^i$.

After obtaining the upper bound $\hat{b}_h^i$, if $\hat{b}_h^i < \varepsilon \leqslant \max_{i=1}^n b_h^i$, we can easily conclude that $\mathbf{x}_i$ cannot be the new initial cluster center. This conclusion can further reduce the amount of distance calculations. Here, ε is a parameter set to the maximum value of all obtained exact b_h^i if they exist; otherwise, is set to zero.

The new acceleration mechanism is shown in Table 1, which describes how to use the above approximate description to obtain rapidly the hth initial cluster center, for $1 < h \leqslant k$. We use the mechanism to expedite the clustering procedure of the FGKM algorithm. The accelerated FGKM algorithm is called FGKM+A which is described as follows:

Table 1
An acceleration mechanism for producing the hth initial cluster center.

```
Set ε = 0;
for each data object x_i in U
  for each subset S_l in S
    set the upper bound of B_h^l = the lower bound of B_h^l = ∅;
    if rule (c) is satisfied
      set ψ̂_h^l(x_i) = ψ_h^l(x_i) = 0;
      continue;
    end if
    if rule (d) is satisfied
      compute ψ_h^l(x_i), according to Eq. (12);
      set ψ̂_h^l(x_i) = ψ_h^l(x_i) and the upper bound of B_h^l = the lower bound of B_h^l = S_l;
    else
      for each data object x_j in S_l
        if |||x_i − c_l||^2 − ||x_j − c_l||^2| ⩽ d_j
          the upper bound of B_h^l = the upper bound of B_h^l ∪ {x_j};
        end if
        if ||x_i − c_l||^2 + ||x_j − c_l||^2 ⩽ d_j
          the lower bound of B_h^l = the lower bound of B_h^l ∪ {x_j};
        end if
      end for
      compute ψ̂_h^l(x_i) according to Eq. (14);
    end if
  end for
  compute b̂_h^i = Σ_{l=1}^{k'} ψ̂_h^l(x_i);
  if b̂_h^i ⩽ ε
    set b_h^i = 0;
    continue;
  else
    for l = 1 to k'
      if |the upper bound of B_h^l| − |the lower bound of B_h^l| ≠ 0
        compute ψ_h^l(x_i), according to Eq. (11);
      end if
    end for
    compute b_h^i = Σ_{l=1}^{k'} ψ_h^l(x_i);
    if ε < b_h^i
      set ε = b_h^i and q = i;
    end if
  end if
end for
output the object x_q;
```

The FGKM+A algorithm

Table 2
Data sets from UCI.

Data set	Objects	Attributes
Handwritten digits	5620	64
Statlog	6435	36
Musk	6598	168
Isolet	7797	617
Coil	9000	86
Letters	20,000	16
Shuttle	58,000	9
Corel image	68,040	89

