

Article

Revisiting Possibilistic Fuzzy C-Means Clustering Using the Majorization-Minimization Method

Yuxue Chen and Shuisheng Zhou * 

School of Mathematics and Statistics, Xidian University, Xi'an 710071, China; chyuxue@hotmail.com

* Correspondence: sszhou@mail.xidian.edu.cn

Abstract: Possibilistic fuzzy c-means (PFCM) clustering is a kind of hybrid clustering method based on fuzzy c-means (FCM) and possibilistic c-means (PCM), which not only has the stability of FCM but also partly inherits the robustness of PCM. However, as an extension of FCM on the objective function, PFCM tends to find a suboptimal local minimum, which affects its performance. In this paper, we rederive PFCM using the majorization-minimization (MM) method, which is a new derivation approach not seen in other studies. In addition, we propose an effective optimization method to solve the above problem, called MMPFCM. Firstly, by eliminating the variable $V \in R^{p \times c}$, the original optimization problem is transformed into a simplified model with fewer variables but a proportional term. Therefore, we introduce a new intermediate variable $s \in R^c$ to convert the model with the proportional term into an easily solvable equivalent form. Subsequently, we design an iterative sub-problem using the MM method. The complexity analysis indicates that MMPFCM and PFCM share the same computational complexity. However, MMPFCM requires less memory per iteration. Extensive experiments, including objective function value comparison and clustering performance comparison, demonstrate that MMPFCM converges to a better local minimum compared to PFCM.

Keywords: possibilistic fuzzy c-means; fuzzy c-means; majorization-minimization; local minimum



Citation: Chen, Y.; Zhou, S. Revisiting Possibilistic Fuzzy C-Means Clustering Using the Majorization-Minimization Method. *Entropy* **2024**, *26*, 670. <https://doi.org/10.3390/e26080670>

Academic Editor: Boris Ryabko

Received: 22 May 2024

Revised: 3 August 2024

Accepted: 4 August 2024

Published: 6 August 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Cluster analysis is one of the most important topics in machine learning [1], and it has been widely applied in various fields, such as data mining [2], image processing [3], and pattern recognition [4]. Clustering algorithms are a type of unsupervised learning method, aiming to partition datasets into multiple clusters based on similarity measures, ensuring that samples within the same cluster are similar [5,6].

Generally, clustering algorithms are divided into hard and soft clustering schemes [7]. The c-means algorithm [8] is the most widely used hard clustering method due to its speed and simplicity. However, it is sensitive to initial points and often falls short of finding an optimal solution. Xu et al. [9] explored an alternative to the c-means algorithm that retains its simplicity while mitigating its tendency to become trapped in local minima, called power k-means. The optimization objective of this method is a power mean function of the distances between sample points and cluster centers, leading to its non-convex and high-dimensional characteristics, which makes it difficult to solve directly. Therefore, the majorization-minimization (MM) method is used to derive a descent scheme. In this paper, the MM method will be employed for the first time to solve the possibilistic fuzzy c-means clustering problem.

As one of the most typical soft clustering methods, fuzzy c-means (FCM) [10,11] is a fuzzy extension method of c-means, which divides n data points into c clusters by the membership grade matrix U . Compared to c-means, FCM achieves better clustering performance on datasets with overlapped clusters. Due to its flexibility and robustness, FCM has been favored by researchers up to now. However, it also suffers from some drawbacks. The first is that this method is sensitive to initialization and may easily become trapped in

local minima [12,13]. The iteratively re-weighted algorithm for FCM (IRWFCM) [14] was proposed to find better local minima by using the iteratively re-weighted method to solve an equivalent minimization problem of the original FCM problem. FCM with the entropy regularization method (ERFCM) [15] introduces a maximum entropy term to FCM, whose purpose is trading off fuzziness and compactness. Recently, the iteratively re-weighted method was also used to optimize the equivalent problem of ERFCM in [16]. IRWFCM and IRWERFCM both have smaller objective function values and better clustering performance with fewer iterations compared to the original optimization method.

