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

A R T I C L E   I N F O

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

A B S T R A C T

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
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 based on partial optimization for \( k \)-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 \( U = \{x_1, x_2, \ldots, x_n\} \) be a set of \( n \) objects. Object \( x_i = \{x_{i1}, x_{i2}, \ldots, x_{im}\} \) 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 \( V \):

\[
F(W, V) = \sum_{l=1}^{k} \sum_{i=1}^{n} w_{li} ||x_i - v_l||^2,
\]

subject to

\[
w_{li} \in \{0, 1\}, \sum_{l=1}^{k} w_{li} = 1.0 < \sum_{i=1}^{n} w_{li} < n, \quad 1 \leq l \leq k, \quad 1 \leq i \leq n.
\]

where

- \( W = [w_{ij}] \) is a \( k \)-by-\( n \) \( \{0, 1\} \) matrix, \( w_{ij} \) is a binary variable, and indicates whether object \( x_i \) belongs to the \( j \)th cluster, \( w_{ij} = 1 \) if \( x_i \) belongs to the \( j \)th cluster and 0 otherwise;
- \( V = [v_1, v_2, \ldots, v_k] \) and \( v_{lj} = [v_{l1}, v_{l2}, \ldots, v_{lm}] \) is the \( j \)th cluster center with \( m \) attributes;
- \( ||x_i - v_l||^2 = \sum_{j=1}^{m} (x_{ij} - v_{lj})^2 \) is Euclidean distance between the object \( x_i \) and the \( j \)th cluster center \( v_l \).

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

Problem \( P_1 \) is solved by

\[
w_{li} = \begin{cases} 1, & \text{if } ||x_i - \tilde{v}_l||^2 \leq ||x_i - \tilde{v}_h||^2, 1 \leq h \leq k, \\ 0, & \text{otherwise}. \end{cases}
\]

for \( 1 \leq i \leq n, 1 \leq l \leq k \).

Problem \( P_2 \) is solved by
null
3. The FGKM*A algorithm

In the FGKM algorithm, computing $b_h$ for each $x_i$ in $U$ is an important step, given that $b_h$ 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, $b_h = \sum_{x_i \in P_h(x_i)} \left( d_{h_i} - ||x_i - x_{h_i}||^2 \right)$, where $P_h(x_i) = \{ x_j ||x_i - x_j||^2 \leq d_{h_{i-1}}, x_j \in U \}$, for $1 \leq i \leq n$. The FGKM algorithm needs to identify a set $P_h(x_i)$ from $U$ by calculating the distance between each object $x_i$ in $U$ and $x_i$. 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 $||x_j - x_i||^2$ for each object $x_j \in U$, we will compute its estimated value and determine whether $x_j$ belongs to $P_h(x_i)$.
- Given that the FGKM algorithm requires only the determination of the object $x_i$ with the maximum $b_h$, computing the exact $b_h$ value for each $x_i$, $1 \leq i \leq n$ is unnecessary. We will first compute an upper bound $b_h$ of $b_h$. If $b_h < \max_{i=1}^{n} b_h$, it is concluded that $x_i$ is impossible to be the new initial center; otherwise, we will compute the exact $b_h$.

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 $S = \{ S_1, S_2, \ldots, S_{k'} \}$, where $S_1 \subset U$, $S_i \cap S_q = \emptyset$, and $\bigcup_{i=1}^{k'} S_i = U$, for $1 \leq l \neq q \leq k'$; $C = \{ c_1, c_2, \ldots, c_{k'} \}$, where $c_i = \sum_{x \in S_i} x ||S_i||$ is the center of $S_i$; $R = \{ r_1, r_2, \ldots, r_{k'} \}$, where $r_i = \max_{x \in S_i} ||x - c_i||^2$ is the radius of $S_i$ for $1 \leq l \leq k'$.

