IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 3, 2018, accepted November 20, 2018, date of publication December 4, 2018,
date of current version December 31, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2884956

A Fully-Unsupervised Possibilistic C-Means Clustering Algorithm

MIIN-SHEN YANG[1], SHOU-JEN CHANG-CHIEN[1], AND YESSICA NATALIANI[1,2]
[1]Department of Applied Mathematics, Chung Yuan Christian University, Chungli 32023, Taiwan
[2]Department of Information Systems, Satya Wacana Christian University, Salatiga 50711, Indonesia

Corresponding author: Miin-Shen Yang (msyang@math.cycu.edu.tw)

This work was supported by the Ministry of Science and Technology, Taiwan, under Grant MOST 107-2118-M-033-002-MY2.

ABSTRACT In 1993, Krishnapuram and Keller first proposed possibilistic C-means (PCM) clustering by relaxing the constraint in fuzzy C-means of which memberships for a data point across classes sum to 1. The PCM algorithm tends to produce coincident clusters that can be a merit of PCM as a good mode-seeking algorithm, and so various extensions of PCM had been proposed in the literature. However, the performance of PCM and its extensions heavily depends on initializations and parameters selection with a number of clusters to be given *a priori*. In this paper, we propose a novel PCM algorithm, termed a fully unsupervised PCM (FU-PCM), without any initialization and parameter selection that can automatically find a good number of clusters. We start by constructing a generalized framework for PCM clustering that can be a generalization of most existing PCM algorithms. Based on the generalized PCM framework, we propose the new type FU-PCM so that the proposed FU-PCM algorithm is free of parameter selection and initializations without a given number of clusters. That is, the FU-PCM becomes a FU-PCM clustering algorithm. Comparisons between the proposed FU-PCM and other existing methods are made. The computational complexity of the FU-PCM algorithm is also analyzed. Some numerical data and real data sets are used to show these good aspects of FU-PCM. Experimental results and comparisons actually demonstrate the proposed FU-PCM is an effective parameter-free clustering algorithm that can also automatically find the optimal number of clusters.

INDEX TERMS Clustering, fuzzy clustering, possibilistic clustering, fuzzy C-means (FCM), possibilistic C-means (PCM), fully-unsupervised PCM (FU-PCM).

I. INTRODUCTION

Cluster analysis is a method for clustering a data set into groups of similar individuals with the most similar objects in the same cluster and the most dissimilar objects between different clusters. It is a branch in statistical multivariate analysis and an unsupervised learning method used in pattern recognition. Since Zadeh [1] proposed fuzzy sets that produced the idea of partial memberships described by a membership function, fuzzy clustering has been widely studied and applied in various areas (see [2]–[5]). In fuzzy clustering, the fuzzy c-means (FCM) clustering algorithm proposed by Dunn [6] and Bezdek [2] is the best known method where the FCM with feature reduction has been recently considered in Yang and Nataliani [7]. Although FCM and its extensions are often used, the memberships do not always correspond well to the membership degree of data points. To improve on this weakness of FCM, Krishnapuram and Keller [8] created a possibilistic approach to clustering, called a possibilistic c-means (PCM), which used a possibilistic type of membership function to describe the degree of belonging by relaxing the constraint of the fuzzy c-partition summation to 1. Krishnapuram and Keller [8] showed algorithms with possibilistic c-memberships, such as PCM, are more robust to noise and outliers than FCM.

However, Barni *et al.* [9] presented PCM may sometimes produce coincident clusters. Afterwards, Krishnapuram and Keller [10] provided more insights and recommendations that PCM can be seen as a mode-seeking algorithm. There are many varieties of PCM in the literature. Zhang and Leung [11] proposed an improved PCM to avoid the tendency of identical clusters by building the fuzzy c-partitions into a PCM objective function. Timm *et al.* [12] introduced a mutual repulsion of the clusters into the PCM. Pal *et al.* [13] proposed a possibilistic FCM algorithm with a hybridization of PCM and FCM. Yang and Wu [14] proposed an extended PCM objective function so the resulting possibilistic memberships become exponential-type functions. Tseng and Kao [15] used the mountain method proposed by Yager and Filev [16] and correlation coefficient algorithm proposed by Yang and Wu [17] to create a so-called similarity-based PCM algorithm. Filippone *et al.* [18] applied the PCM algorithm in kernel-induced spaces, and Chang *et al.* [19] recently applied it to stepwise c-regressions. Yang and Lai [20] proposed an automatic merging possibilistic clustering method (AM-PCM). Xenaki *et al.* [21] developed adaptive possibilistic c-means (APCM) by handling the penalty parameters of PCM.

In this paper, we construct a generalized clustering framework for PCM. Based on the generalized framework, we can induce most of existing PCM algorithms proposed in the literature. Furthermore, we develop a new type of PCM algorithm, called fully-unsupervised PCM (FU-PCM), based on the generalized framework. Although the PCM algorithm can be as a good mode-seeking algorithm, PCM with its various extensions may heavily depends on initializations and parameters selection with a number of clusters to be given a priori. Our new constructing FU-PCM is free of parameter selection and initializations with automatically finding the optimal number of clusters. That is, the proposed FU-PCM becomes a fully-unsupervised PCM algorithm. Some experiments with numerical and real data sets are used to demonstrate the effectiveness and usefulness of our proposed algorithm. The organization of this paper is as follows. Section II describes some related works on PCM and our generalized PCM clustering framework. Section III constructs the proposed FU-PCM algorithm. Section IV gives several examples with numeric and real data sets being made to demonstrate effectiveness and usefulness of the proposed algorithm. Finally, conclusions are stated in Section V.

II. RELATED WORKS

Let $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ be a set of N data points in S-dimensional space with the jth data point $\mathbf{x}_j^T = [x_{jd}]_{1\times S}$. The clustering of $\mathbf{X}$ is a technique used to partition the data set $\mathbf{X}$ into C subsets that can effectively represent the data structure of $\mathbf{X}$. The partition of C clusters can be described by a $C \times N$ partition matrix $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_C]^T = [\mu_{ij}]_{C\times N}$ where each element μ_{ij} of $\mathbf{U}$ represents the membership of $\mathbf{x}_j$ belonging to the ith cluster. In general, there are three kinds of partition matrices used in clustering: 1) The hard C-partitions $\mathbf{U}_H$ with $\mu_{ij} \in \{0, 1\}$ for all i and j, where $\sum_{i=1}^{C} \mu_{ij} = 1$ for each j; 2) The fuzzy C-partitions $\mathbf{U}_F$ with $\mu_{ij} \in [0, 1]$ for all i and j, where $\sum_{i=1}^{C} \mu_{ij} = 1$ for each j; 3) The possibilistic C-memberships $\mathbf{U}_P$ with $\mu_{ij} \in [0, 1]$ for all i and j, where $\sum_{i=1}^{C} \mu_{ij} > 0$ for each j.

The best known clustering algorithm with hard C-partitions $\mathbf{U}_H$ is K-means (or called hard C-means). The fuzzy C-means (FCM) clustering algorithm with fuzzy C-partitions $\mathbf{U}_F$ is a well-known fuzzy extension of K-means. The FCM is an iterative algorithm using the necessary conditions for minimizing the objective function J_{FCM} with

$$J_{FCM}(\mathbf{U}, \mathbf{A}) = \sum_{i=1}^{C}\sum_{j=1}^{N} \mu_{ij}^{m} d^2(\mathbf{x}_j, \mathbf{a}_i) \tag{1}$$

where the weighting exponent $m \in [1, \infty)$ is a fuzziness index; $\mu_{ij} \in \mathbf{U}_F$ are fuzzy C-partitions; $\mathbf{A} = \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_C\}$ over the S-dimensional real space R^S is the set of C cluster centers and d_{ij} is a dissimilarity measure where the Euclidean distance between $\mathbf{x}_j$ and $\mathbf{a}_i$ is generally used. The update equations of μ_{ij} and $\mathbf{a}_i$ are

$$\mu_{ij} = \frac{d^2(\mathbf{x}_j, \mathbf{a}_i)^{-1/m-1}}{\sum\limits_{t=1}^{c} d^2(\mathbf{x}_j, \mathbf{a}_t)^{-1/m-1}} \tag{2}$$

$$\mathbf{a}_i = \sum_{j=1}^{N} \mu_{ij}^{m}\mathbf{x}_j \Big/ \sum_{j=1}^{N} \mu_{ij}^{m} \tag{3}$$

Although FCM and its variant of extensions are useful methods in clustering, their memberships do not always correspond well to the membership degrees of data points to classes and may be inaccurate in a noisy environment [8]. To improve this weakness of FCM, and to produce memberships that provide a better explanation for the memberships of a data point across classes, Krishnapuram and Keller [8] relaxed the constraint $\sum_{i=1}^{C} \mu_{ij} = 1$ in FCM and used the possibilistic C-memberships $\mathbf{U}_P$. To avoid trivial solutions for minimizing the FCM objective function J_{FCM} as relaxing the constraint $\sum_{i=1}^{C} \mu_{ij} = 1$, Krishnapuram and Keller [8] added the constraint term $\sum_{i=1}^{C} \eta_i \sum_{j=1}^{N} (1 - \mu_{ij})^m$ to $J_{FCM}(\mathbf{U}, \mathbf{A})$ and then proposed the PCM objective function J_{PCM} as follows:

$$J_{PCM}(\mathbf{U}, \mathbf{A}) = \sum_{i=1}^{C}\sum_{j=1}^{N} \mu_{ij}^{m} d^2(\mathbf{x}_j, \mathbf{a}_i) + \sum_{i=1}^{C} \eta_i \sum_{j=1}^{N} (1 - \mu_{ij})^m \tag{4}$$

where $m \in [1, \infty)$, $\eta_i > 0$ for all i, $\mu_{ij} \in [0, 1]$ for all i and j, and $\max\limits_{i} \mu_{ij} > 0$ for all j. The update equations of μ_{ij} and $\mathbf{a}_i$ are as follows

$$\mu_{ij} = \frac{1}{1 + \left(d^2(\mathbf{x}_j, \mathbf{a}_i)/\eta_i\right)^{1/m-1}} \tag{5}$$

$$\mathbf{a}_i = \sum_{j=1}^{N} \mu_{ij}^{m}\mathbf{x}_j \Big/ \sum_{j=1}^{N} \mu_{ij}^{m} \tag{6}$$

To solve the selection of the parameter η_i, Yang and Wu [14] proposed the possibilistic clustering algorithm (PCA). The PCA objective function is given with

$$J_{PCA}(\mathbf{U}, \mathbf{A}) = \sum_{i=1}^{C}\sum_{j=1}^{N} \mu_{ij}^{m} d^2(\mathbf{x}_j, \mathbf{a}_i) + \frac{\beta}{m^2\sqrt{C}} \sum_{i=1}^{C}\sum_{j=1}^{N} \left(\mu_{ij}^{m} \ln \mu_{ij}^{m} - \mu_{ij}^{m}\right) \tag{7}$$

where $\beta = \sum_{j=1}^{N} d^2(\mathbf{x}_j, \bar{\mathbf{x}}) \Big/ N$ and $\bar{\mathbf{x}} = \sum_{j=1}^{N} \mathbf{x}_j \Big/ N$. Yang and Wu [14] gave various values of m and used validity indexes to find an optimal cluster number C.The update equations of μ_{ij} and $\mathbf{a}_i$ can be rewritten as

$$\mu_{ij} = \left(\exp\left(-\frac{d^2(\mathbf{x}_j, \mathbf{a}_i)}{\beta}\right)\right)^{m\sqrt{C}} \tag{8}$$

$$\mathbf{a}_i = \frac{\sum_{j=1}^{N}\left(\exp\left(-d^2(\mathbf{x}_j, \mathbf{a}_i)/\beta\right)\right)^{m^2\sqrt{C}} \mathbf{x}_j}{\sum_{j=1}^{N}\left(\exp\left(-d^2(\mathbf{x}_j, \mathbf{a}_i)/\beta\right)\right)^{m^2\sqrt{C}}} \tag{9}$$

If m and C are given a priori in Eqs. (8) and (9), we can obtain the PCA algorithm.