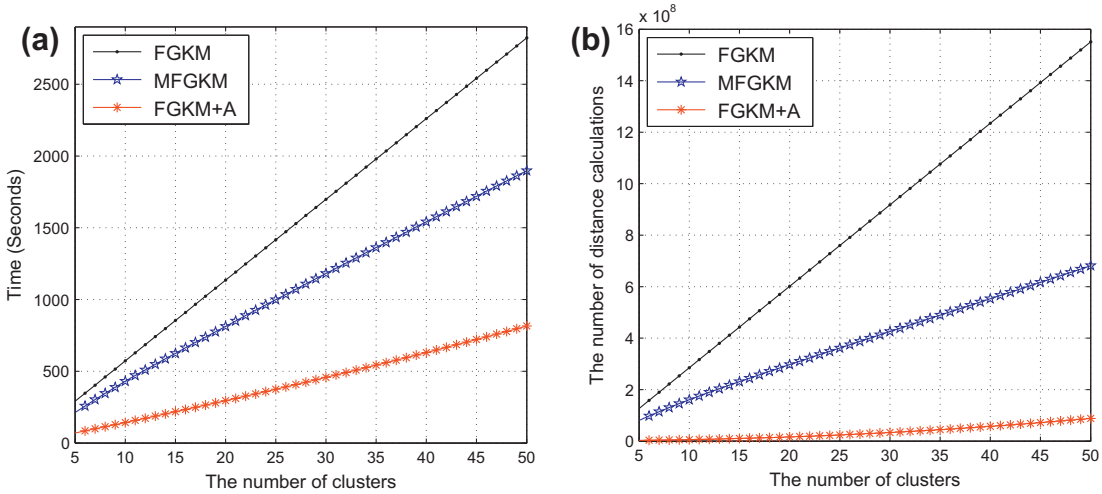

Fig. 4. (a) Computational times for different numbers of clusters on the handwritten digits data. (b) Numbers of distance calculations for different numbers of clusters on the handwritten digits data.

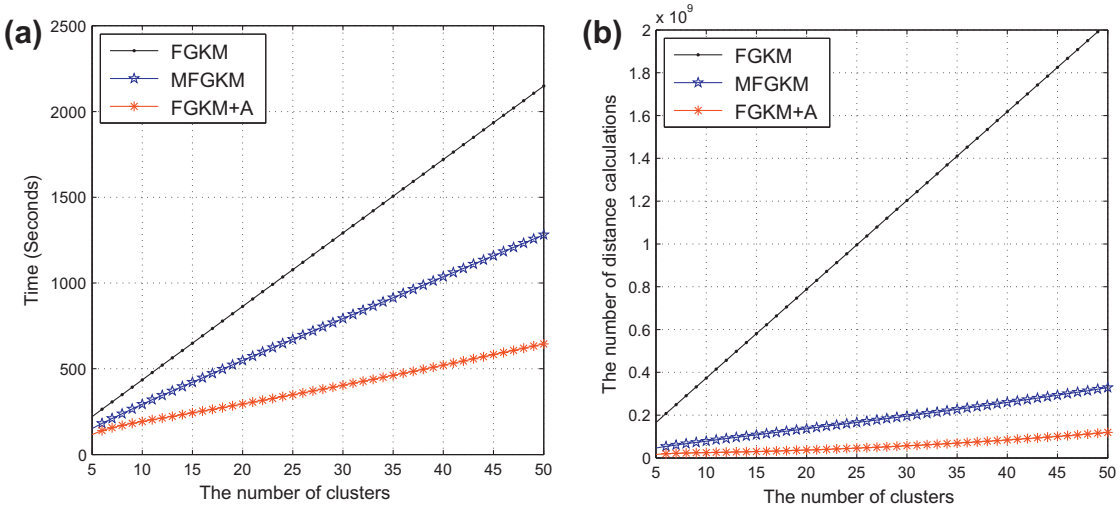

Fig. 5. (a) Computational times for different numbers of clusters on the statlog data. (b) Numbers of distance calculations for different numbers of clusters on the statlog data.

Step 1. Randomly select k' initial cluster centers and apply the k-means algorithm to partition the data set into k' subsets, i.e., $\mathbf{S} = \{S_1, S_2, \ldots, S_{k'}\}$. Furthermore, save the center of each subset and the distances between each object and these centers.

Step 2. Compute $V_1^* = \{\mathbf{v}_1\}$ from the data set X, where $\mathbf{v}_1 = \sum_{i=1}^{n} \mathbf{x}_i / n$ and n is the number of objects in U. Set $h = 1$.

Step 3. Set $h = h + 1$. If $h > k$, then stop.

Step 4. Use the acceleration mechanism in Table 1 to select the object $\mathbf{x}_q$ from U as the hth initial cluster center. Set $V = V_{h-1}^* \cup \{\mathbf{x}_q\}$.

Step 5. Apply the k-means algorithm with the initial set of cluster centers V, save the resulting set of cluster centers into V_h^* and compute d_h^j for each $\mathbf{x}_j \in U$. Goto Step 2.

