Research Article
Incremental Fuzzy Clustering Based on Feature Reduction

Yongli Liu (1), Yajun Zhang (1), and Hao Chao (1)

School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo 454003, Henan, China

Correspondence should be addressed to Yongli Liu; yongli.buaa@gmail.com

Received 16 August 2021; Accepted 2 March 2022; Published 28 March 2022

Academic Editor: Vinod Sharma

Copyright © 2022 Yongli Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In the era of big data, more and more datasets are gradually beyond the application scope of traditional clustering algorithms because of their large scale and high dimensions. In order to break through the limitations, incremental mechanism and feature reduction have become two indispensable parts of current clustering algorithms. Combined with single-pass and online incremental strategies, respectively, we propose two incremental fuzzy clustering algorithms based on feature reduction. The first uses the Weighted Feature Reduction Fuzzy C-Means (WFRFCM) clustering algorithm to process each chunk in turn and combines the clustering results of the previous chunk into the latter chunk for common calculation. The second uses the WFRFCM algorithm for each chunk to cluster at the same time, and the clustering results of each chunk are combined and calculated again. In order to investigate the clustering performance of these two algorithms, six datasets were selected for comparative experiments. Experimental results showed that these two algorithms could select high-quality features based on feature reduction and process large-scale data by introducing the incremental strategy. The combination of the two phases can not only ensure the clustering efficiency but also keep higher clustering accuracy.

1. Introduction

Clustering technique is very important in data mining research. It can generate data groups in an unsupervised way and ensure high intragroup similarity and intergroup dissimilarity. Since the fuzzy set theory [1] was proposed and applied to cluster analysis [2], the research of fuzzy clustering grew rapidly. Today, many fuzzy clustering algorithms, represented by Fuzzy C-Means (FCM) clustering [3–5], have been proposed and widely used.

In the context of big data, datasets mainly show two characteristics: (1) large amount and (2) high dimension. When the FCM algorithm is used to process this kind of datasets, it lets all features participate in clustering with the same importance weights. Obviously, it will be easy to affect clustering accuracy and efficiency. In 2018, Yang [6] proposed Feature Reduction Fuzzy C-Means (FRFCM) clustering. In this algorithm, an objective function of feature reduction based on weighted entropy is designed, and the feature weight is calculated. If the influence of one feature is greater, the weight is higher, and vice versa. According to the weights, the original high-dimensional features are reduced to lower-dimensional space by feature reduction, which can effectively help improve the efficiency.

Although fuzzy clustering can effectively deal with high-dimensional feature data through feature reduction [7–13], it is still difficult to process large-scale data, especially streaming data. Previously, in order to realize large-scale data clustering [14–18], Hore et al. [19, 20] proposed two incremental algorithms, named SPFCM (Single-Pass Fuzzy C-Means) and OFCM (Online Fuzzy C-Means), based on single-pass and online clustering strategies, respectively. The single-pass method divides the whole dataset into several chunks. The centroids obtained by clustering the previous chunk participate in the clustering process of the next chunk until all the chunks are processed completely. However, in the online method, each chunk is clustered separately. Then centroids of each chunk form a new chunk, which is clustered again to generate the final results. Although SPFCM and OFCM both have the ability to deal with large-scale datasets, once the datasets are high-dimensional and sparse, the clustering results are difficult to be satisfactory. Therefore, Mei et al. [21] considered scalability and high-dimensional processing capability and expanded SPFCM and
OFCM to SPHFCM (Single-Pass Hyperspherical Fuzzy C-Means) [21] and OHFCM (Online Hyperspherical Fuzzy C-Means) [21], respectively. In their work, a centroid normalization step is added to each iteration to normalize all centroids to the unit norm, and the Cosine similarity instead of the Euclidean distance is used to measure the distance between centroid and each object.

Based on above analysis, in order to realize the clustering of large-scale and high-dimensional datasets, two incremental fuzzy c-means clustering algorithms based on feature reduction are proposed, named SPFRFCM (Single-Pass Feature Reduction Fuzzy C-Means) and OFRFCM (Online Feature Reduction Fuzzy C-Means). In these two algorithms, different weights are given to the features according to their importance, and the dimension reduction method is used to reduce the data dimension and improve clustering efficiency. Besides, the incremental method is employed so that they are capable of handling large-scale data.

The rest of the paper is organized as follows. In Section 2, we introduce some related work, including the FRFCM algorithm and incremental clustering based on FCM. Section 3 presents our two algorithms, SPFRFCM and OFRFCM. In Section 4, experiments are provided for demonstrating the performance of proposed algorithms. Section 5 concludes this paper.

2. Related Work

2.1. FRFCM. In the FRFCM algorithm, each feature has its own weight, and the weight value will be updated at each iteration. Let \( X = \{x_1, x_2, \ldots, x_N\} \) represent a \( D \)-dimensional dataset, let \( U \) be a membership matrix whose element \( u_{ci} \) represents the fuzzy membership of the \( i \)-th data \((1 \leq i \leq N)\) to the \( c \)-th \((1 \leq c \leq C)\) cluster, let \( V \) be a centroid matrix whose element \( v_{cj} \) represents the value of the \( j \)-th feature of the \( c \)-th centroid, and let \( W \) be a feature weight matrix whose element \( \omega_i \) represents the weight of the \( j \)-th feature \((1 \leq j \leq D)\), where \( N \) and \( C \) are the numbers of objects and clusters, respectively.