III. THE PROPOSED FULLY-UNSUPERVISED POSSIBILISTIC C-MEANS CLUSTERING ALGORITHM

In this section, we propose a fully-unsupervised PCM (FU-PCM) clustering algorithm. First, we construct a generalized framework for PCM clustering. This generalized framework can be a generalization of most PCM algorithms.

A. GENERALIZED FRAMEWORK OF PCM CLUSTERING

Since the term $\sum_{i=1}^{C} \eta_i \sum_{j=1}^{N} (1-\mu_{ij})^m$ in the PCM objective function (4) is a function of μ_{ij}, we may replace it with a more general function $P(\mathbf{U})$ of μ_{ij}. Therefore, we first propose a generalized PCM (GPCM) objective function as follows:

$$J_{GPCM}(\mathbf{U}, \mathbf{A}) = \sum_{i=1}^{C} \sum_{j=1}^{N} \mu_{ij}^m d^2(\mathbf{x}_j, \mathbf{a}_i) + P(\mathbf{U}) \tag{10}$$

where $P(\mathbf{U}) = \sum_{i=1}^{C} \eta_i \sum_{j=1}^{N} p(\mu_{ij})$, $\eta_i > 0$ for all i, and $p(\mu_{ij})$ is a general function of μ_{ij}. By setting $\frac{\partial}{\partial \mu_{ij}} J_{GPCM}(\mathbf{U}, \mathbf{A}) = 0$, we obtain the equation with $m\mu_{ij}^{m-1} d^2(\mathbf{x}_j, \mathbf{a}_i) + \eta_i p'(\mu_{ij}) = 0$. Thus, we have the membership update equation with

$$\mu_{ij}^{m-1} = -\eta_i p'(\mu_{ij}) \Big/ m d^2(\mathbf{x}_j, \mathbf{a}_i) \tag{11}$$

In Eq. (11), we examine the following three properties:

(1) Since m, η_i, μ_{ij}, and $d^2(\mathbf{x}_j, \mathbf{a}_i)$ are nonnegative, then $p'(\mu_{ij}) \le 0$. That is, the function $p(\mu_{ij})$ of μ_{ij} must be a decreasing function.

(2) Since $\mu_{ij} \in [0, 1]$, then a suitable η_i must be related to distances between data points and cluster centers. Further, the parameter η_i should be also related to m.

(3) The degree of membership μ_{ij} becomes larger when the distance between the data point $\mathbf{x}_j$ and the cluster center $\mathbf{a}_i$ becomes smaller.

When we take a solution for $d^2(\mathbf{x}_j, \mathbf{a}_i)$ in terms of μ_{ij} according to Eq. (11), we obtain

$$d^2(\mathbf{x}_j, \mathbf{a}_i) = -\frac{\eta_i p'(\mu_{ij})}{m\mu_{ij}^{m-1}} \tag{12}$$

If we embed Eq. (12) into the GPCM objective function J_{GPCM}, we can obtain another GPCM objective function J_{GPCM} only through μ_{ij} with

$$\begin{aligned} J_{GPCM}(\mathbf{U}) &= \sum_{i=1}^{C}\sum_{j=1}^{N} -\left[\frac{\mu_{ij}}{m}\eta_i p'(\mu_{ij})\right] + \sum_{i=1}^{C}\eta_i \sum_{j=1}^{N} p(\mu_{ij}) \\ &= -\sum_{i=1}^{C}\eta_i \sum_{j=1}^{N}\left(\frac{\mu_{ij}}{m}p'(\mu_{ij}) - p(\mu_{ij})\right). \end{aligned}$$

If we consider a fixed η_i for all i, then minimizing $J_{GPCM}(\mathbf{U})$ over $\mathbf{U}$ is equivalent to maximizing the following objective function J_{GPCM}^* over $\mathbf{U}$ with

$$J_{GPCM}^*(\mathbf{U}) = \sum_{i=1}^{C}\sum_{j=1}^{N}\left(\frac{\mu_{ij}}{m}p'(\mu_{ij}) - p(\mu_{ij})\right) \tag{13}$$

By following the spirit of PCM that has decreasing influences of those data points being far away from the cluster center, we set

$$J^*(\mathbf{U}) = \sum_{i=1}^{C}\sum_{j=1}^{N} \mu_{ij}^{\alpha} \tag{14}$$

where $\alpha \ge 1$ is a fuzziness index like m. It is seen, in Eq. (14), if α becomes larger, then large memberships will become more important and the influences of those small memberships will become weaker when maximizing Eq. (14). Corresponding to Eqs. (13) and (14), we get the following equation:

$$\frac{\mu_{ij}}{m}p'(\mu_{ij}) - p(\mu_{ij}) = \mu_{ij}^{\alpha} \tag{15}$$

Replacing μ_{ij} with x, we can rewrite Eq. (15) as $p'(x) - \frac{m}{x}p(x) = mx^{\alpha-1}$, so it becomes a first-order differential equation. We next solve the differential equation. By the integration factor $\exp(\int -\frac{m}{x}dx) = e^{\ln x^{-m}+c} = c_0 x^{-m}$, we can obtain a general solution for the differential equation with $p(x) = \frac{\int c_0 x^{-m} m x^{\alpha-1} dx}{c_0 x^{-m}} = \frac{\int m x^{\alpha-m-1} dx}{x^{-m}}$. Thus, we get the following two cases:

(1) If $\alpha = m$, then

$$\begin{aligned} p(x) &= x^m \int m x^{-1} dx = x^m (m \ln x + c_1) \\ &= m x^m \ln x + c_1 x^m. \end{aligned}$$

(2) If $\alpha \ne m$, then

$$\begin{aligned} p(x) &= x^m \int m x^{\alpha-m-1} dx = x^m \left(m\frac{x^{\alpha-m}}{\alpha-m} + c_2\right) \\ &= m x^m \frac{x^{\alpha-m}}{\alpha-m} + c_2 x^m \end{aligned}$$

Therefore, we have

$$p'(x) = \begin{cases} m^2 x^{m-1}\ln x + m x^{m-1} + c_1 m x^{m-1}, & \text{if } \alpha = m \\ \dfrac{\alpha}{\alpha-m} m x^{\alpha-1} + c_2 m x^{m-1}, & \text{if } \alpha \ne m \end{cases}$$

TABLE 1. Relations between the GPCM framework and various PCM algorithms.

<table>
<tr><td colspan="3">$J_{GPCM}(\mathbf{U},\mathbf{A})=\sum_{i=1}^{C}\sum_{j=1}^{N}\mu_{ij}^{m}d^2(\mathbf{x}_j,\mathbf{a}_i)+\sum_{i=1}^{C}\eta_i\sum_{j=1}^{N}p(\mu_{ij})$ with $p(\mu_{ij})=\begin{cases}\mu_{ij}^{m}\ln\mu_{ij}^{m}-\mu_{ij}^{m} & ,\ \text{if } \alpha=m\\ \dfrac{m\mu_{ij}^{m}(\mu_{ij}^{\alpha-m}-1)}{\alpha-m} & ,\ \text{if } \alpha\neq m\end{cases}$</td></tr>
<tr><td>Various PCM algorithms</td><td>Objective functions</td><td>Special cases in the GPCM framework</td></tr>
<tr><td>PCM1 (Krishnapuram and Keller [8])</td><td>$\sum_{i=1}^{C}\sum_{j=1}^{N}\mu_{ij}^{m}d^2(\mathbf{x}_j,\mathbf{a}_i)+\sum_{i=1}^{C}\eta_i\sum_{j=1}^{N}(1-\mu_{ij})^m\ ,\ m=2$</td><td>$\alpha=1\ ,\ m=2\ ,\ p(\mu_{ij})=(1-\mu_{ij})^2$</td></tr>
<tr><td>PCM2 (Krishnapuram and Keller [10])</td><td>$\sum_{i=1}^{C}\sum_{j=1}^{N}\mu_{ij}d^2(\mathbf{x}_j,\mathbf{a}_i)+\sum_{i=1}^{C}\eta_i\sum_{j=1}^{N}(\mu_{ij}\ln\mu_{ij}-\mu_{ij})$</td><td>$\alpha=1\ ,\ m=1\ ,$
$p(\mu_{ij})=\mu_{ij}\ln\mu_{ij}-\mu_{ij}$</td></tr>
<tr><td>PCA
(Yang and Wu [14])</td><td>$\sum_{i=1}^{C}\sum_{j=1}^{N}\mu_{ij}^{m}d^2(\mathbf{x}_j,\mathbf{a}_i)+\frac{\beta}{m^2\sqrt{C}}\sum_{i=1}^{C}\sum_{j=1}^{N}\left(\mu_{ij}^{m}\ln\mu_{ij}^{m}-\mu_{ij}^{m}\right)$</td><td>$\alpha=m\ ,\ \eta_i=\beta/m^2\sqrt{C}\ ,$
$p(\mu_{ij})=\mu_{ij}^{m}\ln\mu_{ij}^{m}-\mu_{ij}^{m}$</td></tr>
<tr><td>AMPCM
(Yang and Lai [20])</td><td>$\sum_{i=1}^{C}\sum_{j=1}^{N}\mu_{ij}d^2(\mathbf{x}_j,\mathbf{a}_i)-r\sum_{i=1}^{C}\sum_{j=1}^{N}\mu_{ij}\left(1-\frac{k}{k+1}\mu_{ij}^{1/k}\right),$
$k>1$</td><td>$\alpha-m=1/k\ ,\ m=1\ ,$
$\eta_i=r(\alpha-m)/\alpha\ ,$
$p(\mu_{ij})=k\mu_{ij}(\mu_{ij}^{1/k}-1)-\mu_{ij}$</td></tr>
<tr><td>APCM
(Xenaki et al. [21])</td><td>$\sum_{i=1}^{C}\sum_{j=1}^{N}\mu_{ij}d^2(\mathbf{x}_j,\mathbf{a}_i)+\frac{\lambda}{\alpha_A}\sum_{i=1}^{C}\eta_i\sum_{j=1}^{N}(\mu_{ij}\ln\mu_{ij}-\mu_{ij})$</td><td>$\alpha=1\ ,\ m=1\ ,\ \eta_i=(\hat{\lambda}/\alpha_A)\lambda_i\ ,$
$p(\mu_{ij})=\mu_{ij}\ln\mu_{ij}-\mu_{ij}$</td></tr>
</table>