4. Space and time complexity

4.1. Space complexity

In the FGKM+A algorithm, we need to save a partition vector $Pc = [p_1, p_2, \ldots, p_n]$, where $p_i = l$ if the object $\mathbf{x}_i$ belongs to S_l; the distance matrix $D = [D_{il}]$ which is a $n \times k'$ real matrix; $D_{il} = \|\mathbf{x}_i - \mathbf{c}_l\|^2$ for $1 \leqslant i \leqslant n, 1 \leqslant l \leqslant k'$ and $\|\mathbf{x}_j\|^2$ for each object $\mathbf{x}_i$, $1 \leqslant i \leqslant n$. The above procedure requires $O(n(k'+2))$ space. Given that $k' \ll n$, $n(k'+2) \ll n^2$.

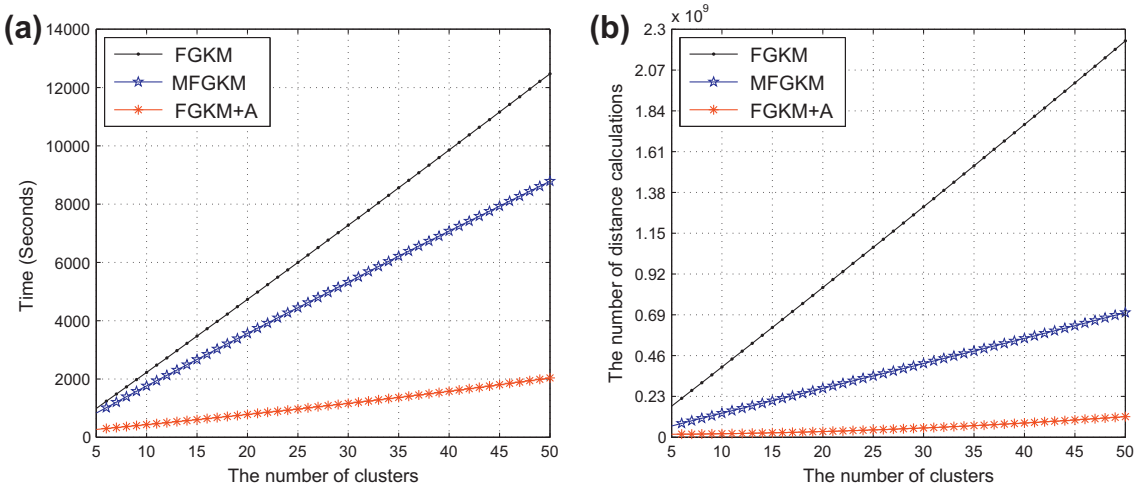

Fig. 6. (a) Computational times for different numbers of clusters on the musk data. (b) Numbers of distance calculations for different numbers of clusters on the musk data.

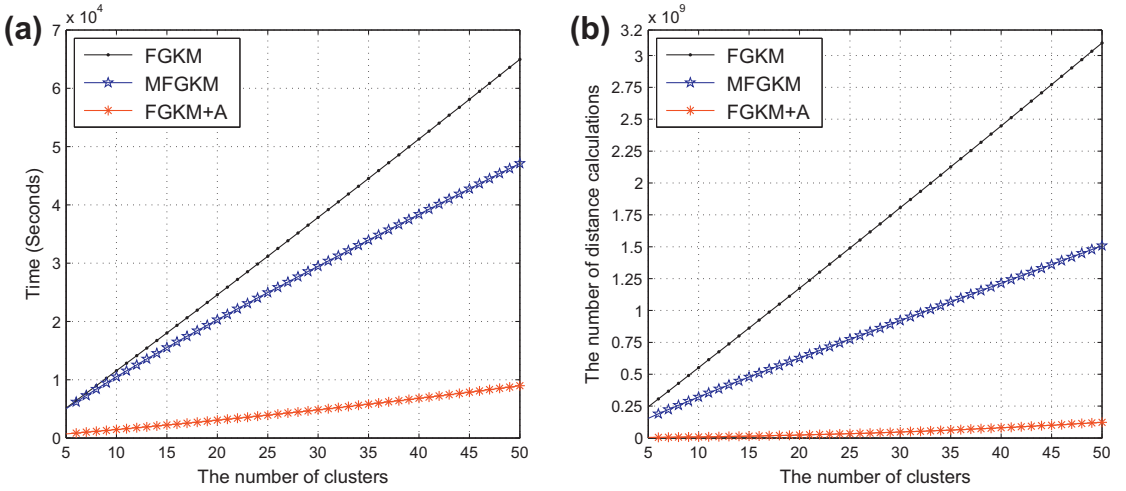

Fig. 7. (a) Computational times for different numbers of clusters on the isolet data. (b) Numbers of distance calculations for different numbers of clusters on the isolet data.