Furthermore, FCM assigns memberships to x_i based on the relative distance from x_i to the c cluster centers. In noisy data environments, this can result in noise points that are far from the c clusters being assigned relatively high memberships, thereby not accurately reflecting the true clustering results [17]. To overcome this shortcoming, the possibilistic c -means (PCM) method was proposed by Krishnapuram and Keller [18], which relaxes the constraint $\sum_{j=1}^c u_{ij} = 1$ ($i = 1, \dots, n$) so that u_{ij} better reflects what we expect for the typicality of the i -th sample to the j -th cluster. However, PCM still suffers from initialization sensitivity and is prone to obtaining a coincident clustering phenomenon [19]. Therefore, Pal et al. [20] proposed the possibilistic fuzzy c -means clustering (PFCM) algorithm, which is a kind of hybrid clustering method based on PCM and FCM, to solve the coincident clustering problem of PCM. This method not only has the stability of FCM but also partly inherits the robustness of PCM, making it widely researched and applied in many practical fields [21,22].

In contrast to traditional distance-based approaches like FCM, PCM, and PFCM, the generalized entropy-based PFCM (GEPFCM) algorithm [23] was proposed for clustering noisy data. GEPFCM utilizes distance-based functions to mitigate noise contributions on the cluster centers. An evaluation based on the distance between the actual and computed cluster centers reveals that the error of GEPFCM is approximately 80% lower than that of PFCM. Moreover, PFCM encounters challenges when dealing with feature-imbalanced multidimensional data, often leading to misclassification, as it treats all features equally. Additionally, in multiclass datasets with strong noise injection, PFCM tends to produce significant deviations and overlapping centers due to difficulties in setting membership-weight parameters and the absence of between-class relationships in possibilistic memberships. To address these issues, the feature-weighted suppressed PFCM clustering (FW-S-PFCM) [24] method was introduced. This method incorporates a feature-weighted approach and a “suppressed competitive learning” strategy into the PFCM model, enhancing its capability to handle these two types of challenging data scenarios. Interested readers can find more PFCM-type algorithms in [25–30], and the relevant citations therein.

While recent advancements in PFCM methods have demonstrated enhanced performance in various aspects, such as handling complex data, distance, hyperparameters, and robustness, they do not address the problem of whether the method finds a better local optimal solution when it converges. However, as an extension of FCM on the objective function, PFCM inevitably inherits the shortcomings of FCM, such as numerous local minimums and susceptibility to falling into poor local minima. Therefore, studying how to avoid PFCM falling into poor local minima is a worthwhile endeavor. In this study, we aim to tackle this issue by first establishing an equivalent formulation of the original PFCM problem and then designing an iterative sub-problem using the MM method. In addition, we also propose an alternative derivation method for PFCM and use the MM method to prove the equivalence of these two methods. The main contributions of this paper are summarized as follows:

1. We propose an alternative derivation method for PFCM, which begins by formulating an equivalent and simplified optimization problem, followed by solving it using the MM method. Finally, we demonstrate the equivalence of the new derivation method with PFCM.

2. Due to the presence of a proportional term in the derived simplified optimization problem, we further transform it into an easily solvable equivalent form by introducing a new intermediate variable s . Then, the MM method is employed to design an iterative sub-problem. We refer to this method as MMPFCM.
3. The complexity analysis indicates that MMPFCM and PFCM share the same computational complexity. However, MMPFCM utilizes the intermediate variable s of $c \times 1$ size instead of the variable V of $c \times p$ size to update U and T , resulting in smaller space complexity.
4. It is theoretically proven that when the inner loop of MMPFCM is executed only once, MMPFCM degenerates to the original PFCM method.
5. Experimental studies show that MMPFCM obtains better local minima compared to PFCM. In addition, compared with other state-of-the-art clustering methods, MMPFCM also shows its superiority.