The objective function and constraints are expressed as follows:

\[
J_{\text{FRFCM}}(U, V, W) = \sum_{c=1}^{C} \sum_{i=1}^{N} \sum_{j=1}^{D} u_{ci}^m \delta_j \left( x_{ij} - v_{cj} \right)^2 + T_\omega \sum_{j=1}^{D} \left( \omega_j \log \delta_j \omega_j \right).
\]

\[
\sum_{c=1}^{C} u_{ci} = 1, 0 \leq u_{ci} \leq 1.
\]

\[
\sum_{j=1}^{D} \omega_j = 1, 0 \leq \omega_j \leq 1,
\]

where \( \delta_j \) is used to adjust the feature weight \( \omega_j \) and \( T_\omega \) depends on the values of \( N \) and \( C \); that is,

\[
\delta_j = \frac{\text{mean}(x_{ij})}{\text{var}(x_{ij})},
\]

\[
T_\omega = \frac{N}{C},
\]

2.2. Incremental Clustering Based on FCM. Both SPFCM and OFCM algorithms are based on WFCM (Weighted Fuzzy C-Means Clustering) [22–25], which assigns different weights to objects according to their importance. The higher the weight, the higher the importance, and vice versa. The objective function of WFCM algorithm is expressed as follows:

\[
J_{\text{WFCM}}(U, V) = \sum_{c=1}^{C} \sum_{i=1}^{N} \sum_{j=1}^{D} u_{ci}^m w_i (x_{ij} - v_{cj})^2,
\]

where \( w_i \) represents the weight of the \( i \)-th object. The iterative formulas of \( u_{ci} \) and \( v_{cj} \) can be obtained, respectively, by Lagrange multiplier method:

\[
u_{ci} = \frac{\left( \sum_{j=1}^{D} (x_{ij} - v_{cj})^2 \right)^{1/m}}{\sum_{c=1}^{C} \left( \sum_{j=1}^{D} (x_{ij} - v_{cj})^2 \right)^{1/m}} \]

\[
v_{cj} = \frac{\sum_{i=1}^{N} w_{ci} u_{ci}^m x_{ij}}{\sum_{i=1}^{N} w_{ci} u_{ci}^m}.
\]

After dividing dataset \( X \) into \( s \) chunks with size \( n \), that is, \( X = \{X^1, X^2, \ldots, X^s\} \), both SPFCM and OFCM process each chunk in turn. In SPFCM, let \( X^z \) represent the \( z \)-th chunk \((1 \leq z \leq s)\). After clustering \( X^z \) with WFCM algorithm, we get a centroid set \( \Delta_z = \{v_{c1}, v_{c2}, \ldots, v_{cz}\} \). Then these centroids and their weights are merged with the objects of the \( z \)-th chunk, and a new chunk \( X^{z+1} = [\Delta_z, X^{z+1}] \) is generated. On this new chunk, the WFCM algorithm will be carried out again. The above process is cycled until all chunks are processed.

Different from SPFCM, OFCM algorithm clusters \( s \) chunks, respectively, at the same time; and the centroids of each chunk form a total chunk, on which the WFCM algorithm will be implemented again to produce the final clustering result.

3. Incremental Fuzzy Clustering Based on Feature Reduction

Based on single-pass and online incremental framework, this paper proposes two fuzzy incremental clustering algorithms based on feature reduction, named SPFRFCM and OFRFCM, respectively. During the implementation of SPFRFCM and OFRFCM algorithms, a weighted FRFCM algorithm, named WFRFCM (Weighted Feature Reduction Fuzzy C-Means), is designed. Its objective function is
$$J_{WFRFCM}(U,V,W) = \sum_{c=1}^{C} \sum_{i=1}^{N} \sum_{j=1}^{D} u_{ci} m \delta_{j} \omega_{i} (x_{ij} - v_{cj})^2 + T_{\omega} \sum_{j=1}^{D} (\omega_{j} \log \delta_{j} \omega_{j}).$$  \hfill (9)

The constraints include
\begin{align*}
\sum_{c=1}^{C} u_{ci} &= 1, \quad 0 \leq u_{ci} \leq 1. \hfill (10) \\
\sum_{j=1}^{D} \omega_{j} &= 1, \quad 0 \leq \omega_{j} \leq 1. \hfill (11) \\
\omega_{i} &> 0. \hfill (12)
\end{align*}

According to the Lagrange multiplier method, by fixing \(U\) and \(W\) and calculating the partial derivative of \(V\), the value of \(v_{cj}\) can be calculated as follows:
\begin{equation}
v_{cj} = \frac{\sum_{i=1}^{N} u_{ci} m x_{ij}}{\sum_{i=1}^{N} u_{ci} m_{ci}}. \tag{13}
\end{equation}

Similarly, the iterative formulas of \(u_{ci}\) and \(\omega_{j}\) can be described as
\begin{align*}
\omega_{j} &= \frac{1/\delta_{j} \exp\left(-C/N \sum_{c=1}^{C} \sum_{i=1}^{N} u_{ci} m \delta_{i} \omega_{i} (x_{ij} - v_{cj})^2\right)}{\sum_{k=1}^{D} 1/\delta_{k} \exp\left(-C/N \sum_{c=1}^{C} \sum_{i=1}^{N} u_{ci} m \delta_{k} \omega_{k} (x_{ik} - v_{ck})^2\right)}, \tag{15}
\end{align*}

Next, we analyze the time complexity of WFRFCM algorithm. This algorithm contains three parts: (1) calculating the membership division \(u_{ci}\), (2) updating the cluster center \(v_{cj}\), and (3) updating the feature weight \(\omega_{j}\). The complexities of these three parts are \(O(NC^2D), O(NCD),\) and \(O(NCD^3),\) respectively. Therefore, the total computational complexity is \(O(NC^2D + NCD^3)\).