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 $S$, we can rewrite $b_h = \sum_{i=1}^{k'} \psi_h(x_i)$ where $\psi_h(x_i) = \sum_{x \in S_i} \max \{ 0, d_{h_{i-1}} - ||x - x_i||^2 \}$, and $P_h(x_i) = \bigcup_{l=1}^{k'} B_{h_i}(x_i)$ where $B_{h_i}(x_i) = \{ x_j ||x_i - x_j||^2 \leq d_{h_{i-1}}, x_j \in S_i \}$ (Fig. 2). When determining whether the objects in $S_i$ belong to $B_{h_i}(x_i)$, we do not directly compute the distance between $x_i$ and each $x_j$ in $S_i$. Considering

$$||x_i - c_i||^2 - ||x_j - c_i||^2 \leq ||x_i - x_j||^2 \leq ||x_i - c_i||^2 + ||x_j - c_i||^2$$

(6)

for each $x_i$ in $S_i$, we use $||x_i - c_i||^2$ and $||x_j - c_i||^2$ to estimate $||x_i - x_j||^2$ and obtain the following inequality rules to reduce several unnecessary operations when constructing $P_h(x_i)$:

(a) If $||x_i - c_i||^2 - ||x_j - c_i||^2 \geq d_{h_{i-1}}$ and $x_j \in S_i$, the object $x_j$ does not belong to $B_{h_i}(x_i)$.
(b) If $||x_i - c_i||^2 + ||x_j - c_i||^2 \leq d_{h_{i-1}}$ and $x_j \in S_i$, the object $x_j$ belongs to $B_{h_i}(x_i)$.

Fig. 1. Pre-processing of a data set.
If \( |x_i - c|^2 - r_i | \geq \max d_{k-1}^i \), all the objects in \( S_i \) do not belong to \( B_h^i(x_i) \).

If \( |x_i + c|^2 + r_i \leq \min_{x_j \in S_i} d_{k-1}^i \), all the objects in \( S_i \) belong to \( B_h^i(x_i) \).

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

\[
\overline{B}_h^i(x_i) = \{ x_j | |x_i - c|^2 - |x_j - c|^2 \leq d_{k-1}^i, x_j \in S_i \} \tag{7}
\]

and

\[
\underline{B}_h^i(x_i) = \{ x_j | |x_i - c|^2 + |x_j - c|^2 \leq d_{k-1}^i, x_j \in S_i \} \tag{8}
\]

We use the two sets to describe approximately \( B_h^i(x_i) \) (Fig. 3). \( \overline{B}_h^i(x_i) \) denotes a set including the objects that belong to \( B_h^i(x_i) \). \( \underline{B}_h^i(x_i) \) denotes a set including the objects that may belong to \( B_h^i(x_i) \). \( S_i - \overline{B}_h^i(x_i) \) denotes a set including the objects that do not belong to \( B_h^i(x_i) \). These sets have the following relation:

\[
\underline{B}_h^i(x_i) \subseteq B_h^i(x_i) \subseteq \overline{B}_h^i(x_i) \tag{9}
\]
The boundary of $B^j_k(x_i)$ is given as follows:

$$B^j_k(x_i) = B^j_k(x_i) - B^j_k(x_i).$$

For each $\psi^j_k(x_i)$, we obtain the following relation:

$$\psi^j_k(x_i) = \sum_{x_i \in B^j_k(x_i)} \max \left\{ 0, d_{h-1}^j - \|x_i - x_j\|^2 \right\}$$

$$= \sum_{x_i \in B^j_k(x_i)} \left( d_{h-1}^j - \|x_i - x_j\|^2 \right) + \sum_{x_i \in \overline{B}^j_k(x_i)} \max \left\{ 0, d_{h-1}^j - \|x_i - x_j\|^2 \right\}$$

$$= \sum_{x_i \in B^j_k(x_i)} d_{h-1}^j - \sum_{x_i \in B^j_k(x_i)} \|x_i - x_j\|^2 + \sum_{x_i \in \overline{B}^j_k(x_i)} \max \left\{ 0, d_{h-1}^j - \|x_i - x_j\|^2 \right\}$$