The rest of this paper is organized as follows. Some preliminaries including notations, PFCM, and the MM method are given in Section 2. In Section 3, we rederive PFCM using the MM method, and another effective optimization method (MMPFCM) is presented in Section 4. A theoretical analysis is given in Section 5. In Section 6, the experimental results and discussions are reported. Section 7 concludes this paper.

2. Related Works

2.1. Notations

Let $X = [x_1, x_2, \dots, x_n] \in R^{p \times n}$ denote the data matrix, where $x_i \in R^p$ is the i -th sample for $i = 1, 2, \dots, n$, and x_i^T denotes the transpose of vector x_i . Let $V = [v_1, v_2, \dots, v_c] \in R^{p \times c}$ denote the cluster center matrix, where $v_j \in R^p$ is the centroid of the cluster C_j for $j = 1, 2, \dots, c$. The fuzzy membership matrix is denoted by $U = (u_{ij})_{n \times c}$, where u_{ij} represents the fuzzy membership degree of the i -th sample to the j -th cluster. The possibilistic membership matrix is denoted by $T = (t_{ij})_{n \times c}$, where t_{ij} represents the possibilistic membership degree of the i -th sample to the j -th cluster. u_j and t_j are the j -th columns of U and T , respectively. $u_j^{(k)}$ and $t_j^{(k)}$ denote u_j and t_j at the k -th iteration, respectively. In addition, e is a column vector with all elements equal to 1. $f(\theta|\theta_t)$ represents the function f evaluated at θ given the condition or parameter value θ_t .

2.2. Possibilistic Fuzzy C-Means Clustering

PFCM [20] is a kind of hybrid clustering method based on PCM and FCM, which contains both membership and typicality components. Its objective function is as follows:

$$\begin{aligned} \min_{(U, T, V)} J_{PFCM}(U, T, V) &= \sum_{i=1}^n \sum_{j=1}^c \left(au_{ij}^m + bt_{ij}^q \right) \|x_i - v_j\|_2^2 + \sum_{j=1}^c \gamma_j \sum_{i=1}^n (1 - t_{ij})^q, \\ \text{s.t. } \sum_{j=1}^c u_{ij} &= 1, 0 \leq u_{ij}, t_{ij} \leq 1, (i = 1, 2, \dots, n). \end{aligned} \quad (1)$$

Here, the membership weights $a \in (0, +\infty)$ and $b \in (0, +\infty)$ control the roles of the fuzzy membership degree and possibilistic membership degree. Furthermore, the exponential factors of fuzzy membership and possibilistic membership are $m \in (1, +\infty)$ and $q \in (1, +\infty)$, respectively. γ_j is a penalty factor that is usually determined by [20]:

$$\gamma_j = K \frac{\sum_{i=1}^n u_{ij}^m \|x_i - v_j\|_2^2}{\sum_{i=1}^n u_{ij}^m},$$

where $K > 0$, with the most common choice being $K = 1$.

PFCM usually updates the fuzzy membership u_{ij} , the possibilistic membership t_{ij} , and the center v_j alternatively by

$$u_{ij}^{(k+1)} = \frac{\|x_i - v_j^{(k)}\|^{\frac{2}{1-m}}}{\sum_{l=1}^c \|x_i - v_l^{(k)}\|^{\frac{2}{1-m}}}, \quad (2)$$

$$t_{ij}^{(k+1)} = \left[1 + \left(\frac{b}{\gamma_j} \|x_i - v_j^{(k)}\|^2 \right)^{\frac{1}{q-1}} \right]^{-1}, \quad (3)$$

$$v_j^{(k+1)} = \frac{\sum_{i=1}^n \left[a \left(u_{ij}^{(k)} \right)^m + b \left(t_{ij}^{(k)} \right)^q \right] x_i}{\sum_{i=1}^n \left[a \left(u_{ij}^{(k)} \right)^m + b \left(t_{ij}^{(k)} \right)^q \right]}. \quad (4)$$

When the membership weight a in the objective function of PFCM is 0, we realize PCM within the PFCM framework, and then Equations (3) and (4) are the iterative functions, respectively. When the membership weight b in the objective function of PFCM is 0, we realize FCM in the PFCM framework, and then Equations (2) and (4) are the corresponding iterative functions.