4.2. Time complexity

In Step 1, we apply the k-means algorithm to partition the data set into k' subsets, which needs $O(nk't)$ distance calculations, where t is the number of iterations. In the procedure, the center $\mathbf{c}_l$ of each subset $S_l(1 \leqslant l \leqslant k')$ and the distances between each object $\mathbf{x}_i(1 \leqslant i \leqslant n)$ and all the centers can be obtained. Furthermore, we need to calculate $\|\mathbf{x}_j\|^2$ for each object $\mathbf{x}_i(1 \leqslant i \leqslant n)$, which is $O(n)$ operations. In Step 4, we need $O(n_1 n_2)$ distance calculations to obtain the hth initial cluster center, where $n_1(\ll n)$ and $n_2(\ll n)$ are the numbers of objects and distance calculations required to obtain the exact b_h^i, respectively. To generate k cluster centers, the proposed algorithm needs $O(nk't + n + n_1 n_2 k + nk^2 t)$ distance calculations. Given that the computational complexity of FGKM is $O(n^2 k + nk^2 t)$ in terms of the number of distance calculations, we may conclude that the proposed algorithm has less computational complexity.

5. Experimental results

To verify the efficiency of the proposed algorithm, experiments with eight standard data sets are conducted on an Intel Q9400 computer with 2G RAM. These data sets (Table 2) are downloaded from the UCI Machine Learning Repository [28].

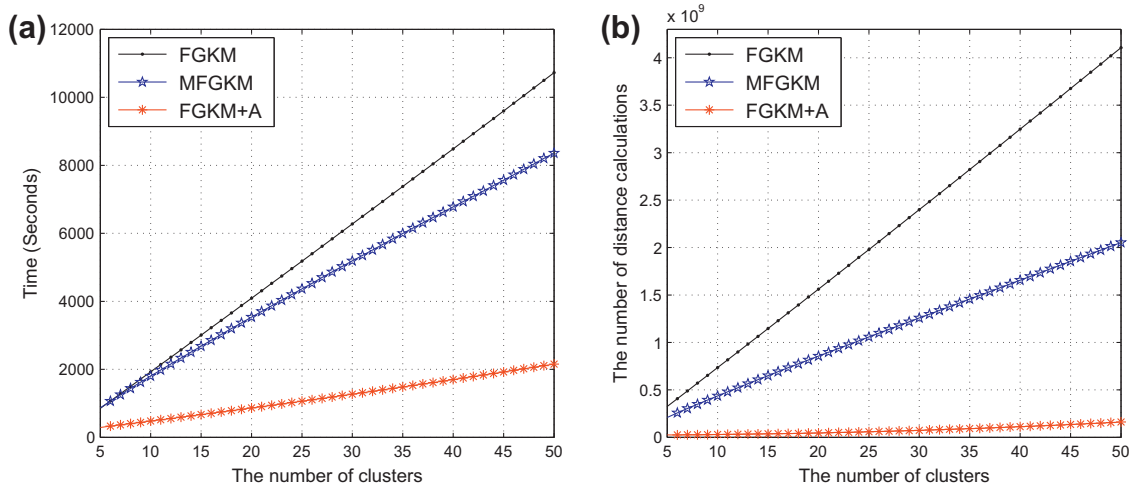

Fig. 8. (a) Computational times for different numbers of clusters on the coil data. (b) Numbers of distance calculations for different numbers of clusters on the coil data.