Because of $0<x=\mu_{ij}\leq 1$, we have $p'(x)\leq 0$. Thus, we obtain, if $\alpha=m$ then $m\ln x+1+c_1\leq 0$ implies $c_1\leq -1-m\ln x$; and if $\alpha\neq m$, then $\frac{\alpha}{\alpha-m}+c_2x^{m-\alpha}\leq 0$ implies $c_2\leq -\frac{\alpha x^{\alpha-m}}{\alpha-m}$. For $0<x=\mu_{ij}\leq 1$, we have, if $\alpha=m$ then $c_1\leq -1$, and if $\alpha\neq m$, then $c_2\leq -\frac{\alpha}{\alpha-m}$. Simply let $c_1=-1$ and $c_2=-\alpha/(\alpha-m)$. We finally obtain the following equation:

$$p(\mu_{ij})=\begin{cases}\mu_{ij}^{m}\ln\mu_{ij}^{m}-\mu_{ij}^{m}, & \text{if } \alpha=m\\ \dfrac{m\mu_{ij}^{m}(\mu_{ij}^{\alpha-m}-1)}{\alpha-m}, & \text{if } \alpha\neq m\end{cases}\tag{16}$$

with $m\geq 1,\alpha\geq 1$.

We combine the GPCM with Eq. (16). Thus, we propose a generalized framework of PCM with the generalized objective function $J_{GPCM}(\mathbf{U},\mathbf{A})$ of Eq. (10) with following updating equations of μ_{ij}:

$$\text{If } \alpha=m,\quad \mu_{ij}=\exp(-\frac{d^2(\mathbf{x}_j,\mathbf{a}_i)}{m\eta_i})\ \text{(See Theorem 1)}\tag{17}$$

If $\alpha\neq m$,

$$\mu_{ij}=\begin{cases}0, & \text{if } d^2(\mathbf{x}_j,\mathbf{a}_i)>\dfrac{\alpha}{\alpha-m}\eta_i\\ \left[1-\dfrac{\alpha-m}{\alpha}.\dfrac{d^2(\mathbf{x}_j,\mathbf{a}_i)}{\eta_i}\right]^{\frac{1}{\alpha-m}}, & \\ & \text{if } d^2(\mathbf{x}_j,\mathbf{a}_i)\leq\dfrac{\alpha}{\alpha-m}\eta_i\end{cases}\tag{18}$$

According to derived Eqs. (10), (17) and (18), Table 1 summarize the relations between the GPCM framework and these extended PCM algorithms. These are PCM1 [8], PCM2 [10], PCA [14], AMPCM [20], and APCM [21].

B. FULLY-UNSUPERVISED POSSIBILISTIC C-MEANS

In the previous subsection, we had constructed the generalized PCM (GPCM) framework. We also showed most existing extensions of PCM in the literature are special cases of the generalized PCM objective function. One of them is PCA proposed by Yang and Wu [14]. The PCA algorithm is presented below.

Algorithm 1 PCA Algorithm

```
Step 1. Fix $m \geq 1, 2 \leq C \leq N$, and any $\varepsilon > 0$. Give initials
        $\mathbf{A}^{(0)} = \{\mathbf{a}_1^{(0)}, \mathbf{a}_2^{(0)}, \ldots, \mathbf{a}_C^{(0)}\}$ and let the iteration
        counter $t = 0$.
Step 2. Compute $\mathbf{U}^{(t)}$ by Eq. (8).
Step 3. Update $\mathbf{A}^{(t+1)}$ by Eq. (9).
    IF $\|\mathbf{A}^{(t+1)} - \mathbf{A}^{(t)}\| < \varepsilon$ (convenient matrix norm),
    THEN stop; ELSE set $t = t + 1$ and RETURN to Step 2.
```

As mentioned in Section II, PCA can solve the drawback of PCM that depend on η_i and m which heavily influence the clustering performance. We next give an example to demonstrate it.

Example 1: The data set with 400 data points, as shown in Fig. 1(a), is generated from a two-component Gaussian mixture distribution where one component has the proportion 0.75, mean $(5,0)^T$, and covariance matrix $\begin{pmatrix}0.16 & 0\\ 0 & 0.25\end{pmatrix}$; Another component has the proportion 0.25, means $(5,7)^T$, and covariance matrix $\begin{pmatrix}2.56 & 0\\ 0 & 4\end{pmatrix}$. Fixing $m=2$ and $C=2$ we implement the PCA Algorithm for the data set using different initial cluster centers. We obtain clustering results

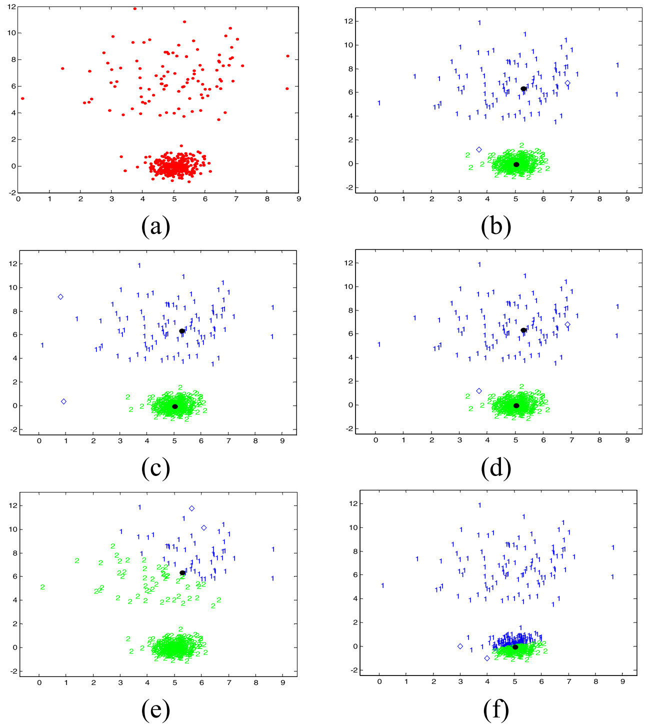


FIGURE 1. **(a) Data set; (b)-(f) Clustering results of PCA Algorithm using different initial cluster centers.**

with two different clusters, as shown in Figs. 1(b)~1(f), where final cluster centers are denoted by symbol "●", and initial cluster centers are denoted by symbol "◇". Clearly, different initial cluster centers actually affect clustering results. In Figs. 1(b) and 1(c), one initial cluster center is near the larger spread cluster and another is near the smaller spread cluster so we obtain good clustering results. Figs. 1(d) and 1(e) show two initial cluster centers are near the larger spread cluster, and so two final cluster centers are agglomerated together. Fig. 1(f) shows two initial cluster centers are near the smaller spread cluster, and therefore two final cluster centers are agglomerated together near the smaller spread cluster. The drawbacks of the PCA Algorithm are: (i) It is always affected by initial cluster centers; (ii) To a certain extent it depends on m and C. Thus, we need to solve these two drawbacks. We next solve the first problem that is an initialization problem.

We mentioned the PCA in Yang and Wu [14] used various selection values of m and validity indexes to find an optimal cluster number C. Obviously, it is necessary to perform PCA repeatedly with different values of m and C. In fact, we may consider a more simple way to determine the number of clusters. Example 1 demonstrated two final cluster centers will be agglomerated together if the two initial cluster centers are near the same cluster. This gives us a hint to find the number of clusters. Our method is described as follows: (i) If we set all data points as initial cluster centers, then we can precisely solve initialization problem; (ii) Most data sets with N data points usually have at most $\sqrt{N}$ clusters, i.e., in general, $C \leq \sqrt{N}$. Thus, we assign $C = \sqrt{N}$ in Eqs. (8) and (9), and so we have Eqs. (19) and (20) as follows:

$$\mu_{ij} = \left(\exp\left(-d^2(\mathbf{x}_j, \mathbf{a}_i)/\beta\right)\right)^{m\sqrt[4]{N}} \tag{19}$$

$$\mathbf{a}_i = \frac{\sum_{j=1}^{N} \left(\exp\left(-d^2(\mathbf{x}_j, \mathbf{a}_i)/\beta\right)\right)^{m^2\sqrt[4]{N}} \mathbf{x}_j}{\sum_{j=1}^{N} \left(\exp\left(-d^2(\mathbf{x}_j, \mathbf{a}_i)/\beta\right)\right)^{m^2\sqrt[4]{N}}} \tag{20}$$

According to the above discussion, we improve the PCA Algorithm, termed the Robust-PCA Algorithm, and summarize the algorithm as follows:

Algorithm 2 Robust-PCA Algorithm

```
Step 1. Fix $m \geq 1$ and $\varepsilon > 0$. Give initial $\mathbf{A}^{(0)} = \mathbf{X}$ and let
        iteration counter $t = 0$.
Step 2. Compute $\mathbf{U}^{(t)}$ by Eq. (17).
Step 3. Update $\mathbf{A}^{(t+1)}$ by Eq. (18).
    IF $\left\|\mathbf{A}^{(t+1)} - \mathbf{A}^{(t)}\right\| < \varepsilon$ (convenient matrix norm),
    THEN stop; ELSE set $t = t + 1$ and RETURN to Step 2.
```

Example 1 (Continued): Fix $m = 1.05$, we implement the Robust-PCA Algorithm for the data set in Fig. 1(a). The states of the cluster centers after different iterations are shown in Figs. 2(a)-2(e). We can see that all cluster centers are centralized to two locations. This means that a proper number of clusters is 2. In fact, if values of m are between 1 and 1.3,