2.3. Majorization-Minimization Method

The majorization-minimization (MM) method [31] is an iterative optimization algorithm. It works by constructing a surrogate function that is simpler to minimize.

Consider the following optimization problem:

$$\min_{\theta} g(\theta), \text{ s.t. } \theta \in \Theta,$$

where Θ is a nonempty closed set in R^n and $g : \Theta \rightarrow R$ is a continuous function. The MM method iteratively reduces a series of surrogate functions $f(\theta|\theta_t)$ majorizing the objective function $g(\theta)$ at the current iterate θ_t . The principle of majorization involves two key aspects: first, equality is achieved between the surrogate function and the objective function at the current iterate θ_t , denoted as $f(\theta_t|\theta_t) = g(\theta_t)$; and second, the surrogate function dominates the objective function, meaning $f(\theta|\theta_t) \geq g(\theta)$ for all θ .

The update rule of the MM method is defined as follows [9]:

$$\theta_{t+1} \leftarrow \arg \min_{\theta} f(\theta|\theta_t),$$

which implies the descent property

$$g(\theta_{t+1}) \leq f(\theta_{t+1}|\theta_t) \leq f(\theta_t|\theta_t) = g(\theta_t).$$

By successively minimizing these surrogate functions, we aim to converge to a local minimum of the original optimization problem. This process continues until convergence is achieved or another termination criterion is met. It finds wide applications in various fields, including machine learning [9], signal processing [32], and statistics [33].

3. Alternative Derivation Method for Possibilistic Fuzzy C-Means Clustering

In this section, we propose an alternative derivation method for PFCM, which interprets its iterative process using the MM method. Firstly, a simplified optimization model, which only contains the variables U and T , is obtained by eliminating V . Secondly, the MM method is applied to solve the new problem. Finally, we provide proof to demonstrate that this new derivation method is the same as the original PFCM method.

3.1. Formulation

Let $F = U^m$, $G = T^q$, and $Z = aF + bG$, and then according to Equation (4), we have

$$v_j = \frac{\sum_{i=1}^n (af_{ij} + bg_{ij})x_i}{\sum_{i=1}^n (af_{ij} + bg_{ij})} = \frac{X(af_j + bg_j)}{(af_j + bg_j)^T e} = \frac{Xz_j}{z_j^T e}.$$

Substituting v_j into Equation (1), we have

$$\begin{aligned} \min_{(U,T)} J_{PFCM}(U, T) &= \sum_{i=1}^n \sum_{j=1}^c z_{ij} \left(x_i^T x_i - 2x_i^T v_j + v_j^T v_j \right) + \sum_{j=1}^c \gamma_j \sum_{i=1}^n (1 - t_{ij})^q \\ &= \sum_{i=1}^n \sum_{j=1}^c z_{ij} x_i^T x_i - 2 \sum_{j=1}^c z_j^T X^T v_j + \sum_{j=1}^c z_j^T e v_j^T v_j + \sum_{j=1}^c \gamma_j \sum_{i=1}^n (1 - t_{ij})^q \quad (5) \\ &= \sum_{i=1}^n \sum_{j=1}^c z_{ij} x_i^T x_i - \sum_{j=1}^c \frac{z_j^T X^T X z_j}{z_j^T e} + \sum_{j=1}^c \gamma_j \sum_{i=1}^n (1 - t_{ij})^q. \end{aligned}$$