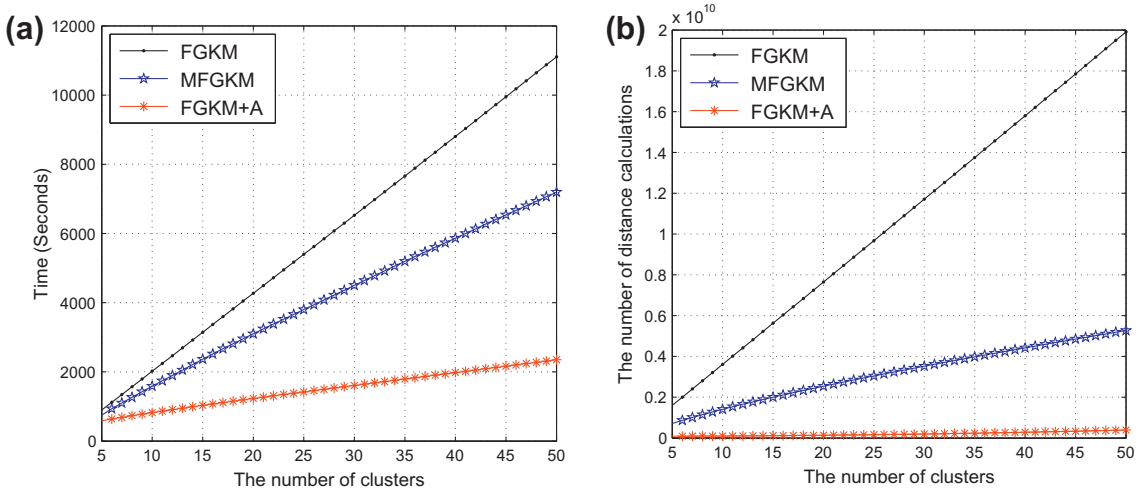

Fig. 9. (a) Computational times for different numbers of clusters on the letters data. (b) Numbers of distance calculations for different numbers of clusters on the letters data.

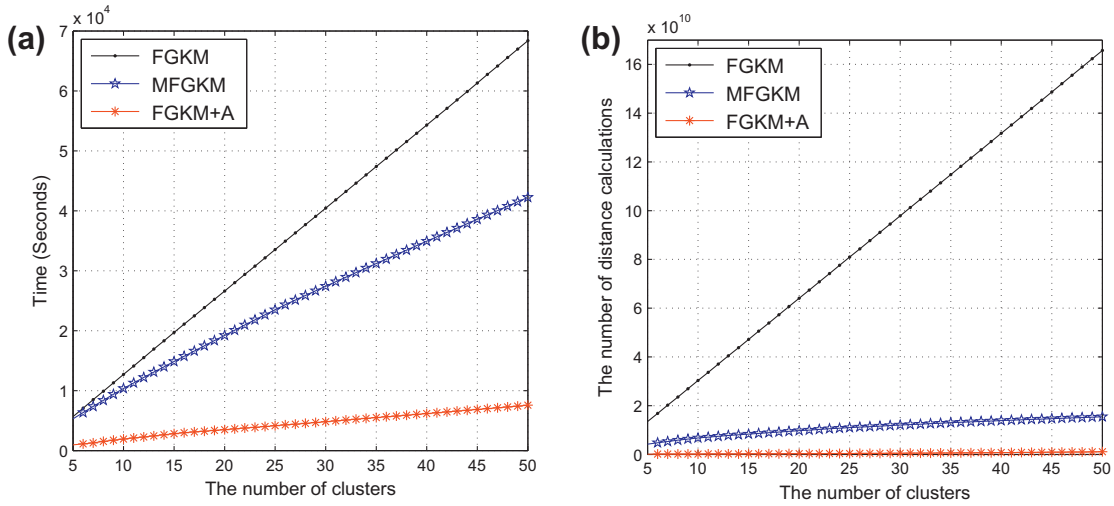

Fig. 10. (a) Computational times for different numbers of clusters on the shuttle data. (b) Numbers of distance calculations for different numbers of clusters on the shuttle data.