3.1 SPFRFCM. Before the SPFRFCM algorithm is implemented, the dataset is divided into \(s\) chunks with size \(n\), and the weight vector \(\omega_{z}\) is initialized with a \(1 \times n\) vector, that is, \(\omega_{z} = [1, 1, \ldots, 1]\). Let \(z\) be the index of chunk. Then this algorithm will run by performing the following steps:

(1) When \(z = 1\), WFRFCM algorithm is used to group this chunk directly; and we can obtain the centroid matrix \(\Delta_{1} = [v_{c1}, v_{c2}, \ldots, v_{c1}]\) and the corresponding data weight matrix:
\begin{equation}
u_{c1} = \sum_{i=1}^{n} u_{ci} \omega_{1}, \tag{16}\end{equation}

where \(\omega_{1} = 1, \leq i \leq n_{z}\), and \(n_{z}\) represents the size of the \(z\)-th chunk. In addition, we can also obtain the weight matrix \(\omega^{1} = [\omega_{11}, \omega_{12}, \ldots, \omega_{11}]\) of the features. The feature weight is
\begin{equation}
\omega_{1} = \frac{1/\delta_{1} \exp\left(-C/N \sum_{c=1}^{C} \sum_{i=1}^{N} u_{ci} m \delta_{i} \omega_{i} (x_{ij} - v_{cj})^2\right)}{\sum_{k=1}^{D} 1/\delta_{k} \exp\left(-C/N \sum_{c=1}^{C} \sum_{i=1}^{N} u_{ci} m \delta_{k} \omega_{k} (x_{ik} - v_{ck})^2\right)}. \tag{17}
\end{equation}

By minimizing the objective function, the formulas of membership \(u_{ci}\) and centroid \(v_{cj}\) after clustering chunk \(X^{1}\) are
\begin{align*}
u_{ci} &= \frac{\sum_{j=1}^{D} \delta_{j} \omega_{j} (x_{ij} - v_{cj})^2}{\sum_{c=1}^{C} \sum_{j=1}^{D} \delta_{j} \omega_{j} (x_{ij} - v_{cj})^2}^{1/1-m}, \tag{18} \\
v_{cj} &= \frac{\sum_{i=1}^{n} \omega_{1} \omega_{i} m x_{ij}}{\sum_{i=1}^{n} \omega_{1} \omega_{i} m_{ci}}. \tag{19}
\end{align*}

(2) When \(z > 1\), the centroids of previous chunk are added to current chunk, and a new chunk \(X^{z} = [\Delta_{1}, X^{z}]\) is obtained. Let \(w_{z}^{i}\) be the weights of elements in \(X^{z}\), and it contains \(C\) weights of centroids from the previous chunk and \(n_{z}\) weights of objects in current chunk. Then we implement WFRFCM clustering on this new chunk; and the corresponding data weight can be calculated as
\begin{equation}
u_{c1} = \sum_{i=1}^{C n_{z}} u_{ci} \omega_{1}, \tag{20}\end{equation}

Here \(\omega_{1} = 1\), and the feature weight can be updated as
\begin{equation}
\omega_{1} = \frac{1/\delta_{1} \exp\left(-C/N \sum_{c=1}^{C} \sum_{i=1}^{N} u_{ci} m \delta_{i} \omega_{i} (x_{ij} - v_{cj})^2\right)}{\sum_{k=1}^{D} 1/\delta_{k} \exp\left(-C/N \sum_{c=1}^{C} \sum_{i=1}^{N} u_{ci} m \delta_{k} \omega_{k} (x_{ik} - v_{ck})^2\right)}. \tag{21}
\end{equation}

By minimizing the objective function, we can get the formulas of membership \(u_{ci}\) and centroid \(v_{cj}\):
\begin{align*}
u_{ci} &= \frac{\sum_{j=1}^{D} \delta_{j} \omega_{j} (x_{ij} - v_{cj})^2}{\sum_{c=1}^{C} \sum_{j=1}^{D} \delta_{j} \omega_{j} (x_{ij} - v_{cj})^2}^{1/1-m}, \tag{22} \\
v_{cj} &= \frac{\sum_{i=1}^{n} \omega_{1} \omega_{i} m x_{ij}}{\sum_{i=1}^{n} \omega_{1} \omega_{i} m_{ci}}. \tag{23}
\end{align*}

The pseudocode of SPFRFCM algorithm is as follows:
Step 1: Initialize some parameters including the number of clusters \(C\), fuzzy index \(m = 2\), membership matrix \(U^{00}\), characteristic weight \(W^{00} = [\omega_{1}^{1/1-m}]\), number of iterations \(t = 1\), parameter \(\epsilon_{1} = 1/(DN)^{1/2}\), and parameter \(\epsilon_{2} = 1 \times 10^{-8}\).
Step 2: Divide the dataset into \( s \) chunks, that is, \( X = \{X^1, X^2, \ldots, X^s\} \); initialize the data weight, that is, \( w_z = [1, 1, \ldots, 1] \), and set \( z = 1 \).

Step 3: Calculate \( V^{(0)} \) with formula (19) or (23).

Step 4: Calculate \( W^{(0)} \) with formula (16) or (20).

Step 5: Calculate \( w_j \) with formula (17).

Step 6: If \( \omega_j \leq \epsilon_1 \), delete the \( j \)-th feature; update the number of data features \( d_{new} = D - d_r \), where \( d_r \) represents the number of features to be deleted.

Step 7: Calculate \( U^{(0)} \) with formula (18) or (22).

Step 8: \( z \prec s \) if \( \|U^t - U^{t-1}\|_2 > \epsilon_5 \), make \( t + 1 \longrightarrow t \) and go to Step 3; otherwise, add the centroids and weights to the next chunk, obtain a new chunk, and make \( z + 1 \longrightarrow z \) and go to Step 3.

Step 9: \( z \geq s \) if \( \|U^t - U^{t-1}\|_2 \geq \epsilon_5 \), make \( t + 1 \longrightarrow t \) and go to Step 3; otherwise, this algorithm ends.