Because the second term in problem (5) involves a proportional term, it makes problem (5) difficult to solve directly. Therefore, we introduce the MM method to optimize this equivalent optimization problem, and the next primary issue is to find a surrogate function that is easier to optimize. Considering that

$$\phi(Z) = \frac{Z^T X^T X Z}{Z^T e} \quad (6)$$

is a convex function (as proven in Appendix A), then its opposite function is concave. Based on the property of concave functions, which states that the tangent line of a concave function at any point lies above the graph at that point, the following inequality holds:

$$-\phi(z_j) \leq -\phi(z_j^{(k)}) - \left(\omega_j^{(k)} \right)^T (z_j - z_j^{(k)}),$$

where

$$\omega_j^{(k)} = \frac{2X^T X z_j^{(k)}}{\left(z_j^{(k)} \right)^T e} - \frac{\left(z_j^{(k)} \right)^T X^T X z_j^{(k)} e}{\left[\left(z_j^{(k)} \right)^T e \right]^2} \quad (7)$$

is the first-order derivative function of $\phi(z_j)$ with respect to z_j at the current iterate $(u_j^{(k)}, t_j^{(k)})$. Therefore, $-\left(\omega_j^{(k)} \right)^T z_j$ is further optimized as the surrogate function for $-\phi(z_j)$. Substituting this surrogate function into problem (5), we have

$$\begin{aligned} \min_{(U,T)} J_{PFCM}(U, T) &= \sum_{i=1}^n \sum_{j=1}^c z_{ij} x_i^T x_i - \sum_{j=1}^c \left(\omega_j^{(k)} \right)^T z_j + \sum_{j=1}^c \gamma_j \sum_{i=1}^n (1 - t_{ij})^q \\ &= \sum_{i=1}^n \sum_{j=1}^c z_{ij} x_i^T x_i - \sum_{i=1}^n \sum_{j=1}^c z_{ij} \omega_{ij}^{(k)} + \sum_{j=1}^c \gamma_j \sum_{i=1}^n (1 - t_{ij})^q \quad (8) \\ &= \sum_{i=1}^n \sum_{j=1}^c \left(au_{ij}^m + bt_{ij}^q \right) \left(x_i^T x_i - \omega_{ij}^{(k)} \right) + \sum_{j=1}^c \gamma_j \sum_{i=1}^n (1 - t_{ij})^q. \end{aligned}$$

Further, the update formulas for the variables U and T are derived using the alternating iteration method.

3.2. Optimization Procedure

We optimize problem (8) with respect to one variable, with the other variables being fixed, which leads to the following two sub-problems.

Firstly, fixing the variable T , the U -update step involves minimizing problem (8) with respect to the variable U , which can be expressed as follows:

$$\min_U \sum_{i=1}^n \sum_{j=1}^c a u_{ij}^m (x_i^T x_i - \omega_{ij}^{(k)}), \quad (9)$$

$$s, t, \sum_{j=1}^c u_{ij} = 1, 0 \leq u_{ij} \leq 1, (i = 1, \dots, n).$$

We use the Lagrange multiplier method to derive the iterative function of u_{ij} as follows:

$$u_{ij}^{(k+1)} = \frac{\left(x_i^T x_i - \omega_{ij}^{(k)}\right)^{\frac{1}{1-m}}}{\sum_{l=1}^c \left(x_i^T x_i - \omega_{il}^{(k)}\right)^{\frac{1}{1-m}}}. \quad (10)$$

Secondly, fixing the variable U , the T -update step involves minimizing problem (8) with respect to the variable T , which can be expressed as follows:

$$\min_T \sum_{i=1}^n \sum_{j=1}^c b t_{ij}^q (x_i^T x_i - \omega_{ij}^{(k)}) + \sum_{j=1}^c \gamma_j \sum_{i=1}^n (1 - t_{ij})^q. \quad (11)$$

Taking the derivative of Equation (11) with respect to t_{ij} and setting it to zero, we have

$$t_{ij}^{(k+1)} = \frac{1}{1 + \left[\frac{b(x_i^T x_i - \omega_{ij}^{(k)})}{\gamma_j} \right]^{\frac{1}{q-1}}}. \quad (12)$$

Obviously, according to Equation (7), we know that w_j is dependent on U and T . When we obtain U and T by calculating Equations (10) and (12), U and T are repeatedly utilized to update w_j using Equation (7). This means that we can calculate U , T , and w_j iteratively. The detailed process for solving problem (5) is summarized in Algorithm 1.