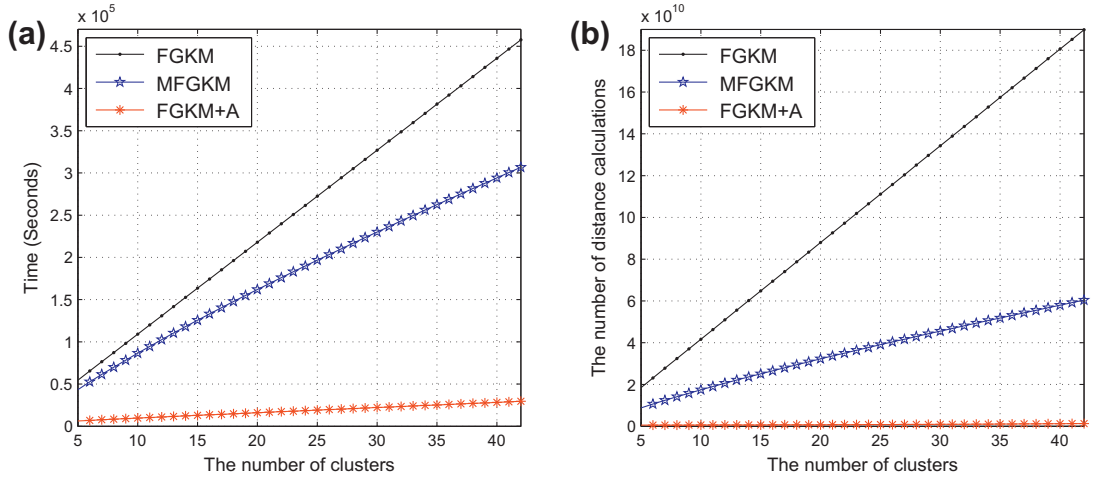

Fig. 11. (a) Computational times for different numbers of clusters on the corel image data. (b) Numbers of distance calculations for different numbers of clusters on the corel image data.

We compare the FGKM+A algorithm with the FGKM algorithm proposed by Likas et al. [22] and the MFGKM algorithm proposed by Lai and Huang [18] in terms of computing time and number of distance calculations. In the following experiments, we set $k' = \lfloor \sqrt{n} \rfloor$ for FGKM+A.

Figs. 4–11 show the total execution time and distance calculations of these algorithms on the eight data sets with the different numbers of clusters, respectively. These data sets include the handwritten digits, statlog, musk, isolet, coil, letters, shuttle and corel image data sets which have different sizes. On each of these provided data sets, the FGKM+A algorithm outperforms the FGKM and MFGKM algorithms in terms of computing time and distance calculations. When the number of clusters k increases, the efficiency of the proposed algorithm becomes more remarkable than the FGKM and MFGKM algorithms.

Furthermore, we test the scalability with the different numbers of dimensions on the two data sets, namely, the musk and isolet data sets. We fix the numbers of clusters k to be as 20. Figs. 12 and 13 show that the FGKM+A algorithm exhibits better scalability with increasing dimensions, compared with the FGKM and MFGKM algorithms.