3.2. OFRFCM. OFRFCM is implemented based on the online incremental framework. Firstly, \( s \) chunks are clustered, respectively, and then the centroids and weights of each chunk are merged to form a new chunk and weight matrix. This new chunk can be expressed as \( X_{new} = \{\Delta_1 \cup \Delta_2 \cup \ldots \cup \Delta_s\} \), and the weight of each chunk after clustering is expressed as \( u_i, z = [w_1, w_2, \ldots, w_1, w_2, \ldots, w_2, \ldots, w_s, w_1, ws, \ldots, ws, c] \). The specific steps are as follows:

1. Cluster each chunk, and obtain the centroid \( \Delta_j = [v_{1j}, v_{2j}, \ldots, v_{cj}] \) and the corresponding weight matrix:

\[
\omega_j = \sum_{i=1}^{n_j} u_{ij}w_i^2.
\]

In addition, the feature weight matrix \( \omega = [\omega_{1j}, \omega_{2j}, \ldots, \omega_{nj}] \) is also obtained. The feature weight is

\[
\omega_j = 1/\|D\| \sum_{i=1}^{n_j} \sum_{k=1}^{C} \omega_{k|j} \delta_{ik} w_i \delta_{ij} (x_{ij} - v_{cj})^2.
\]

By minimizing the objective function, we obtain the formulas of membership \( u_{ci} \) and centroid \( v_{cj} \):

\[
\begin{align*}
\mu_{ci} &= \left( \frac{\sum_{j=1}^{C} \delta_{ij} \omega_j (x_{ij} - v_{cj})^2}{\sum_{j=1}^{C} \delta_{ij} \omega_j} \right)^{1/(1-m)} \quad (26) \\
v_{cj} &= \frac{\sum_{i=1}^{n_c} \mu_{ci} x_{ij}}{\sum_{i=1}^{n_c} \mu_{ci}} \quad (27)
\end{align*}
\]

The centroids with different weights obtained from each chunk form a new dataset, on which the WFRFCM algorithm will be implemented again to generate the final results.

The formulas of membership \( u_{ci} \) and centroid \( v_{cj} \) obtained from the new chunk are

\[
\begin{align*}
\mu_{ci} &= \left( \frac{\sum_{j=1}^{C} \delta_{ij} \omega_j (x_{ij} - v_{cj})^2}{\sum_{j=1}^{C} \delta_{ij} \omega_j} \right)^{1/(1-m)} \quad (28) \\
v_{cj} &= \frac{\sum_{i=1}^{n_c} \mu_{ci} x_{ij}}{\sum_{i=1}^{n_c} \mu_{ci}} \quad (29)
\end{align*}
\]

where \( 1 \leq i \leq n_r \) and \( n_r \) is the number of objects in the new chunk.

The clustering process of OFRFCM algorithm can be described as the following pseudocode:

Step 1: Initialize some parameters including the number of clusters \( C \), fuzzy index \( m = 2 \), membership matrix \( U^{(0)} \), feature weight \( W^{(0)} = [w_j]_{1 \times D} \), iteration time \( t = 1 \), centroid matrix \( X_{new} \), data weight \( Q \), parameter \( \epsilon_1 = 1/(DN)^{1/2} \), and parameter \( \epsilon_2 = 1 \times 10^{-5} \).

Step 2: Divide the dataset into \( s \) chunks, that is, \( X = \{X^1, X^2, \ldots, X^s\} \); initialize the data weight, that is, \( w_z = [1, 1, \ldots, 1] \), and set \( z = 1 \).

Step 3: Calculate \( V^{(0)} \) with formula (27) or (29).

Step 4: Calculate \( W^{(0)} \) with formula (24).

Step 5: Calculate \( w_j \) with formula (25).

Step 6: If \( \omega_j \leq \epsilon_1 \), delete the \( j \)-th feature; update the number of data features \( d_{new} = D - d_r \), where \( d_r \) represents the number of features to be deleted.

Step 7: Calculate \( U^{(0)} \) with formula (26) or (28).

Step 8: \( z \prec s \) if \( \|U^t - U^{t-1}\|_2 > \epsilon_5 \), make \( t + 1 \longrightarrow t \) and go to Step 3; otherwise, add the centroids and weights to the centroid matrix \( X_{new} \) and object weight matrix \( Q \), respectively, make \( z + 1 \longrightarrow z \), and go to Step 3.

Step 9: \( z \geq s \) if \( \|U^t - U^{t-1}\|_2 \geq \epsilon_5 \), make \( t + 1 \longrightarrow t \) and go to Step 3; otherwise, this algorithm ends.

4. Experiments

4.1. Experimental Preparation. In order to verify the effectiveness of our algorithms, six datasets and six incremental comparison clustering algorithms are selected for experiments. The specific information of each dataset is listed in Table 1; and the comparison algorithms include three single-pass algorithms and three online algorithms. The three single-pass algorithms are SPFCM, SPFCM, and SPFCOM (Single-Pass Fuzzy C-Ordered-Means clustering) [26], and three online algorithms are OFCOM, OFFCM, and OFCOM (Online Fuzzy C-Ordered-Means clustering) [26]. SPFCOM and OFCOM are two incremental fuzzy c-ordered-means clustering algorithms using single-pass and online architectures, respectively, and these two algorithms can achieve better robustness. The final experimental results will be
evaluated by two evaluation criteria, accuracy (AC) and F-Measure (FM) [27].

4.2. Experimental Results. The experiments are divided into three parts. The first part evaluates clustering accuracy by comparing the clustering results of our algorithms with six comparison algorithms, the second part of experiments verifies the effectiveness of the dimension reduction, and the third part evaluates the robustness of our algorithms.

4.2.1. Algorithm Accuracy. This group of experiments first compares our SPFRFCM algorithm with three single-pass comparison algorithms and then compares our OFRFCM algorithm with three online comparison algorithms. In our experiment, the chunk size is set to 5%, 10%, 20%, and 50% of the total number of objects in each dataset, so as to verify the clustering performance under different chunk sizes. In addition, the experiments adopt the same parameter setting as FRFCM algorithm, and the final clustering result is the average value of each algorithm running independently for 10 times. The experimental results of four single-pass algorithms and four online algorithms are shown in Tables 2 and 3, respectively.