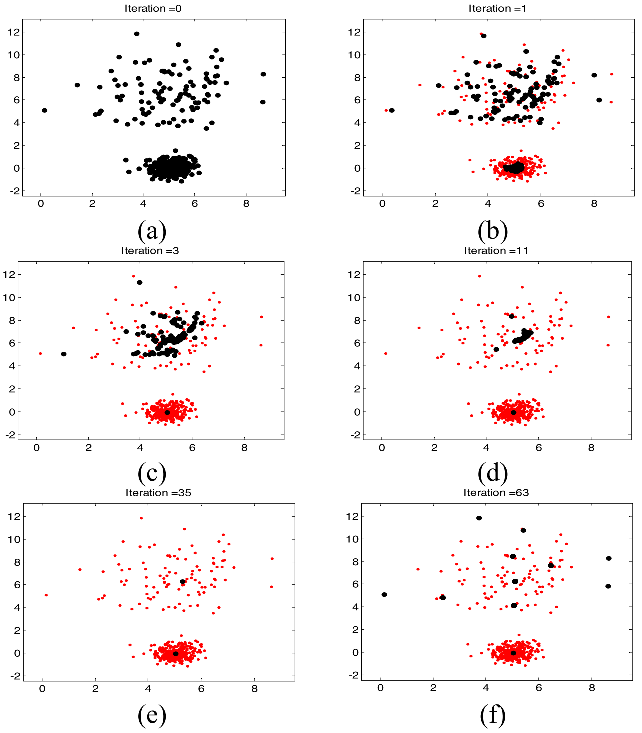


FIGURE 2. **(a)-(e): Cluster centers with *m* = 1.3 after 0, 1, 3, 11, and 35 iterations, respectively; (f) Final cluster centers with *m* = 2 after 45 iterations.**

then we also obtain the same result. Fig. 2(f) shows the results when $m = 2$ where the final states of cluster centers are centralized to eleven locations with an incorrect number of clusters. Thus, different values of m actually influence the search for the number of clusters, and so we need to solve the problem for selection of m. We solve this problem in the next section.

The remaining problem is how to estimate m. We first consider a parameter γ with $\gamma = m^2\sqrt{C}$. Thus, we have $m = \sqrt{\gamma/\sqrt{C}}$. In the Robust-PCA Algorithm, we substituted $C = \sqrt{N}$. On the other hand, we know the fuzziness index m has a restriction with $m \geq 1$. Therefore, we obtain formula for m as

$$m = \max\left\{\sqrt{\gamma/\sqrt[4]{N}}, 1\right\} \qquad (21)$$

The parameter γ influences the number of clusters and also the clustering results. We should focus on finding a good estimate for γ, and then consider an estimate for m according to γ. We next illustrate the influence of γ and discuss how to choose γ by following Example 2.

Example 2: In this example, we consider a data set with 1000 data points generated from a three-component Gaussian mixture distribution, as shown in Fig. 3(a). We implement the Robust-PCA Algorithm for the data set and take different γ values with $\gamma = 1, 5, 10, 20$, respectively. We analyze the number of clusters and clustering results. We find all cluster centers are merged to one point when $\gamma = 1$, as shown in Fig. 3(b). Therefore, we obtain the cluster number of $C^* = 1$ with only one cluster, as shown in Fig. 3(c). Fig. 3(d) shows all cluster centers are merged to three points when $\gamma = 5$. This indicates the cluster number of $C^* = 3$ and obtains three clusters, as shown in Fig. 3(e). If we have $\gamma = 10$, then all cluster centers are also merged to three points, as shown in Fig. 3(f), and its clustering results with three clusters are shown in Fig. 3(g). This means the values of γ from 5 to 10 could obtain good clustering results. However, if we have $\gamma = 20$, then all cluster centers are merged to seven different points, as shown in Fig. 3(h), with bad clustering results, as shown in Fig. 3(i). This means too large value of γ will lead to an incorrect cluster number of $C^* = 7$ and also bad clustering results.

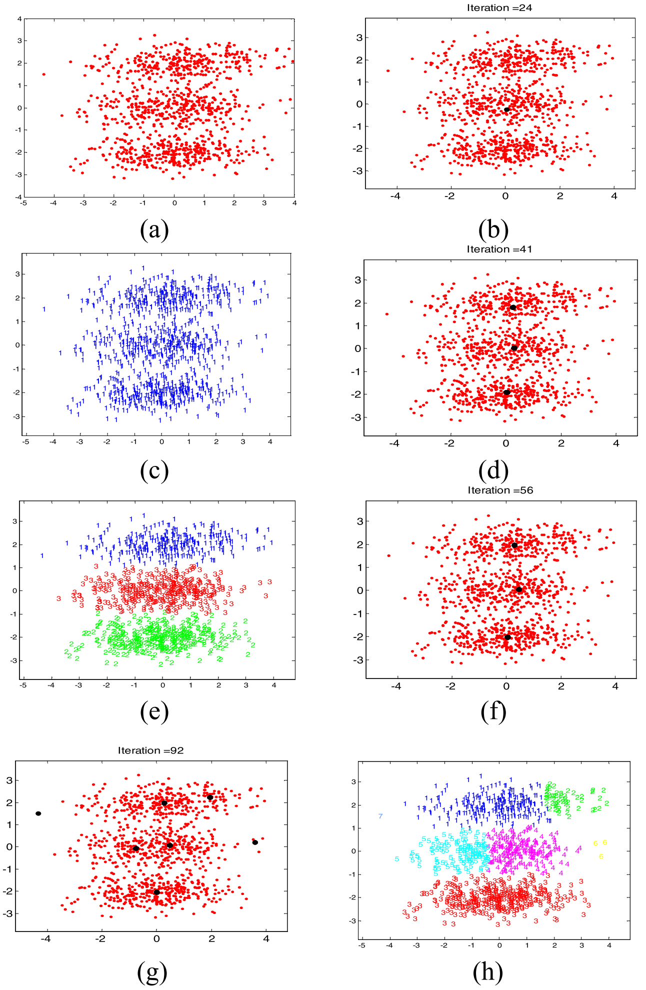
The above demonstration shows different γ values may lead to different cluster numbers and clustering results. In fact, the parameter γ is related to the mountain function with

$$f_s(\mathbf{x}_i) = \sum_{j=1}^{N}\left(\exp(-d^2(\mathbf{x}_j, \mathbf{x}_i)/\beta)\right)^{\gamma}, \quad i = 1, 2, \ldots, N \qquad (22)$$

The parameter γ can precisely control the number of peaks of the mountain function f_s. We next use the three-component data set to analyze this phenomenon. This mountain function f_s denotes the mountain height on data points. It is similar to the sum of $\mu_{ij}^{(0)}$. We employ f_s to show the mountain function on the three-component data set with $\gamma = 1, 5, 10, 20$. Fig. 4(a) shows f_s with $\gamma = 1$ has only one peak. This is the reason all cluster centers are merged to one point for Robust-PCA Algorithm when $\gamma = 1$. Figs. 4(b) and 4(c) show f_s with $\gamma = 5$ and 10 has three peaks. In Figs. 3(d) and 3(f), the Robust-PCA Algorithm with $\gamma = 5$ and 10 obtains an identical cluster number of $C^* = 3$. Finally, Fig. 4(d) shows f_s has more than three peaks when $\gamma = 20$. We also see the Robust-PCA Algorithm with $\gamma = 20$ clusters the three-component data set into seven clusters, as shown in Fig. 3(i).

According to our experiments, we can see the parameter γ greatly effects the performance of the Robust-PCA Algorithm. We next give some properties to explain how the parameter γ influences the cluster centers derived from Eq. (9).

Property 1: When γ tends to zero, all cluster centers are centralized to the sample mean.



(a) (b) (c) (d) (e) (f) (g) (h)

FIGURE 3. (a) Three-component data set; (b)-(c): Final cluster centers and clustering results when $\gamma = 1$; (d)-(e): Final cluster centers and clustering results when $\gamma = 5$; (f)-(g): Final cluster centers and clustering results when $\gamma = 10$; (h)-(i): Final cluster centers and clustering results when $\gamma = 20$.

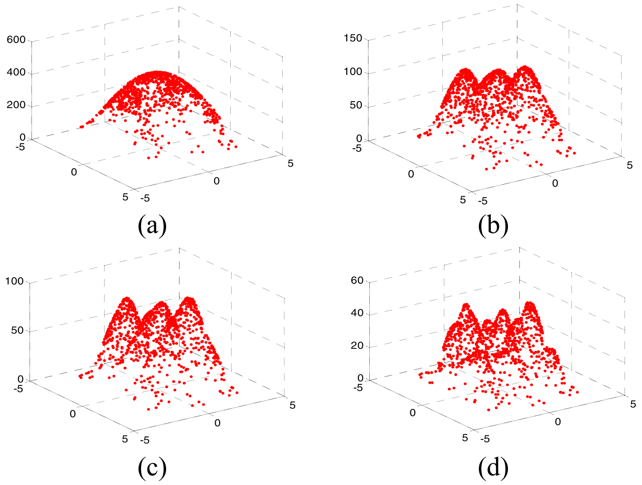


FIGURE 4. Mountain function on the three-component data set; (a): $\gamma = 1$; (b): $\gamma = 5$; (c): $\gamma = 10$; (d): $\gamma = 20$.

Proof:

$$\lim_{\gamma\to 0}\mathbf{a}_i = \lim_{\gamma\to 0}\sum_{j=1}^{N}\left(e^{-\frac{d^2(\mathbf{x}_j,\mathbf{a}_i)}{\beta}}\right)^{\gamma}\mathbf{x}_j \Big/ \sum_{j=1}^{N}\left(e^{-\frac{d^2(\mathbf{x}_j,\mathbf{a}_i)}{\beta}}\right)^{\gamma}$$

$$= \lim_{\gamma\to 0}\sum_{j=1}^{N}\mathbf{x}_j/N = \bar{\mathbf{x}}.$$

■

Note that Property 1 tell us that, as $\gamma \to 0$, the data set has only one cluster, and so the cluster center is the sample mean.

Property 2: When γ tends to infinity, all cluster centers are individual data points.