$$= \sum_{x_i \in B^j_k(x_i)} d_{h-1}^j - \sum_{x_i \in B^j_k(x_i)} \|B^j_k(x_i)\||x_j|^2 - \sum_{x_i \in B^j_k(x_i)} \|x_j\|^2 + 2\sum_{x_i \in \overline{B}^j_k(x_i)} x_j$$

$$+ \sum_{x_i \in \overline{B}^j_k(x_i)} \max \left\{ 0, d_{h-1}^j - \|x_i - x_j\|^2 \right\}$$

$$= \sum_{x_i \in B^j_k(x_i)} d_{h-1}^j - \sum_{x_i \in B^j_k(x_i)} \|B^j_k(x_i)\||x_j|^2 - \sum_{x_i \in B^j_k(x_i)} \|x_j\|^2$$

$$+ \sum_{x_i \in \overline{B}^j_k(x_i)} \max \left\{ 0, d_{h-1}^j - \|x_i - x_j\|^2 \right\}$$

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

(1) If rule (c) is satisfied, then $\overline{B}^j_k(x_i) = \emptyset$. In this case, we can directly reject all the objects in $S_l$ and set $\psi^j_k(x_i) = 0$.

(2) If rule (d) is satisfied, then $B^j_k(x_i) = S_l$. In this case, we can directly compute

$$\psi^j_k(x_i) = \sum_{x_i \in S_l} d_{h-1}^j - |S_l| \left( ||x_i - c||^2 - ||c||^2 \right) - \sum_{x_i \in S_l} ||x_j||^2.$$

On the basis of the above analyses, we can reduce lots of unnecessary distance calculations by $B^j_k(x_i)$ and $\overline{B}^j_k(x_i)$ for $1 \leq l \leq k$. We will further reduce the computational complexity. According to Step 4 of the FGKM algorithm, only the object $x_i$ with the maximum $b^i_k$ should be identified, indicating that computing the exact $b^i_k$ value for each $x_i$, $1 \leq i \leq n$, is unnecessary. Therefore, we will first use the approximate description of $B^j_k(x_i)$ by $B^j_k(x_i)$ and $\overline{B}^j_k(x_i)$ to calculate the upper bound $b^i_k$ of $b^i_k$ for $1 \leq i \leq n$, which is defined as follows:
\[ \hat{b}_h = \sum_{l=1}^{k'} \hat{\psi}_h(x_l). \] (13)

where

\[
\hat{\psi}_h(x_l) = \begin{cases} 
0 & \text{if } \overline{B}_h(x_l) = \emptyset, \\
|S_i| \left( ||x_l - c||^2 - ||c||^2 \right) + \sum_{x_i \in S_i} ||x_i||^2, & \text{if } \overline{B}_h(x_l) = S_i, \\
\sum_{x_i \in \overline{B}_h(x_l)} \left( d_{i-1}^l - \left( ||x_l - c||^2 - ||x_i - c||^2 \right) \right), & \text{otherwise.}
\end{cases}
\] (14)

Given that \( \hat{\psi}_h(x_l) \leq \hat{\psi}_h(x_k) \) for \( 1 \leq l \leq k' \), we have \( b_{l-1} \leq b_{l} \).