It can be seen from Table 2 that when the chunk size is 5%, compared with SPFCOM, SPHFCM, and SPFCM algorithms, the average FM value of SPFRFCM algorithm on six datasets is increased by 10.98%, 13.89%, and 18.55%, respectively, and the average AC index is increased by 10.97%, 12.71%, and 18.22%, respectively. When the chunk size is 10%, the average FM index of SPFRFCM algorithm is 13.65%, 15.12%, and 17.94% higher than those of SPFCOM, SPHFCM, and SPFCM algorithms, respectively, and the AC index is 9.93%, 11.96%, and 15.34% higher, respectively. When the chunk size is 20%, the improvement ranges of average FM index are 14.33%, 15.22%, and 14.28%, respectively, and the improvement ranges of AC index are 11.48%, 15.22%, and 12.35%, respectively. When the chunk size is 50%, the improvement ranges in terms of FM are 11.26%, 15.12%, and 19.02%, respectively, and the improvement ranges in terms of AC are 7.37%, 13.06%, and 17.53%, respectively. From the result distribution, SPFRFCM algorithm has greatly improved on such three datasets as WR, WB, and SE, and the FM and AC indexes have increased by 20.16% and 10.15%, respectively, on average. The improvements on GS, PB, and IR datasets are less, and the FM and AC indexes increase by 5.76% and 5.88%, respectively, on average.

Table 3 lists the clustering accuracy comparison results of the four online algorithms. It can be seen from this table that when the chunk size is 5%, the average FM index of the OFRFCM algorithm on the six datasets is 14.98%, 25.19%, and 14.65% higher than those of the OFCOM, OHFCM, and OFCM algorithms, respectively, and the AC index is 14.20%, 34.88%, and 18.66% higher, respectively. When the chunk size is 10%, the average FM index of OFRFCM algorithm is 13.13%, 22.95%, and 13.74% higher, and the AC index is 9.22%, 21.14%, and 16.37% higher, respectively. When the chunk size is 20%, the improvements of FM index of OFRFCM algorithm are 14.76%, 23.63%, and 13.49%, respectively, and the improvements of AC index are 11.45%, 31.20%, and 15.51%, respectively. When the chunk size is 50%, the improvement ranges of FM index are 13.40%, 26.72%, and 16.97%, respectively, and the improvement ranges of AC index are 9.77%, 33.29%, and 24.62%, respectively. From the result distribution, OFRFCM algorithm has greatly improved on SE, WR, and WB datasets, and the FM and AC indexes have increased by 27.40% and 25.68%, respectively, on average. The improvements in GS, PB, and IR datasets are less, and the FM and AC indexes increase by 8.20% and 14.37%, respectively, on average.

Through this group of experiments, it can be seen that both SPFRFCM and OFRFCM algorithms obtain higher accuracy than the comparative experimental algorithms on the experimental dataset, and the improvements are obvious on some datasets. The reason is that some features with low weights may play a negative role in the clustering process. Therefore, it may help improve clustering accuracy by filtering out these features using feature reduction methods.

4.2.2. Feature Reduction. In the process of feature reduction, the features whose weights are lower than the threshold will be discarded. In order to verify the effectiveness of the feature reduction, the second part of experiments also sets the chunk size with 5%, 10%, 20%, and 50% of the total number of objects in each dataset and records the number of features. The experimental results are shown in Figures 1 and 2.

Figures 1 and 2 describe the performances of SPFRFCM and OFRFCM algorithms in the process of feature reduction, respectively. On IR dataset, both algorithms reduce data features by 50%. On WB dataset, SPFRFCM algorithm reduces features by up to 70%, while OFRFCM algorithm reduces features by up to 33%. On WR dataset, SPFRFCM algorithm reduces the number of features by 73%, while OFRFCM algorithm reduces it by 9%. On PB dataset, both algorithms do not reduce the number of features. On SE dataset, SPFRFCM algorithm is better than OFRFCM algorithm in feature reduction when the chunk size is 5% and 10%, and they are equivalent when the chunk size is 20% and 50%. On GS dataset, the number of features of SPFRFCM algorithm is reduced to 22 on average, while the number of features of OFRFCM algorithm is reduced to 120 on average. It can be seen that, in our experiments, the feature reduction method can effectively reduce the number of features.
| Datasets | Algorithms | SPRFCM | OFRFCM | SPFCOM | OFCOM | SPHFCM | OHFCM | SPPCM | OFCM |
|----------|------------|--------|--------|--------|-------|--------|-------|-------|------|
|          |            | FM    | AC    | FM    | AC    | FM    | AC    | FM    | AC   |
| IR       | 5%         | 0.895 | 0.941 | 0.880 | 0.933 | 0.881 | 0.873 | 0.778 | 0.854 |
|          | 10%        | 0.906 | 0.951 | 0.858 | 0.911 | 0.893 | 0.893 | 0.790 | 0.869 |
|          | 20%        | 0.904 | 0.949 | 0.888 | 0.937 | 0.886 | 0.813 | 0.796 | 0.875 |
|          | 50%        | 0.900 | 0.947 | 0.900 | 0.923 | 0.855 | 0.831 | 0.788 | 0.846 |
| WB       | 5%         | 0.823 | 0.874 | 0.751 | 0.794 | 0.624 | 0.717 | 0.708 | 0.729 |
|          | 10%        | 0.881 | 0.930 | 0.710 | 0.726 | 0.634 | 0.737 | 0.744 | 0.803 |
|          | 20%        | 0.863 | 0.918 | 0.698 | 0.717 | 0.631 | 0.735 | 0.782 | 0.847 |
|          | 50%        | 0.857 | 0.916 | 0.764 | 0.818 | 0.658 | 0.767 | 0.852 | 0.909 |
| WR       | 5%         | 0.372 | 0.375 | 0.281 | 0.287 | 0.266 | 0.281 | 0.267 | 0.254 |
|          | 10%        | 0.346 | 0.329 | 0.280 | 0.279 | 0.264 | 0.277 | 0.263 | 0.251 |
|          | 20%        | 0.347 | 0.360 | 0.270 | 0.281 | 0.275 | 0.291 | 0.271 | 0.261 |
|          | 50%        | 0.347 | 0.341 | 0.268 | 0.291 | 0.267 | 0.284 | 0.259 | 0.246 |
| PB       | 5%         | 0.608 | 0.698 | 0.512 | 0.606 | 0.578 | 0.670 | 0.443 | 0.550 |
|          | 10%        | 0.607 | 0.690 | 0.492 | 0.586 | 0.582 | 0.661 | 0.448 | 0.545 |
|          | 20%        | 0.609 | 0.675 | 0.506 | 0.617 | 0.586 | 0.664 | 0.462 | 0.572 |
|          | 50%        | 0.611 | 0.680 | 0.514 | 0.635 | 0.589 | 0.648 | 0.440 | 0.529 |
| SE       | 5%         | 0.430 | 0.457 | 0.418 | 0.420 | 0.411 | 0.420 | 0.418 | 0.438 |
|          | 10%        | 0.432 | 0.455 | 0.423 | 0.509 | 0.414 | 0.426 | 0.414 | 0.425 |
|          | 20%        | 0.431 | 0.446 | 0.381 | 0.429 | 0.416 | 0.421 | 0.417 | 0.432 |
|          | 50%        | 0.425 | 0.435 | 0.397 | 0.415 | 0.412 | 0.424 | 0.418 | 0.435 |
| GS       | 5%         | 0.553 | 0.585 | 0.456 | 0.540 | 0.365 | 0.434 | 0.411 | 0.485 |
|          | 10%        | 0.557 | 0.603 | 0.449 | 0.526 | 0.379 | 0.451 | 0.442 | 0.522 |
|          | 20%        | 0.611 | 0.658 | 0.412 | 0.527 | 0.359 | 0.426 | 0.451 | 0.531 |
|          | 50%        | 0.609 | 0.669 | 0.488 | 0.551 | 0.359 | 0.429 | 0.424 | 0.508 |