Proof: Let $E = \max_j \exp(-d^2(\mathbf{x}_j, \mathbf{a}_i)/\beta)$ and $E'_j = \exp\left(-\frac{d^2(\mathbf{x}_j,\mathbf{a}_i)}{\beta}\right)\Big/E$. Then, we have that

$$\lim_{\gamma\to\infty}\mathbf{a}_i = \lim_{\gamma\to\infty}\sum_{j=1}^{N}\left(e^{-\frac{d^2(\mathbf{x}_j,\mathbf{a}_i)}{\beta}}\right)^{\gamma}\mathbf{x}_j \Big/ \sum_{j=1}^{N}\left(e^{-\frac{d^2(\mathbf{x}_j,\mathbf{a}_i)}{\beta}}\right)^{\gamma}$$

$$= \lim_{\gamma\to\infty}\frac{\sum_{j=1}^{N}\left(E'_j\right)^{\gamma}\mathbf{x}_j}{\sum_{j=1}^{N}\left(E'_j\right)^{\gamma}} = \frac{\sum_{E'_j=1}\mathbf{x}_j}{\sum_{E'_j=1}1}.$$

This means that, as $\gamma \to \infty$, all cluster centers are individual data points. ■

From the above analysis, we know finding a proper estimated value for γ is important. Compare Fig. 4(b) with Fig. 4(c), we observe the mountain functions with $\gamma = 5$ and $\gamma = 10$ have similar shapes with almost the same peaks. In fact, the shapes of the mountain functions with $\gamma = 5 \sim 10$ are similar. This means the values of correlation of the mountain function with these different γ values are high. Thus, we can use the value to estimate γ such that it produces stable clustering results. From this point of view, we consider the correlation comparison algorithm (CCA) proposed by Yang and Wu [17] to estimate the parameter γ. Yang and Wu [17] suggested a threshold ε_{CCA} of CCA around 0.97~0.999. We use $\varepsilon_{CCA} = 0.97$ as the threshold in this paper. The CCA is summarized as follows:

Algorithm 3 Correlation Comparison Algorithm (CCA)

```
Step 1. Give $l = 1$ and $\varepsilon_{CCA} = 0.97$.
Step 2. Compute the correlation values of $f_s(\mathbf{x}_i)$, $i = 1, 2, \ldots, N$ with pair $(\gamma = 5l, \gamma = 5(l+1))$.
Step 3. IF the correlation value greater than or equal to $\varepsilon_{CCA}$, THEN $\gamma = 5l$; ELSE $l = l + 1$ and GOTO Step 2.
```

In Example 1 (continued), we can obtain $\gamma = 5$ by the CCA. Therefore, $m = \sqrt{5/\sqrt[4]{400}} = 1.0574$. This estimate of m can have the Robust-PCA Algorithm obtain a correct number of clusters. In Example 2, we find both of $\gamma = 5$ and $\gamma = 10$ have good clustering results. We obtain $\gamma = 10$ by the CCA. Since the three-component data set contains 1000 data points, and so $m = \sqrt{10/\sqrt[4]{1000}} = 1.33$. Now we had solved the drawbacks of the PCA Algorithm. We next propose a fully-unsupervised PCM algorithm.

According to analysis mentioned earlier, we give a novel PCM algorithm that is exactly a fully-unsupervised clustering algorithm, called a fully-unsupervised PCM (FU-PCM) algorithm. We summarize the FU-PCM algorithm as follows:

Algorithm 4 FU-PCM Clustering Algorithm

```
Step 1. Call CCA to obtain $\gamma$.
Step 2. Estimate $m$ by Eq. (21).
Step 3. Fix $\varepsilon_1 > 0$ and $0 < \varepsilon_2 < 1$. Give $\mathbf{A}^{(0)} = \mathbf{X}$. Let iteration counter $t = 0$.
Step 4. Compute $\mathbf{U}^{(t)}$ by Eq. (19).
Step 5. Update $\mathbf{A}^{(t+1)}$ by Eq. (20).
        IF $\left\|\mathbf{A}^{(t+1)} - \mathbf{A}^{(t)}\right\| < \varepsilon_1$, THEN go to Step 6,
        ELSE set $t = t + 1$ and RETURN to Step 4.
Step 6. Construct the matrix $D$, with $D_{ij}$ as the distance of pairs $\mathbf{a}_i^{(t+1)}$ and $\mathbf{a}_j^{(t+1)}$ in the set $\mathbf{A}^{(t+1)}$ of cluster centers, where $i, j = 1, \ldots, C$.
Step 7. For all $i$, find $T_i = \left\{\bigcup \mathbf{a}_j^{(t+1)} \middle| D_{ij} < \varepsilon_2, \forall j\right\}$ and assign all elements of column $j$ in the matrix $D$ with 1.
Step 8. Let $I = \{i \,|\, T_i \neq \emptyset\}$, then compute $C^* = |I|$ and $\mathbf{a}_i^* = \mathrm{mean}(T_i)$, for $i \in I$. Reorder the index $i$ of the new set $\{\mathbf{a}_i^*, i \in I\}$ to be $\{1, \ldots, C^*\}$.
Step 9. Assign all points to their cluster by computing the minimum distance of points and cluster centers, with $Cluster_i = \left\{\mathbf{x}_j \middle| \left\|\mathbf{x}_j - \mathbf{a}_i^*\right\|^2 \leq \left\|\mathbf{x}_j - \mathbf{a}_p^*\right\|^2, \forall j, 1 \leq p \leq C^*\right\}$, $i = 1, \ldots, C^*$.
```

We know the PCM algorithm has a tendency to produce coincident clusters that can be a good mode-seeking algorithm. However, the performance of PCM heavily depends on the parameter η_i. Some research has attempted to find an appropriate η_i in the literature. In the proposed GPCM, we set $\eta_i = \beta/m^2\sqrt{C}$ for finding a good estimated value of γ We have shown the parameter γ becomes another parameter to affect the performance of the GPCM, and so we use the CCA to find a good estimated value for γ. That is, we use another schema for finding the parameter γ instead of finding an appropriate η_i, and we finally propose the GPCM algorithm such that it becomes a fully-unsupervised clustering algorithm, called the fully-unsupervised PCM (FU-PCM) algorithm.

We next analyze the computational complexity of the FU-PCM algorithm. In CCA, we need to compute the value of the mountain function on data points in S-dimensional space, and so its computational complexity is $O(N^2St_1)$ where t_1 denotes the number of iterations in CCA. In Step 3, we set all points as initial cluster centers. The computational complexity in each iteration for computing $\mathbf{U}$ and $\mathbf{A}$ are $O(N^2S)$ and $O(NS)$. If t_2 denotes the number of iterations in Steps 4 and 5, then the computational complexity is $O((N^2S + NS)t_2)$.

Finally, we analyze the robustness of the generalized PCM algorithm to noise and outliers. Let $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ be a data set and θ be an estimated parameter. A location M-estimator is given by M-estimator $= \arg\min_{\theta} \sum_{j=1}^{N} \rho(\mathbf{x}_j - \theta)$, where ρ is an arbitrary loss measure function. Then, the M-estimator can be obtained by solving the equation $\frac{d}{d\theta} \sum_{j=1}^{N} \rho(\mathbf{x}_j - \theta) = \sum_{j=1}^{N} \varphi(\mathbf{x}_j - \theta) = 0, \varphi(\mathbf{x}_j - \theta) = \frac{d}{d\theta} \sum_{j=1}^{N} \rho(\mathbf{x}_j - \theta)$. The influence function (IF) of the M-estimator is given by $IF(x; F, \theta) = \frac{\varphi(x-\theta)}{\int \varphi(x-\theta)dF(x)}$, where $F(x)$ is the distribution function of X. The M-estimator has shown IF is proportional to its φ function. If the influence function of an estimator is unbounded, an outlier might cause trouble where the φ function is used to denote the degree of influence. The cluster center $\mathbf{a}_i$ in Eq. (9) is a location M-estimator with a loss measure function given by $\rho(\mathbf{x}_j - \mathbf{a}_i) = 1 - (e^{-\frac{d^2(\mathbf{x}_j, \mathbf{a}_i)}{\beta}})^{\gamma}$, with $d(\mathbf{x}_j, \mathbf{a}_i) = \left\|\mathbf{x}_j - \mathbf{a}_i\right\|$. Then, $\varphi(\mathbf{x}_j - \mathbf{a}_i) = \frac{d}{d\mathbf{a}_i}\rho(\mathbf{x}_j - \mathbf{a}_i) = \frac{2\gamma}{\beta}(\mathbf{a}_i - \mathbf{x}_j)(e^{-\frac{d^2(\mathbf{x}_j, \mathbf{a}_i)}{\beta}})^{\gamma}$. By applying the L'Hospital's rule, we can derive $\lim_{\mathbf{x}_j \to \pm\infty} \varphi(\mathbf{x}_j - \mathbf{a}_i) = 0$. We can also find the maximum influence by solving $\frac{d}{d\mathbf{x}_j}\varphi(\mathbf{x}_j - \mathbf{a}_i) = 0$. From the above discussion, we can see the φ function is bounded. Since the influence function is proportional to its φ function, the influence function is also bounded. We also have $IF(\mathbf{x}_j; F; \mathbf{a}_i)$ being equal to zero when $\mathbf{x}_j$ tends to infinity or negative infinity. This means the influence of an extremely large or small $\mathbf{x}_j$ on the cluster center $\mathbf{a}_i$ is very small. For example, by considering $\mathbf{a}_i = 2.5$, $\beta = 1$, and $\gamma = 1, 3, 5$, we can demonstrate the graphs of the φ functions, as shown in Fig. 5. From Fig. 5, we can see the results: (i) The influence of $\mathbf{x}_j$ becomes very small when $d^2(\mathbf{x}_j, \mathbf{a}_i) > \beta$; (ii) The influence of an outlier becomes small when γ is large.

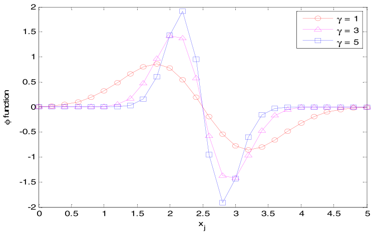


FIGURE 5. Graphs of φ functions.