Algorithm 1 Alternative derivation method for PFCM

- 1: **Input** X and c .
 - 2: **Initialize** $U, T, \gamma_j (j = 1, 2, \dots, c)$, and set $k = 0$.
 - 3: **repeat**
 - 4: Calculate $\omega_j (j = 1, 2, \dots, c)$ by Equation (7);
 - 5: Calculate U by Equation (10);
 - 6: Calculate T by Equation (12);
 - 7: $k = k + 1$;
 - 8: **until** convergence
 - 9: **Output** U and T .
-

Next, we provide proof to demonstrate that this new derivation method is the same as the original PFCM method, as shown in Theorem 1.

Theorem 1. Algorithm 1 is equivalent to the original PFCM algorithm.

Proof. From Equation (7), we can further deduce the value of each item in matrix W as follows:

$$\omega_{ij}^{(k)} = 2x_i^T v_j^{(k)} - \left(v_j^{(k)}\right)^T v_j^{(k)}, \quad (13)$$

and then we have

$$x_i^T x_i - \omega_{ij}^{(k)} = \|x_i - v_j^{(k)}\|^2. \quad (14)$$

Substituting Equation (14) into Equation (10), we observe that Equation (10) is the same as Equation (2). Similarly, substituting Equation (14) into Equation (12), we observe

that Equation (12) is the same as Equation (3). Therefore, based on the above analysis, Algorithm 1 is equivalent to the original PFCM algorithm, as their final iterative functions of U and T are the same. $\square$

To sum up, for the original PFCM problem, we provide a solution using the MM method by proving the convexity of the proportional term in problem (5). This represents a new derivation approach not found in other studies.

4. Majorization-Minimization Method for Possibilistic Fuzzy C-Means Clustering

In this section, another method is proposed to optimize the problem (5), called MMPFCM. Compared to PFCM, MMPFCM obtains better local minima.

4.1. Formulation

Here, we first present a lemma that introduces a simple optimization problem. In this lemma, a new variable s is introduced to eliminate the ratio term in the optimization problem, transforming the original optimization problem into an easily solvable equivalent form. The technique of introducing new variables in the lemma provides a basis for the equivalent transformation of the optimization model in this section.

Lemma 1. For a constant $\mu > 0$, the equation $-\frac{\eta^2}{\mu} = \min_s s^2\mu - 2s\eta$ holds, and the optimization objective reaches its minimum when $s = \frac{\eta}{\mu}$.

According to Lemma 1, let $\mu_j = z_j^T e$ and $\eta_j = \sqrt{z_j^T X^T X z_j}$, and we have

$$s_j = \frac{\sqrt{z_j^T X^T X z_j}}{z_j^T e}. \quad (15)$$

Therefore, in order to eliminate the proportional term in problem (5), we introduce a new variable $s = (s_1, s_2, \dots, s_c)^T$, and then problem (5) can be transformed into

$$\min_{(U, T, s)} J_{PFCM}(U, T, s) = \sum_{i=1}^n \sum_{j=1}^c z_{ij} x_i^T x_i + \sum_{j=1}^c \left[s_j^2 z_j^T e - 2s_j \sqrt{z_j^T X^T X z_j} \right] + \sum_{j=1}^c \gamma_j \sum_{i=1}^n (1 - t_{ij})^q. \quad (16)$$

Before solving problem (16), the following theorem provides the equivalence proof between problem (5) and problem (16).

Theorem 2. Problem (16) is equivalent to problem (5).

Proof. In order to prove that those two problems are equivalent, we substitute s_j in Equation (15) into problem (16), and it can be immediately concluded that problem (16) is equivalent to problem (5). $\square$

Problem (16) also involves three variables U , T , and s . However, compared to the variable V of $p \times c$ size in PFCM, the variable s in MMPFCM is a vector with c elements, resulting in smaller space complexity. Next, the alternating iteration method is used to solve problem (16).

4.2. Optimization Procedure

We optimize problem (16) with respect to one variable, with the other variables being fixed, which leads to the following three sub-problems.