Table 2: Accuracy comparison of four online algorithms.

| Datasets | Algorithms | OFRFCM | OFCOM | OHFCM | OFCM |
|----------|------------|--------|-------|-------|------|
|          |            | FM    | AC    | FM    | AC   |
| IR       | 5%         | 0.924 | 0.960 | 0.882 | 0.913 |
|          | 10%        | 0.925 | 0.960 | 0.887 | 0.909 |
|          | 20%        | 0.919 | 0.957 | 0.896 | 0.933 |
|          | 50%        | 0.910 | 0.952 | 0.908 | 0.924 |
| WB       | 5%         | 0.825 | 0.892 | 0.682 | 0.694 |
|          | 10%        | 0.732 | 0.744 | 0.695 | 0.707 |
|          | 20%        | 0.873 | 0.926 | 0.670 | 0.698 |
|          | 50%        | 0.826 | 0.865 | 0.726 | 0.760 |
| WR       | 5%         | 0.394 | 0.387 | 0.307 | 0.319 |
|          | 10%        | 0.412 | 0.391 | 0.299 | 0.315 |
|          | 20%        | 0.330 | 0.350 | 0.307 | 0.322 |
|          | 50%        | 0.391 | 0.343 | 0.290 | 0.319 |
| PB       | 5%         | 0.912 | 0.909 | 0.861 | 0.851 |
|          | 10%        | 0.923 | 0.904 | 0.860 | 0.855 |
|          | 20%        | 0.895 | 0.898 | 0.896 | 0.899 |
|          | 50%        | 0.895 | 0.898 | 0.875 | 0.876 |
| SE       | 5%         | 0.553 | 0.585 | 0.456 | 0.540 |
|          | 10%        | 0.557 | 0.603 | 0.449 | 0.526 |
|          | 20%        | 0.611 | 0.658 | 0.412 | 0.527 |
|          | 50%        | 0.609 | 0.669 | 0.488 | 0.551 |
| GS       | 5%         | 0.316 | 0.351 | 0.291 | 0.305 |
|          | 10%        | 0.324 | 0.365 | 0.324 | 0.365 |
|          | 20%        | 0.305 | 0.324 | 0.305 | 0.324 |
|          | 50%        | 0.279 | 0.353 | 0.267 | 0.320 |

Table 3: Accuracy comparison of four single-pass algorithms.
In addition, it can be seen from the first part of experiments that the clustering accuracy with feature reduction has been improved, which indicates that some noise features are filtered in the feature reduction process. Obviously, noisy features will not only affect the clustering accuracy but also reduce the clustering efficiency. When noisy features are filtered, the robustness of algorithms will be improved, and the clustering accuracy is also easy to be improved.

After feature reduction, clustering time will be reduced, and thus clustering efficiency will be improved. The average running time of each algorithm in the experiments is shown in Table 4. It can be seen from this table that, for high-dimensional datasets, compared with other single-pass comparison algorithms SPFCOM, SPHFCM, and SPFCM, the running time of SPFRFCM algorithm is reduced by 70.2%, 12.6%, and 38.5%, respectively. The running time of OFRFCM algorithm is equivalent to that of other online comparison algorithms and has not been significantly improved, which is related to the incremental strategy. Due to the different incremental strategies of single-pass and online, when clustering each chunk, the single-pass incremental algorithm will update the feature weight and filter out the low weight features. The updated features participate in the calculation of subsequent chunks and directly improve the clustering efficiency of subsequent chunks. Therefore, when SPFRFCM completes the clustering task of the whole dataset, it can greatly reduce the clustering time. However, the online incremental algorithm groups each chunk separately, and the low weight features in each chunk are filtered out in the clustering process. It can be seen that the feature reduction of each chunk is relatively independent, resulting in little change in the overall efficiency. Therefore, the performance of single-pass incremental algorithm in feature reduction and efficiency improvement is better than online incremental algorithm.

4.2.3. Robustness. This part of experiments aims to evaluate the robustness of our algorithms. In our experiments, the size of the data chunk is set to 50%, and the noise data is added to 6 datasets in the proportions of 5%, 10%, 20%, and 50% for clustering, respectively. The experimental results are shown in Figures 3–8.