After obtaining the upper bound \( b_{l-1} \), if \( b_{l-1} < \varepsilon \leq \max_{l=1}^{k'} b_{l} \), we can easily conclude that \( x_l \) cannot be the new initial cluster center. This conclusion can further reduce the amount of distance calculations. Here, \( \varepsilon \) is a parameter set to the maximum value of all obtained exact \( b_{l} \) 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 \( h \)th initial cluster center, for \( 1 < h \leq k \). 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>
<thead>
<tr>
<th>Table 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>An acceleration mechanism for producing the ( h )th initial cluster center.</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>9</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>11</td>
</tr>
<tr>
<td>12</td>
</tr>
<tr>
<td>13</td>
</tr>
<tr>
<td>14</td>
</tr>
<tr>
<td>15</td>
</tr>
<tr>
<td>16</td>
</tr>
<tr>
<td>17</td>
</tr>
<tr>
<td>18</td>
</tr>
<tr>
<td>19</td>
</tr>
<tr>
<td>20</td>
</tr>
<tr>
<td>21</td>
</tr>
<tr>
<td>22</td>
</tr>
<tr>
<td>23</td>
</tr>
<tr>
<td>24</td>
</tr>
<tr>
<td>25</td>
</tr>
<tr>
<td>26</td>
</tr>
<tr>
<td>27</td>
</tr>
<tr>
<td>28</td>
</tr>
<tr>
<td>29</td>
</tr>
<tr>
<td>30</td>
</tr>
<tr>
<td>31</td>
</tr>
<tr>
<td>32</td>
</tr>
<tr>
<td>33</td>
</tr>
<tr>
<td>34</td>
</tr>
<tr>
<td>35</td>
</tr>
<tr>
<td>36</td>
</tr>
<tr>
<td>37</td>
</tr>
<tr>
<td>38</td>
</tr>
<tr>
<td>39</td>
</tr>
</tbody>
</table>
The FGKM+A algorithm

Table 2
Data sets from UCI.

<table>
<thead>
<tr>
<th>Data set</th>
<th>Objects</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Handwritten digits</td>
<td>5620</td>
<td>64</td>
</tr>
<tr>
<td>Statlog</td>
<td>6435</td>
<td>36</td>
</tr>
<tr>
<td>Musk</td>
<td>6598</td>
<td>168</td>
</tr>
<tr>
<td>Isolet</td>
<td>7797</td>
<td>617</td>
</tr>
<tr>
<td>Coil</td>
<td>9000</td>
<td>86</td>
</tr>
<tr>
<td>Letters</td>
<td>20,000</td>
<td>16</td>
</tr>
<tr>
<td>Shuttle</td>
<td>58,000</td>
<td>9</td>
</tr>
<tr>
<td>Corel image</td>
<td>68,040</td>
<td>89</td>
</tr>
</tbody>
</table>

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., \( S = \{ S_1, S_2, \ldots, S_k \} \). Furthermore, save the center of each subset and the distances between each object and these centers.

Step 2. Compute \( V_1 = \{ v_1 \} \) from the data set \( X \), where \( v_1 = \sum_{i=1}^{n} x_i / n \) 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 \( x_q \) from \( U \) as the \( h \)th initial cluster center. Set \( V = V_{h-1} \cup \{ x_q \} \).

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 \( d_{ij} \) for each \( x_j \in U \). Goto Step 2.

4. Space and time complexity

4.1. Space complexity

In the FGKM+A algorithm, we need to save a partition vector \( P_c = [p_1, p_2, \ldots, p_n] \), where \( p_i = l \) if the object \( x_i \) belongs to \( S_l \); the distance matrix \( D = [D_{ik}] \) which is a \( n \times k' \) real matrix; \( D_{ik} = ||x_i - c_l||^2 \) for \( 1 \leq i \leq n, 1 \leq l \leq k' \) and \( ||x_i||^2 \) for each object \( x_i \), \( 1 \leq i \leq n \). 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 $c_l$ of each subset $S_l$ and the distances between each object $x_i$ and all the centers can be obtained. Furthermore, we need to calculate $||x_i||^2$ for each object $x_i$, which is $O(n)$ operations. In Step 4, we need $O(n_1n_2)$ distance calculations to obtain the $h$th initial cluster center, where $n_1(n < n)$ and $n_2(n < n)$ are the numbers of objects and distance calculations required to obtain the exact $b_i$, respectively. To generate $k$ cluster centers, the proposed algorithm needs $O(nk't + n_1n_2k + nk^2t)$ distance calculations. Given that the computational complexity of FGKM is $O(n^2k + nk^2t)$ 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 2GB 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.](image1)

![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.](image2)
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 $k_0 = \sqrt{n}$ 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.
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