IV. COMPARISONS AND EXPERIMENTAL RESULTS

In this section, we use some numerical and real data sets to illustrate the performance of the FU-PCM algorithm. We also compare the proposed FU-PCM algorithm with some existing algorithms, such as k-means, FCM, PCM, PCM2 [10], PCA [14], AMPCM [20], and APCM [21]. Since the k-means, FCM, PCM, PCM2, and PCA algorithms need to use validity indices to find the number C^* of clusters, we consider the validity indices of the partition coefficient (PC) [22], partition entropy (PE) [23], and Xie and Beni (XB) [24]. The PC, PE, and XB are defined as follows: $PC(C) = \frac{1}{N}\sum_{i=1}^{C}\sum_{j=1}^{N}\mu_{ij}^2$, $PE(C) = -\frac{1}{N}\sum_{i=1}^{C}\sum_{j=1}^{N}\mu_{ij}\ln\mu_{ij}$, $XB(C) = \frac{\sum_{i=1}^{C}\sum_{j=1}^{N}\mu_{ij}^{m}\left\|\mathbf{x}_j - \mathbf{a}_i\right\|^2}{N\min_{i,j}\left\|\mathbf{a}_i - \mathbf{a}_j\right\|^2}$. We find the optimal C^* by solving $\max_{2 \le C \le N-1} PC(C)$, $\min_{2 \le C \le N-1} PE(C)$, and $\min_{2 \le C \le N-1} XB(C)$, respectively. However, the possibilistic c-membership μ_{ij} does not satisfy the condition $\sum_{i=1}^{C}\mu_{ij} = 1$. Yang and Wu [14] normalized μ_{ij} to be μ'_{ij} by $\mu'_{ij} = \mu_{ij} \Big/ \sum_{i=1}^{c}\mu_{ij}$, and then generalized PC, PE, and XB to be GPC, GPE, and GXB for PCA by replacing μ_{ij} with μ'_{ij}, respectively. Finally, we apply the FU-PCM to image segmentation.

Example 3: In this example, we first consider an artificial data set, as shown in Fig. 6(a), where there are two circle-shape clusters, three strip-shape clusters, and one ellipse-shape cluster. We implement the FU-PCM for the data set. From the CCA results, we obtain $\gamma = 15$. The data set contains 4480 data points, and so the estimated value of m is 1.35. Figs. 6(b)-6(d) are states of cluster centers after 11, 43, and 149 iterations, respectively. The FU-PCM algorithm gives the final clustering results with the number of clusters $C^* = 6$ as shown in Fig. 6(e). We also implement the FCM and PCA algorithms for this data set. To obtain the number C^* of clusters for FCM, we consider validity indices PC, PE, XB, and for PCA, we use GPC, GPE, and GXB. We perform FCM and PCA with different fuzziness $m = 1.5, 2,$ and 2.5. We find these validity indices for FCM and PCA with $m = 1.5$, 2, and 2.5 obtain $C^* = 2$, but only the XB index indicates the cluster number $C^* = 6$ when FCM with $m = 1.5$ and $m = 2$. By assigning the cluster number $C = 6$ and giving the same initial cluster centers where denoted by symbol "$\diamond$",

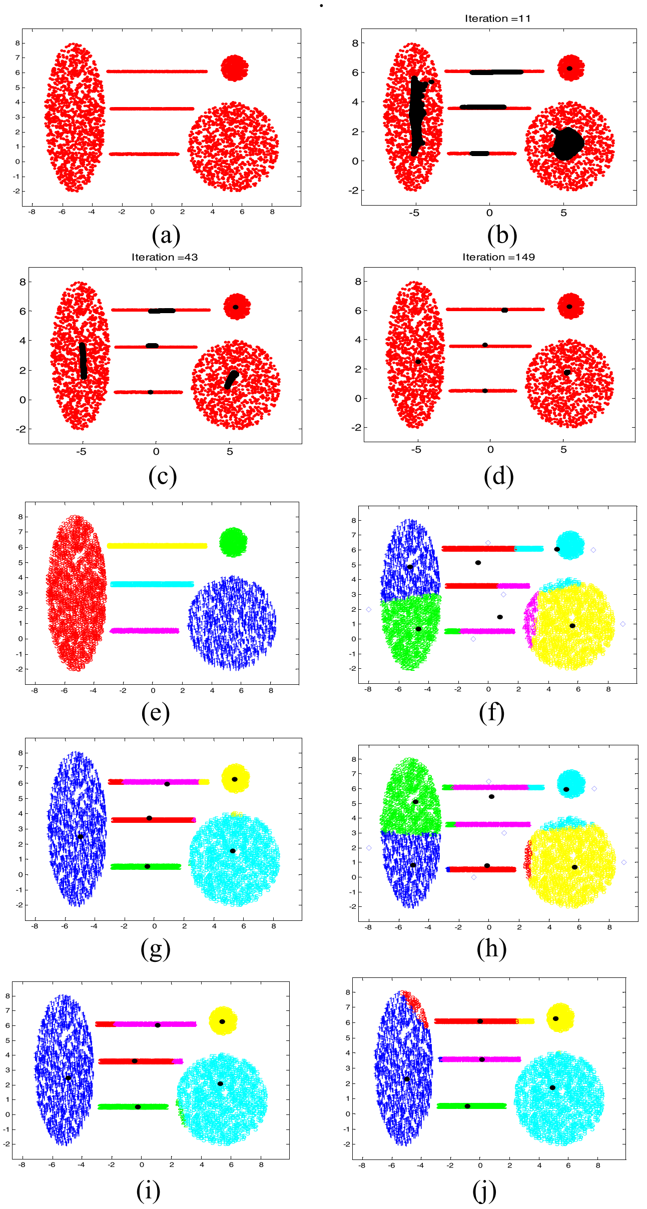


FIGURE 6. **(a) Data set; (b)-(d): States of cluster centers of FUPCM after 11, 43, and 149 iterations, respectively; (e): Final clustering results of FU-PCM with $C^* = 6$; (f): Clustering results of FCM with $m = 2$; (g): Clustering results of PCA with $m = 2$; (h): Clustering results of k-means; (i): Clustering results of PCM2; (j): Clustering results of APCM with $\alpha_A = 1$ and $C_{initial} = 18$.**

the clustering results of FCM ($m = 2$) and PCA ($m = 2$) are shown in Fig. 6(f) and Fig. 6(g), respectively. We find FCM ($m = 2$) cannot obtain correct clustering results. PCA have better clustering results, but they are not good enough. We also implement k-means and PCM2 with initial cluster centers "◇". The clustering results of k-means and PCM2 are shown in Figs. 6(h) and 6(i), respectively. We can see k-means has bad clustering results and the clustering results from FCM and PCM2 are not good. Further, we also consider the AMPCM and APCM algorithms. The clustering results from the AMPCM are the same as the FU-PCM, as shown in Fig. 6(e). The AMPCM can find $C^* = 6$ and obtain correct clustering results. For the APCM proposed by Xenaki *et al.* [21], they suggested the parameter α_A is chosen from the interval [1, 3] and initial cluster number $C_{initial}$ is 3 or 4 times of the correct cluster number for the APCM. Therefore, we implement the APCM with $C_{initial} = 18$ and $\alpha_A = 1$ for the data set in Fig. 6(a). The clustering results from the APCM are shown in Fig. 6(j). The APCM finds the final cluster number is $C^* = 6$, but clustering results are not good as these of the FU-PCM and AMPCM.

Example 4: In this example, we consider a three-dimensional data set, as shown in Fig. 7(a). This data set consists of 200 data points where 100 data points are generated from the plane $z = 0$, and another 100 data points are generated from the plane $z = 0.4$. The distance between two consecutive data points is 0.1. The mean of 100 "×" data points is (0.55, 0.55, 0), and another mean of 100 "○" data points is (0.55, 0.55, 0.40). The CCA algorithm gives $\gamma = 5$. The $m = 1.15$ is obtained. Fig. 7(b) shows the final states of the cluster centers of FU-PCM. All clusters are centralized

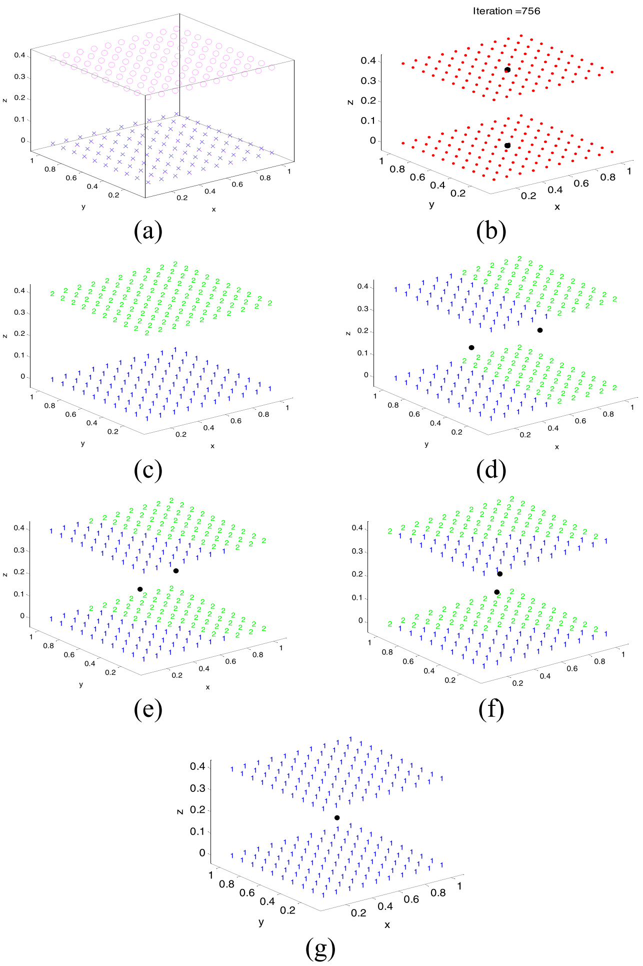


FIGURE 7. **(a) Data set; (b): Final states of cluster centers of FUPCM; (c): Clustering results of FU-PCM; (d)-(f): Clustering results of FCM with $m = 1.5, 2, 2.5$, respectively; (g): Clustering results of AMPCM.**

TABLE 2. Final cluster centers from different algorithms.

FU-PCM	FCM ($m = 1.5$)	FCM ($m = 2$)	FCM ($m = 2.5$)	PCA ($m = 1.5$)
(0.5500, 0.5500, 0.0095) (0.5500, 0.5500, 0.3905)	(0.3127, 0.5489, 0.2000) (0.7873, 0.5511, 0.2000)	(0.3631, 0.4810, 0.2000) (0.7369, 0.6190, 0.2000)	(0.4483, 0.4298, 0.2000) (0.6517, 0.6702, 0.2000)	(0.5483, 0.5522, 0.0595) (0.5517, 0.5478, 0.3405)
PCA ($m = 2$)	k-means	PCM2	AMPCM	APCM ($\alpha_A = 3,\ C_{initial} = 6$)
(0.5816, 0.5871, 0.0054) (0.5184, 0.5129, 0.3946)	(0.3000, 0.5500, 0.2000) (0.8000, 0.5500, 0.2000)	(0.6080, 0.5265, 0.0000) (0.4920, 0.5735, 0.4000)	(0.5500, 0.5500, 0.2000)	(0.5620, 0.5695, 0.0192) (0.5473, 0.5452, 0.3717)

TABLE 3. Average accuracy rates of FU-PCM, K-means, FCM, PCM2, PCA, and AMPCM.