For the four single-pass algorithms, when the noise data proportion is 5%, the average FM values of SPFRFCM algorithm on the six datasets are 8.86%, 6.40%, and 1.75% higher than those of SPFCM, SPHFCM, and SPFCOM algorithms, respectively, and the AC values are 8.18%, 10.65%, and 2.3% higher, respectively. When the noise data proportion is 10%, the mean improvements of SPFRFCM algorithm in terms of FM index are 8.10%, 9.81%, and 2.4%, respectively, and the mean performances in terms of AC index are 6.10% and 10.30% higher than those of SPFCM and SPHFCM, respectively and 0.67% lower than that of SPFCOM algorithm. When the proportion is 20%, the mean values of FM index of SPFRFCM algorithm are 6.28% and 7.85% higher than those of SPFCM and SPHFCM.
Table 4: Comparison of running time.

| Datasets | SPFRFCM | SPFCOM | SPHFCM | SPCFM | OFRFCM | OFCOM | OHFCM | OFCM |
|----------|---------|--------|--------|-------|--------|-------|-------|------|
| IR       | 0.21    | 0.93   | 0.253  | 0.50  | 0.31   | 1.05  | 0.26  | 0.17 |
| WR       | 2.16    | 9.65   | 4.123  | 2.30  | 3.83   | 10.81 | 3.16  | 1.29 |
| WR       | 124.25  | 371.35 | 136.67 | 97.10 | 183.93 | 623.54| 162.17| 67.45|
| PB       | 74.34   | 378.74 | 106.38 | 90.46 | 78.26  | 377.46| 48.18 | 80.63|
| SE       | 8.20    | 181.30 | 13.28  | 10.75 | 45.30  | 174.00| 41.27 | 21.32|
| GS       | 1727.61 | 5558.61| 1955.02| 2949.61| 4826.75| 7419.58| 4631.62| 3652.84|

Figure 3: Robustness comparison on the IR dataset. (a) Accuracy; (b) F-Measure.

Figure 4: Robustness comparison on the WB dataset. (a) Accuracy; (b) F-Measure.
algorithms and 1.01% lower than that of SPFCOM algorithm, and the mean values of AC index are 8.05% and 11.1% higher than those of SPFCM and SPHFCM algorithm and 0.4% lower than that of SPFCOM algorithm. When the noise data proportion is 50%, the mean values of FM index are 7.60% and 6.93% higher than those of SPFCM and SPHFCM algorithms and 4.27% lower than that of SPFCOM algorithm, and the mean values of AC index are 7.95% and 9.56% higher than those of SPFCM and SPHFCM algorithm and 4.83% lower than that of SPFCOM algorithm. The above experimental results show that the robustness of SPFRFCM algorithm is higher than that of SPFCOM algorithm and that of SPHFCM algorithm. Compared with SPFCOM algorithm, when the noise ratio is less than 10%, the robustness of SPFRFCM algorithm is better; and when the noise ratio is between 10% and 20%, there is little difference between these two algorithms. However, when the noise ratio is higher than 20%, the robustness of SPFCOM algorithm is higher than that of SPFRFCM algorithm. We know that the Fuzzy C-Ordered-Means (FCOM) clustering is famous for its scenery and its insensitivity to the presence of noise and outliers in data, which is further verified by our experiments.

For the four online algorithms, when the noise data proportion is 5%, the average FM index values of OFRFCM...
algorithm on the six datasets are 5.38%, 8.18%, and 1.53% higher than those of OFCM, OHFCM, and OFCOM algorithms, respectively, and the AC index values are 6.81%, 10.56%, and 1.85% higher, respectively. When the noise data proportion is 10%, the average FM index values of OFRFCM algorithm are 7.33%, 10.86%, and 3.55% higher, respectively, and the AC index values are 7.88%, 11.28%, and 0.98% higher, respectively. When the proportion is 20%, the mean improvements of OFRFCM algorithm in terms of FM index are 7.20%, 9.00%, and 0.26%, respectively, and the mean improvements in terms of AC index are 9.98%, 11.40%, and −0.98%. When the noise data proportion is 50%, the mean values of FM index of OFRFCM algorithm are 5.66%, 8.18%, and 1.53% higher than those of OFCM, OHFCM, and OFCOM algorithms, respectively, and the mean values of AC index are 8.01% and 8.20% higher than those of OFCM and OHFCM, respectively, and 3.17% lower than that of OFCOM algorithm. This part of experiments show that the robustness of OFRFCM algorithm is better than that of OFCM algorithm and that of OHFCM algorithm and proves the better robustness performance of FCOM again.

Figure 7: Robustness comparison on the SE dataset. (a) Accuracy; (b) F-Measure.

Figure 8: Robustness comparison on the GS dataset. (a) Accuracy; (b) F-Measure.
5. Conclusions

While the scale of dataset is becoming larger and larger, the dimension is also increasing. Traditional clustering algorithms are difficult to deal with large-scale and high-dimensional data. Therefore, two incremental fuzzy c-means clustering algorithms, named SPFRFCM and OFRFCM, are proposed. The former divides the whole dataset into several chunks and clusters the chunks successively based on WFRFCM algorithm, and the clustering results of the previous chunk participate in the clustering process of the next chunks. The latter clusters each chunk based on WFRFCM algorithm and then merges clustering results of all chunks to cluster again. These two algorithms combine the advantages of incremental framework and feature reduction. They not only have the ability to deal with large-scale data but also can reduce feature dimension and help to improve the robustness and efficiency of clustering through feature reduction. In order to evaluate the effectiveness of these two algorithms, experiments were carried out on six datasets. Experimental results show that SPFRFCM algorithm can effectively reduce the feature dimension through feature reduction and obtain higher clustering accuracy and efficiency than the comparison algorithms. In OFRFCM, although the clustering efficiency has not been significantly improved due to the strategy, the feature dimension has also been reduced and the clustering accuracy has also been improved.

Data Availability

All the datasets used in this paper are derived from the UCI (University of California Irvine) Machine Learning Repository. Please visit https://archive.ics.uci.edu/ml/datasets.php.

Conflicts of Interest

The authors claim that there are no conflicts of interest in terms of the publication of this paper.