Firstly, when the variables U and T are fixed, the s -update step is the minimization of problem (16) with respect to the variable s . Because the c components of s are separable, we have the following optimization problem:

$$\min_{s_j} s_j^2 z_j^T e - 2s_j \sqrt{z_j^T X^T X z_j}. \quad (17)$$

Taking the derivative of Equation (17) with respect to s_j and setting the derivative value to zero, we obtain s_j , as shown in Equation (15).

Secondly, when the variable s is fixed, the optimization problem involving the variables U and T can be denoted as follows:

$$\min_{(U,T)} J_{PFCM}(U, T) = \sum_{i=1}^n \sum_{j=1}^c z_{ij} x_i^T x_i + \sum_{j=1}^c \left[s_j^2 z_j^T e - 2s_j \sqrt{z_j^T X^T X z_j} \right] + \sum_{j=1}^c \gamma_j \sum_{i=1}^n (1 - t_{ij})^q. \quad (18)$$

Since problem (18) involves an optimization objective with a square root, the derivative for direct optimization is quite complex. Therefore, we continue to rely on the principles of the MM method to search for surrogate functions as a further optimization model. It is obvious that $X^T X$ is a positive semidefinite matrix, so $\sqrt{z_j^T X^T X z_j}$ is a convex function about z_j . Then, we can immediately conclude that $-\sqrt{z_j^T X^T X z_j}$ is a concave function about z_j . Concavity supplies the linear majorization

$$-\sqrt{z_j^T X^T X z_j} \leq -\sqrt{\left(z_j^{(k)}\right)^T X^T X z_j^{(k)}} - \left(\alpha_j^{(k)}\right)^T \left(z_j - z_j^{(k)}\right),$$

where

$$\alpha_j^{(k)} = \frac{X^T X z_j^{(k)}}{\sqrt{\left(z_j^{(k)}\right)^T X^T X z_j^{(k)}}} \quad (19)$$

is the first-order derivative function of $\sqrt{z_j^T X^T X z_j}$ with respect to z_j at the current iterate $\left(u_j^{(k)}, t_j^{(k)}\right)$. Therefore, $-\left(\alpha_j^{(k)}\right)^T z_j$ is chosen as the surrogate function for $-\sqrt{z_j^T X^T X z_j}$. Substituting this surrogate function into problem (18), we have

$$\begin{aligned} \min_{(U,T)} J_{PFCM}(U, T) &= \sum_{i=1}^n \sum_{j=1}^c (af_{ij} + bg_{ij}) x_i^T x_i + \sum_{j=1}^c \left[s_j^2 (af_j + bg_j)^T e - 2s_j \left(\alpha_j^{(k)}\right)^T (af_j + bg_j) \right] \\ &\quad + \sum_{j=1}^c \gamma_j \sum_{i=1}^n (1 - t_{ij})^q. \end{aligned} \quad (20)$$

Further, fixing the variables T , the U -update step involves minimizing problem (20) with respect to the variable U , which can be expressed as follows:

$$\begin{aligned} &\min_U \sum_{i=1}^n \sum_{j=1}^c (af_{ij}) x_i^T x_i + \sum_{j=1}^c \left[s_j^2 (af_j)^T e - 2s_j \left(\alpha_j^{(k)}\right)^T (af_j) \right] \\ &\Leftrightarrow \\ &\min_U \sum_{i=1}^n \sum_{j=1}^c (af_{ij}) x_i^T x_i + \sum_{i=1}^n \sum_{j=1}^c \left[s_j^2 (af_{ij}) - 2s_j (af_{ij}) \alpha_{ij}^{(k)} \right] \\ &\Leftrightarrow \\ &\min_U \sum_{i=1}^n \sum_{j=1}^c u_{ij}^m \left(ax_i^T x_i + as_j^2 - 2as_j \alpha_{ij}^{(k)} \right). \end{aligned} \quad (21)$$

u_{ij} is solved using the Lagrange multiplier method, and we have

$$u_{ij}^{(k+1)} = d_{ij}^{\frac{1}{1-m}} \bigg/ \sum_{l=1}^c d_{il}^{\frac{1}{1-m}} \quad (22)$$

with $d_{ij} = x_i^T x_i + s_j^2 - 2s_j \alpha_{ij}^{(k)}$.