Clustering methods	FU-PCM	k-means	FCM ($m = 2$)	PCM2	PCA ($m = 2$)	AMPCM
Accuracy rate	0.9048	0.8905	0.8952	0.8952	0.8905	0.8714

in (0.55, 0.55, 0.0095) and (0.55, 0.55, 0.3905), where there are very closed to planes $z = 0$ and $z = 0.4$, respectively. Fig. 7(c) shows the clustering results of FU-PCM which are well fitted to the data structure. Validity indices PC and PE find $C^* = 2$. Validity indices GPC, GPE, and GXB also find $C^* = 2$ for PCA when $m = 1.5$ or $m = 2$. The clustering results of PCA with $m = 1.5$ or $m = 2$ are the same as Fig. 7(c). The clustering results of FCM with $m = 1.5$, $m = 2$, and $m = 2.5$ are shown in Figs. 7(d)-7(f), respectively. Clearly, they do not fit well. The final cluster centers are shown in Table 2. Table 2 shows the cluster centers of FCM are laid on plane $z = 0.2$. FCM cannot perform well. The cluster centers of PCA when $m = 2$ are closer to the original planes. PCA with $m = 2$ performs better than PCA with $m = 1.5$. The cluster results of k-means are the same as Fig. 7(d). The cluster centers of k-means are also laid on plane $z = 0.2$ like FCM. The cluster results of PCM2 are the same as Fig. 7(c). Table 2 shows that the cluster centers of PCM2 are on the original planes. PCM2 performs very well for this data set. Fig. 7(e) shows the clustering results of AMPCM. AMPCM finds $C^* = 1$ for this data set. The cluster center of AMPCM is (0.55, 0.55, 0.20). The clustering results of APCM with $\alpha_A = 3$ and $C_{initial} = 6$ are the same as Fig. 7(c). APCM find $C^* = 2$ for this data set. Table 2 shows the final cluster centers of APCM are close to the original planes, like FU-PCM and PCA.

Example 5: In this example, we use a data set with 11 blocks which are consisted of 4900 data points and 1000 randomly noisy points (denoted "+" sign), as shown in Fig. 8(a). The CCA algorithm gives $\gamma = 15$ for the data set. Figs. 8(b)-(e) are states of cluster centers by running FU-PCM after 1, 11, 31, and 179 iterations, respectively. The FU-PCM algorithm converges after 179 iterations with $C^* = 11$ for the data set. The identified eleven clusters are shown in Fig. 8(f). We can see the influence of noisy points on FU-PCM is very small. Figs. 8(g)-(l) are plots of validity indices. Note, the large values and small values of GXB are quite different. We manifest charts by using a nature logarithm function. The results show the validity indices cannot find a correct cluster number C^*. By assigning $C = 11$ and giving good initial cluster centers, PCA with $m = 2.5$ obtains better clustering results, as shown in Fig. 8(m). The clustering results of AMPCM are shown in Fig. 8(k). AMPCM finds $C^* = 11$ and obtains good clustering results. We consider $C_{initial} = 33$ and α_A from 1 to 3 for APCM, but APCM cannot find $C^* = 11$. Hence, we implement APCM with $C_{initial} = 33$ and $\alpha_A = 0.5$. The cluster number $C^* = 11$ is obtained, but clustering results are inadequate, as shown in Fig. 8(o).

Example 6: In this example, we consider the seeds data set from the UCI Machine Learning Repository [25]. The seeds data set contains three kinds of wheat kernels (Kama, Rosa, and Canadian). Each kind has 70 samples. Each data point was measured by seven characteristics of wheat kernels: area, parameter, compactness, length of kernel, width of the kernel, asymmetry, and length of kernel groove. By implementing FU-PCM, we obtain $\gamma = 10$ and final cluster number $C^* = 3$ which is fitted to the three kinds of wheat kernels. All validity indices of PC, PE, XB, GPC, GPE, and GXB for FCM and PCA indicate $C^* = 2$. By implementing AMPCM, we obtain the number $C^* = 3$ of clusters. By implementing APCM with $C_{initial} = 9$ and $\alpha_A = 1$, it also obtains the number $C^* = 3$ of clusters. Note, the FU-PCM and AMPCM set all data points as initial cluster centers, and so if we implement FU-PCM or AMPCM 100 times for the data set, then 100 consistent clustering results with the same cluster number C^* are obtained. However, APCM uses the results of FCM as initial values. We know that the results of FCM are affected by the initial values. It induces APCM to obtain different clustering results and a different number C^* of clusters for different initials. Thus, we implement APCM with 100 different initials for the data set, and there are about 35% to obtain $C^* = 3$. Further, by assigning the number $C = 3$ of clusters to k-means, FCM, PCM2, and PCA, we run each algorithm 100 times and compute the average accuracy rates, as shown in Table 3.

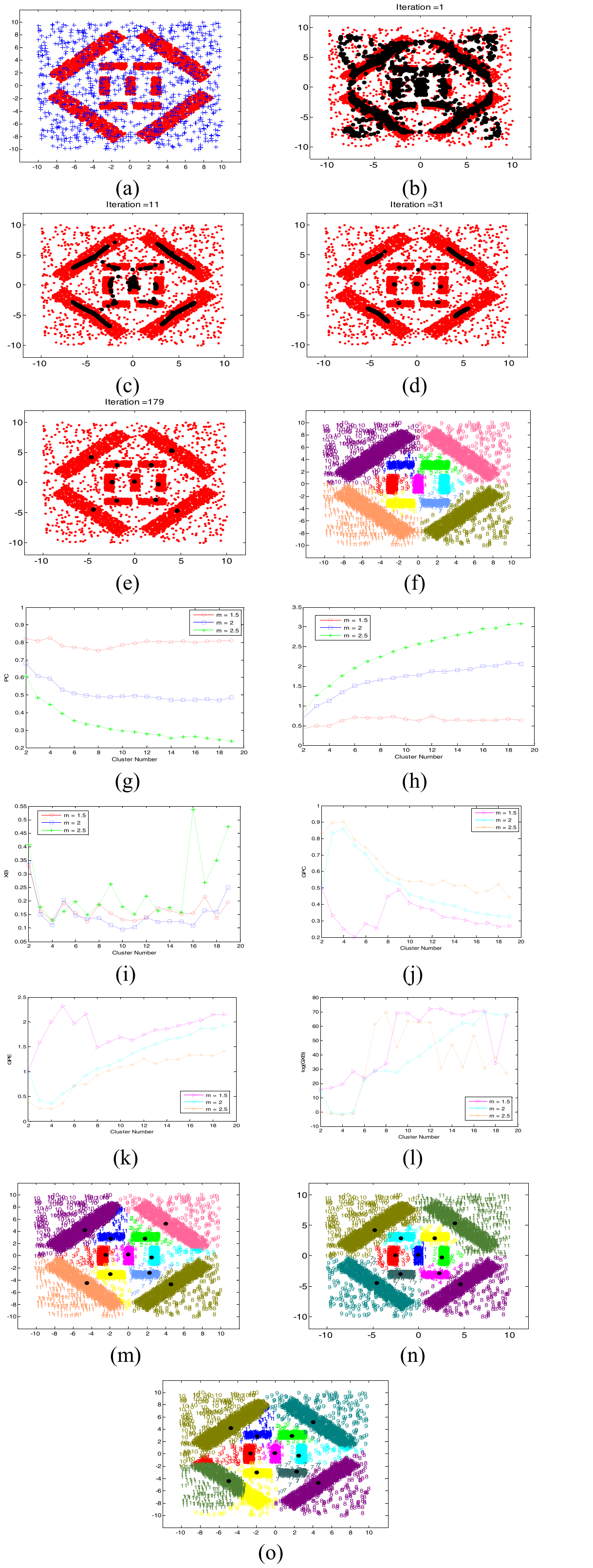


FIGURE 8. (a) Data set; (b)-(e): States of cluster centers of FUPCM after 1, 11, 31, and 179 iterations, respectively; (f): Clustering results of FU-PCM; (g)-(l): Plots of PC, PE, XB, GPC, GPE, and GXB indices, respectively; (m): Clustering results of PCA ($m = 2.5$); (n): Clustering results of AMPCM; (o): Clustering results of APCM ($\alpha_A = 0.5$, $C_{initial} = 33$).

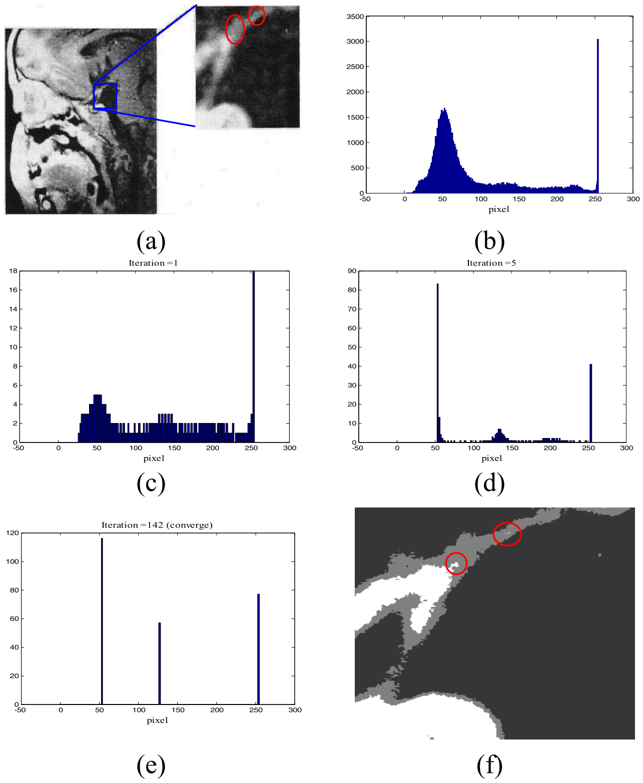


FIGURE 9. (a) Original MRI and its window selection; (b) Histogram of window segmentation image; (c)-(e) Locations of cluster centers after iterations 1, 5, and 142, respectively; (f) Segmentation result of FU-PCM.

We find FU-PCM actually has the best average accuracy rate among these algorithms.

Example 7: In this example, we consider the Iris data set from the UCI Machine Learning Repository [25]. The Iris data set contains 150 data points with three clusters (setosa, versicolor, and virginica), where each cluster has 50 points. As we know, two clusters in this data set overlap. Thus, it is prevalent to get $C^* = 2$ if the cluster number is not given. By implementing FU-PCM and AMPCM, they can obtain $C^* = 2$ without any parameter and initialization assignment. By using APCM, from 90 experiments with different initializations and parameters, there are about 20% to obtain $C^* = 3$. The other cluster numbers obtained by APCM are $C^* = 1, 2, 4, 5, 6, 10, 11$.