Acknowledgments

The authors would like to thank members of the IR&DM Research Group from Henan Polytechnic University for their invaluable advice that makes this paper successfully completed. The authors would like to acknowledge the support of National Science Fund subsidized project under Grant no. 61872126.

References

[1] L. A. Zadeh, "Fuzzy sets," Information and Control, vol. 8, no. 3, pp. 338–353, 1965.
[2] E. H. Ruspini, "A new approach to clustering," Information and Control, vol. 15, no. 1, pp. 22–32, 1969.
[3] J. C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," Journal of Cybernetics, vol. 3, no. 3, pp. 32–57, 1973.
[4] J. C. Bezdek, Objective Function clustering[M]/Pattern Recognition with Fuzzy Objective Function Algorithms, Springer, Boston, MA, pp. 43–93, 1981.
[5] J. C. Bezdek, R. Ehrlich, and W. F. C. M. Full, "The fuzzy c-means clustering algorithm[]", Computers & Geosciences, vol. 10, no. 2-3, pp. 191–203, 1984.
[6] M. S. Yang and Y. Nataliani, "A feature-reduction fuzzy clustering algorithm based on feature-weighted entropy[]", IEEE Transactions on Fuzzy Systems, vol. 26, no. 2, pp. 817–835, 2017.
[7] N. Van Pham, L. T. Pham, W. Pedrycz et al., "Feature-reduction fuzzy co-clustering approach for hyper-spectral image analysis[]", Knowledge-Based Systems, vol. 216, no. 2021, Article ID 106549, 2021.
[8] D. Kshirsagar and S. Kumar, "An efficient feature reduction method for the detection of DoS attack," ICT Express, vol. 7, no. 3, pp. 371–375, 2021.
[9] Z. Sun, H. Huo, J. Huan, and J. S. Vitter, "Feature reduction based on semantic similarity for graph classification," Neurocomputing, vol. 397, pp. 114–126, 2020.
[10] R. Jain and W. Xu, "RHDSII: a novel dimensionality reduction based algorithm on high dimensional feature selection with interactions," Information Sciences, vol. 574, pp. 590–605, 2021.
[11] K. Malarvizhi and K. Amshakala, "Feature linkage weight based feature reduction using fuzzy clustering method[]", Journal of Intelligent and Fuzzy Systems, vol. 40, no. 3, pp. 1–10, 2021, Preprint.
[12] J. Kianat, M. A. Khan, M. Sharif, T. Akram, A. Rehman, and T. Saba, "A joint framework of feature reduction and robust feature selection for cucumber leaf diseases recognition," Optik, vol. 240, Article ID 166566, 2021.
[13] M. P. Uddin, M. A. Mamun, and M. A. Hossain, "PCA-based feature reduction for h remote sensing image classification," IEEE Technical Review, vol. 38, no. 4, pp. 377–396, 2021.
[14] A. V. Ushakov and I. Vasilyev, "Near-optimal large-scale k-medoids clustering," Information Sciences, vol. 545, no. 3, pp. 344–362, 2021.
[15] Y. Yang, S. Deng, J. Lu et al., "GraphLSHC: towards large scale spectral hypergraph clustering," Information Sciences, vol. 544, pp. 117–134, 2021.
[16] Y. Yin, Y. Zhao, H. Li, and X. Dong, "Multi-objective evolutionary clustering for large-scale dynamic community detection," Information Sciences, vol. 549, pp. 269–287, 2021.
[17] A. Keshavarzi, A. Toroghi Haghighat, and M. Bohlouli, "Clustering of large scale QoS time series data in federated clouds using improved variable Chromosome Length Genetic Algorithm (CQGA)," Expert Systems with Applications, vol. 164, Article ID 113840, 2021.
[18] Y. Chen, L. Zhou, N. Bouguila, C. Wang, Y. Chen, and J. Du, "BLOCK-DBSCAN: e,” Pattern Recognition, vol. 109, Article ID 107624, 2021.
[19] P. Hore, L. O. Hall, and D. B. Goldgof, "Single pass fuzzy c means," in Proceedings of the 2007 IEEE International Fuzzy Systems Conference, pp. 1–7, IEEE, London, UK, July 2007.
[20] P. Hore, L. O. Hall, and D. B. Goldgof, "A scalable framework for segmenting magnetic resonance images[]", Journal of signal processing systems, vol. 54, no. 1, pp. 183–203, 2009.
[21] C. H. Kim, M. S. Kim, and J. J. Lee, "Incremental hyperplane-based fuzzy clustering for system modeling," in Proceedings of the IECON 2007-33rd Annual Conference of the IEEE Industrial Electronics Society, pp. 614–619, IEEE, Taipei, Taiwan, November 2007.
[22] H.-J. Xing and M.-H. Ha, “Further improvements in feature-weighted fuzzy C-means,” *Information Sciences*, vol. 267, pp. 1–15, 2014.

[23] Y. Unal, K. Polat, and H. Erdinc Kocer, “Pairwise FCM based feature weighting for improved classification of vertebral column disorders,” *Computers in Biology and Medicine*, vol. 46, pp. 61–70, 2014.

[24] K. Polat, “Classification of Parkinson’s disease using feature weighting method on the basis of fuzzy C-means clustering [J],” *International Journal of Systems Science*, vol. 43, no. 4-6, pp. 597–609, 2012.

[25] M.-S. Yang and K. P. Sinaga, “Collaborative feature-weighted multi-view fuzzy c-means clustering,” *Pattern Recognition*, vol. 119, Article ID 108064, 2021.

[26] Y. Liu, C. Guo, H. Wang, and H. Chao, “Incremental fuzzy C-ordered means clustering[J],” *Journal of Beijing University of Posts and Telecommunications*, vol. 41, no. 4, pp. 29–36, 2018.

[27] A. Maratea, A. Petrosino, and M. Manzo, “Adjusted F-measure and kernel scaling for imbalanced data learning,” *Information Sciences*, vol. 257, pp. 331–341, 2014.