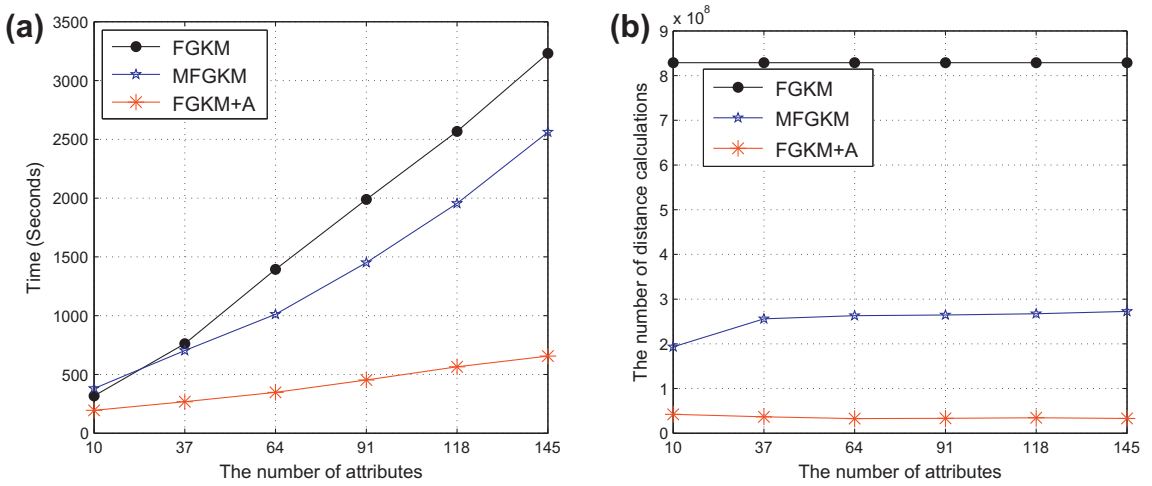

Fig. 12. (a) Computational times for different numbers of attributes on the musk data. (b) Numbers of distance calculations for different numbers of attributes on the musk data.

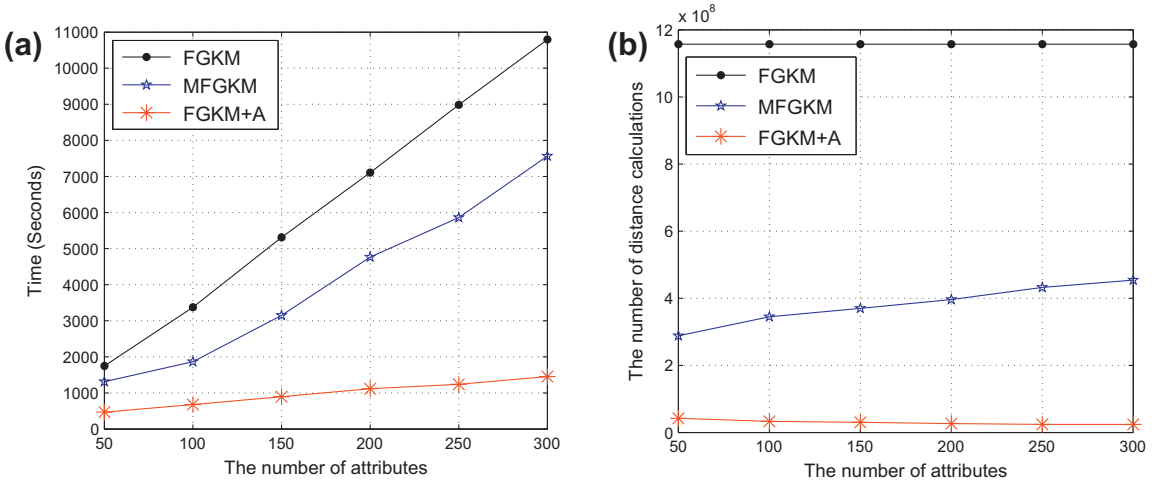

Fig. 13. (a) Computational times for different numbers of attributes on the isolet data. (b) Numbers of distance calculations for different numbers of attributes on the isolet data.

6. Conclusions

To improve the efficiency of the FGKM clustering algorithm, an acceleration mechanism has been developed in this paper by using the local geometrical information of data objects. In the development, an approximate description of an object set has been proposed to help users reduce the computational complexity of determining new cluster centers. Compared with the FGKM and MFGKM algorithms, the accelerated FGKM algorithm, i.e., FGKM+A, requires less computing time and fewer distance calculations while retaining the same clustering results. The performance of the proposed algorithm is more remarkable as the number of dimensions or clusters of a data set increases.

Acknowledgements

The authors are very grateful to the editors and reviewers for their valuable comments and suggestions. This work was supported by the National Natural Science Foundation of China (No. 71031006), the Foundation of Doctoral Program Research of Ministry of Education of China (No. 20101401110002), the National Key Basic Research and Development Program of China (973) (No. 2013CB329404).