In the next example, we apply the FU-PCM algorithm to image segmentation. Image data usually consists of large pixel data points. Since we set all data points as initial cluster centers in FU-PCM, it may take more time to perform FU-PCM on the image data set. Therefore, we set up a technique to reduce the computational time. The technique is stated as follows. Suppose that $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ is a large data set where each data point only takes values on $\{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_M\}$. Let $n_1, n_2, \ldots, n_M$ be the data set $\mathbf{X}$ on corresponding counts of $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_M$, respectively. Then the update equation of the cluster center $\mathbf{a}_i$ in Eq. (20) can be rewritten as

$$\mathbf{a}_i = \sum_{j=1}^{M} \left(e^{-\frac{d^2(\mathbf{y}_j,\mathbf{a}_i)}{\beta}} \right)^{m^2 \sqrt[4]{N}} n_j \mathbf{y}_j \Bigg/ \sum_{j=1}^{M} \left(e^{-\frac{d^2(\mathbf{y}_j,\mathbf{a}_i)}{\beta}} \right)^{m^2 \sqrt[4]{N}} n_j \tag{23}$$

Therefore, we can set $\{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_M\}$ as initial clusters and replace Eq. (20) by Eq. (23) in the FU-PCM algorithm for large data sets.

Example 8: This example applies the FU-PCM algorithm to MRI segmentation [26], as shown in Fig. 9(a). The MRI data set is from 2-years old female patient. She was diagnosed with retinoblastoma of her left eye, an inborn malignant neoplasm of the retina with frequent metastasis beyond the lacrimal cribrosa. The window segmentation in Fig. 9(a) contains three kinds of tissue (connective tissue, nervous tissue, and tumor tissue (red circles)). This image is processed at $283 \times 292 = 82636$ pixels and takes the maximum gray level with 255 and the minimum gray level with 5. Fig. 9(b) shows the histogram of the window segmentation image. Two peaks are more obvious in Fig. 9(b). The CCA for window segmentation image obtains $\gamma = 15$ and we compute $m = 1.94$. Figs. 9(c)-(e) are the locations of the cluster centers following iterations 1, 5, and 142, respectively. The FU-PCM finds the final cluster number $C^* = 3$ after 142 iterations. The segmentation results of the FU-PCM are shown in Fig. 9(f). Clearly, FU-PCM can detect tumor tissue (red circles).

V. CONCLUSIONS

In this paper, we propose a generalized framework for possibilistic c-means (GPCM) clustering. Based on the GPCM framework and the merit of PCM with a tendency to produce coincident clusters, we gave a novel PCM clustering algorithm, called FU-PCM. The proposed FU-PCM algorithm could automatically estimate parameters and also generate an optimal number of clusters. The advantages of the FU-PCM are robust to initial values, cluster numbers and different cluster volumes and shapes. Several examples with numerical and real data sets demonstrated the effectiveness and superiority of the proposed FU-PCM algorithm. Comparisons of the FU-PCM algorithm with k-means, FCM, PCM, PCA, AMPCM, and APCM were made. The FU-PCM outperforms these algorithms according to these results. Further, the FU-PCM also shows good performance for segmentation of images. Overall, the proposed FU-PCM algorithm is actually effective and useful as a fully-unsupervised clustering algorithm. Our future work will continue developing more novel PCM algorithms based on the GPCM framework, such as the feature-weighting GPCM with a feature-reduction procedure or sparse-regularization feature selection based on L1-norm penalty terms.

REFERENCES

[1] L. A. Zadeh, "Fuzzy sets," *Inf. Control*, vol. 8, no. 3, pp. 338–353, 1965.

[2] J. C. Bezdek, *Pattern Recognition With Fuzzy Objective Function Algorithms*. New York, NY, USA: Plenum Press, 1981.

[3] M. S. Yang, "A survey of fuzzy clustering," *Math. Comput. Model.*, vol. 18, no. 11, pp. 1–16, 1993.

[4] A. Baraldi and P. Blonda, "A survey of fuzzy clustering algorithms for pattern recognition. I," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 29, no. 6, pp. 778–785, Dec. 1999.

[5] F. Höppner, F. Klawonn, R. Kruse, and T. Runkler, *Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition*. New York, NY, USA: Wiley, 1999.

[6] J. C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," *J. Cybern.*, vol. 3, no. 3, pp. 32–57, 1973.

[7] M.-S. Yang and Y. Nataliani, "A feature-reduction fuzzy clustering algorithm based on feature-weighted entropy," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 2, pp. 817–835, Apr. 2018.

[8] R. Krishnapuram and J. M. Keller, "A possibilistic approach to clustering," *IEEE Trans. Fuzzy Syst.*, vol. 1, no. 2, pp. 98–110, May 1993.

[9] M. Barni, V. Cappellini, and A. Mecocci, "Comments on 'a possibilistic approach to clustering,'" *IEEE Trans. Fuzzy Syst.*, vol. 4, no. 3, pp. 393–396, Aug. 1996.

[10] R. Krishnapuram and J. M. Keller, "The possibilistic C-means algorithm: Insights and recommendations," *IEEE Trans. Fuzzy Syst.*, vol. 4, no. 3, pp. 385–393, Aug. 1996.

[11] J.-S. Zhang and Y.-W. Leung, "Improved possibilistic C-means clustering algorithms," *IEEE Trans. Fuzzy Syst.*, vol. 12, no. 2, pp. 209–217, Apr. 2004.

[12] H. Timm, C. Borgelt, C. Döring, and R. Kruse, "An extension to possibilistic fuzzy cluster analysis," *Fuzzy Sets Syst.*, vol. 147, no. 1, pp. 3–16, 2004.

[13] N. R. Pal, K. Pal, J. M. Keller, and J. C. Bezdek, "A possibilistic fuzzy C-means clustering algorithm," *IEEE Trans. Fuzzy Syst.*, vol. 13, no. 4, pp. 517–530, Aug. 2005.

[14] M.-S. Yang and K.-L. Wu, "Unsupervised possibilistic clustering," *Pattern Recognit.*, vol. 39, no. 1, pp. 5–21, 2006.

[15] V. S. Tseng and C.-P. Kao, "A novel similarity-based fuzzy clustering algorithm by integrating PCM and mountain method," *IEEE Trans. Fuzzy Syst.*, vol. 15, no. 6, pp. 1188–1196, Dec. 2007.

[16] R. R. Yager and D. P. Filev, "Approximate clustering via the mountain method," *IEEE Trans. Syst., Man, Cybern.*, vol. 24, no. 8, pp. 1279–1284, Aug. 1994.

[17] M.-S. Yang and K.-L. Wu, "A similarity-based robust clustering method," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 4, pp. 434–448, Apr. 2004.

[18] M. Filippone, F. Masulli, and S. Rovetta, "Applying the possibilistic c-means algorithm in kernel-induced spaces," *IEEE Trans. Fuzzy Syst.*, vol. 18, no. 3, pp. 572–584, Jun. 2010.

[19] S.-T. Chang, K.-P. Lu, and M.-S. Yang, "Stepwise possibilistic c-regressions," *Inf. Sci.*, vol. 334–335, pp. 307–322, Mar. 2016.

[20] M.-S. Yang and C.-Y. Lai, "A robust automatic merging possibilistic clustering method," *IEEE Trans. Fuzzy Syst.*, vol. 19, no. 1, pp. 26–41, Feb. 2011.

[21] S. D. Xenaki, K. D. Koutroumbas, and A. A. Rontogiannis, "A novel adaptive possibilistic clustering algorithm," *IEEE Trans. Fuzzy Syst.*, vol. 24, no. 4, pp. 791–810, Aug. 2016.

[22] J. C. Bezdek, "Cluster validity with fuzzy sets," *J. Cybern.*, vol. 3, no. 3, pp. 58–73, 1974.

[23] J. C. Bezdek, "Numerical taxonomy with fuzzy sets," *J. Math. Biol.*, vol. 1, no. 1, pp. 57–71, 1974.

[24] X. L. Xie and G. Beni, "A validity measure for fuzzy clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 8, pp. 841–847, Aug. 1991.

[25] C. L. Blake and C. J. Merz. (1998). *UCI Repository of Machine Learning Databases, A Huge Collection of Artificial and Real-World Data Sets*. [Online]. Available: http://archive.ics.uci.edu/ml/datasets.html

[26] M.-S. Yang, Y.-J. Hu, K. C. Lin, and C. C. Lin, "Segmentation techniques for tissue differentiation in MRI of ophthalmology using fuzzy clustering algorithms," *Magn. Reson. Imag.*, vol. 20, no. 2, pp. 173–179, 2002.

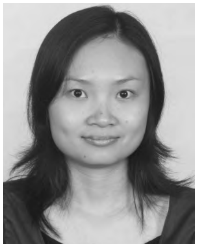
MIIN-SHEN YANG received the B.S. degree in mathematics from Chung Yuan Christian University (CYCU), Chungli, Taiwan, in 1977, the M.S. degree in applied mathematics from National Chiao-Tung University, Hsinchu, Taiwan, in 1980, and the Ph.D. degree in statistics from the University of South Carolina, Columbia, USA, in 1989.

In 1989, he joined the faculty of the Department of Mathematics, CYCU, as an Associate Professor, where he has been a Professor since 1994. From 1997 to 1998, he was a Visiting Professor with the Department of Industrial Engineering, University of Washington, Seattle, USA. From 2001 to 2005, he was the Chairman of the Department of Applied Mathematics, CYCU, where he has been a Distinguished Professor with the Department of Applied Mathematics and the Director of Chaplain's Office since 2012. His research interests include applications of statistics, control charts, fuzzy clustering, pattern recognition, and machine learning.

Dr. Yang was an Associate Editor of the IEEE Transactions on Fuzzy Systems from 2005 to 2011. He is an Associate Editor of the *Applied Computational Intelligence and Soft Computing* and Editor-in-Chief of *Advances in Computational Research*.

SHOU-JEN CHANG-CHIEN received the B.S., M.S., and Ph.D. degrees in applied mathematics from Chung Yuan Christian University, Chungli, Taiwan, in 2000, 2002, and 2011, respectively.

He is currently a Post-Doctoral Fellow with the Department of Applied Mathematics, Chung Yuan Christian University. His research interests include cluster analysis and pattern recognition.



YESSICA NATALIANI received the B.S. degree in mathematics and the M.S. degree in computer science from Gadjah Mada University, Yogyakarta, Indonesia, in 2004 and 2006, respectively. She is currently pursuing the Ph.D. degree with the Department of Applied Mathematics, Chung Yuan Christian University, Taiwan. Her research interests include cluster analysis and pattern recognition.