References

[1] A. Ahmadi, F. Karray, M.S. Kamel, Model order selection for multiple cooperative swarms clustering using stability analysis, Information Sciences 182 (2012) 169–183.
[2] B. Akay, D. Karaboga, A modified artificial bee colony algorithm for real-parameter optimization, Information Sciences 192 (2012) 120–142.
[3] G. Babu, M. Murty, A near-optimal initial seed value selection for k-means algorithm using genetic algorithm, Pattern Recognition Letters 14 (1993) 763–769.
[4] A. Bagirov, Modified global k-means algorithm for sum-of-squares clustering problem, Pattern Recognition 41 (2008) 3192–3199.
[5] A. Bagirov, J. Ugon, D. Webb, Fast modified global k-means algorithm for incremental cluster construction, Pattern Recognition 44 (2011) 866–876.
[6] Z. Che, A. Unler, Clustering and selecting suppliers based on simulated annealing algorithms, Computers and Mathematics with Applications 63 (1) (2012) 228–238.
[7] W. Chen, Y. Song, H. Bai, C. Lin, E. Chang, Parallel spectral clustering in distributed systems, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (3) (2011) 568–586.
[8] R. Duda, P. Hart, Pattern Classification and Scene Analysis, Wiley, New York, 1973.
[9] R. Forsatia, M. Mahdavib, M. Shamsfarda, M.R. Meybodi, Efficient stochastic algorithms for document clustering, Information Sciences 220 (2013) 269–291.
[10] A. Ghoting, S. Parthasarathy, M. Otey, Fast mining of distance-based outliers in high-dimensional datasets, Data Mining and Knowledge Discovery 16 (2008) 349–364.
[11] Z. Gungr, A. Unler, K-harmonic means data clustering with simulated annealing heuristic, Applied Mathematics and Computation 184 (2007) 199–209.
[12] J. Han, M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufman, 2001.
[13] A. Hatamlou, Black hole: A new heuristic optimization approach for data clustering, Information Sciences 222 (2013) 175–184.
[14] A. Jain, R. Dubes, Algorithms for Clustering Data, Prentice Hall, 1988.
[15] K. Krishna, M. Murty, Genetic k-means algorithm, IEEE Transactions on Systems, Man, and Cybernetics 29 (3) (1999) 433–439.
[16] R. Kuo, Y. Syu, Z. Chen, F. Tien, Integration of particle swarm optimization and genetic algorithm for dynamic clustering, Information Sciences 195 (2012) 124–140.
[17] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, Berkeley, 1967, pp. 281–297.
[18] J. Lai, J. Huang, Fast global k-means clustering using cluster membership and inequality, Pattern Recognition 43 (2010) 1954–1963.
[19] M. Laszlo, S. Mukherjee, A genetic algorithm using hyper-quadtrees for low-dimensional k-means clustering, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (4) (2006) 533–543.
[20] J. Li, M. Ng, Y. Cheung, Z. Huang, Agglomerative fuzzy k-Means clustering algorithm with selection of number of clusters, IEEE Transactions on Knowledge and Data Engineering 20 (11) (2008) 1519–1534.
[21] J. Liang, J. Wang, Y. Qian, A new measure of uncertainty based on knowledge granulation for rough sets, Information Sciences 17 (9) (2009) 458–470.
[22] A. Likas, M. Vlassis, J. Verbeek, The global k-means clustering algorithm, Pattern Recognition 35 (2) (2003) 451–461.
[23] R. Liu, L. Jiao, X. Zhang, Y. Li, Gene transposon based clone selection algorithm for automatic clustering, Information Sciences 204 (2012) 1–22.
[24] Z. Pawlak, Rough Sets-Theoretical Aspects of Reasoning about Data, Dordrecht, Boston, Kluwer Academic Publishers., London, 1991.
[25] Y. Qian, J. Liang, Y. Yao, MGRS: a mulit-granulation rough set, Information Sciences 180 (2010) 949–970.
[26] H. Xiong, J. Wu, J. Chen, K-means clustering versus validation measures: a data distribution perspective, Journal of Information Retrieval 1 (2004) 67–88.
[27] L. Zhang, Q. Cao, A novel ant-based clustering algorithm using the kernel method, Information Sciences 181 (2011) 4658–4672.
[28] UCI, UCI Machine Learning Repository, 2011. <http://www.ics.uci.edu/mlearn/MLRepository.html>.