Then, fixing the variables U , the T -update step involves minimizing problem (20) with respect to the variable T , which can be expressed as follows

$$\begin{aligned}
 & \min_T \sum_{i=1}^n \sum_{j=1}^c (bg_{ij}) x_i^T x_i + \sum_{j=1}^c \left[s_j^2 (bg_j)^T e - 2s_j (\alpha_j^{(k)})^T (bg_j) \right] + \sum_{j=1}^c \gamma_j \sum_{i=1}^n (1 - t_{ij})^q \\
 & \Leftrightarrow \\
 & \min_T \sum_{i=1}^n \sum_{j=1}^c (bg_{ij}) x_i^T x_i + \sum_{i=1}^n \sum_{j=1}^c \left[s_j^2 (bg_{ij}) - 2s_j (bg_{ij}) \alpha_{ij}^{(k)} \right] + \sum_{j=1}^c \gamma_j \sum_{i=1}^n (1 - t_{ij})^q \quad (23) \\
 & \Leftrightarrow \\
 & \min_T \sum_{i=1}^n \sum_{j=1}^c t_{ij}^q \left(bx_i^T x_i + bs_j^2 - 2bs_j \alpha_{ij}^{(k)} \right) + \sum_{j=1}^c \gamma_j \sum_{i=1}^n (1 - t_{ij})^q.
 \end{aligned}$$

Taking the derivative of Equation (23) with respect to t_{ij} and setting it to zero, we have

$$t_{ij}^{(k+1)} = \left[1 + \left(\frac{bd_{ij}}{\gamma_j} \right)^{1/(q-1)} \right]^{-1} \quad (24)$$

with $d_{ij} = x_i^T x_i + s_j^2 - 2s_j \alpha_{ij}^{(k)}$.

From Equation (19), we find that α_j is changed by the change in U and T . So, when we calculate U using Equation (22) and T using Equation (24), α_j is updated accordingly, which means that we can calculate α_j , U , and T iteratively. This process continues until convergence is achieved or another termination criterion is met.

What needs to be pointed out is that this loop is nested within the loop of the proposed method. Therefore, we refer to this loop as the inner loop of the algorithm, with its termination condition set as the maximum number of iterations of the loop being less than or equal to K . It should be noted that this paper selects $K = 5$ as the reference value for the comparative experiments, as detailed in Section 6.

In summary, a new optimization method, called MMPFCM, is summarized in Algorithm 2.

Algorithm 2 Majorization-minimization method for possibilistic fuzzy c-means clustering (MMPFCM)

- 1: **Input** X and c .
 - 2: **Initialize** U , T , γ_j ($j = 1, 2, \dots, c$), the number of times for the inner loop K , and set $t = 0$.
 - 3: **repeat**
 - 4: Calculate s_j ($j = 1, 2, \dots, c$) by Equation (15);
 - 5: **for** $k = 1 : K$
 - 6: Calculate α_j ($j = 1, 2, \dots, c$) by Equation (19);
 - 7: Calculate U by Equation (22);
 - 8: Calculate T by Equation (24);
 - 9: **end for**
 - 10: $t = t + 1$;
 - 11: **until** convergence
 - 12: **Output** U and T .
-

As is well known, PFCM is an extension of FCM and PCM. Therefore, MMPFCM can also be interpreted as a generalization of MMFCM and MMPCM, where MMFCM and MMPCM are two cases when $b = 0$ or $a = 0$ in MMPFCM, respectively. Relevant experiments regarding these two methods are provided in Appendix B.

4.3. An Interesting Observation

Algorithm 2 consists of two nested loops: an inner loop and an outer loop. Through analysis, we have made an interesting observation, that is, when the inner loop is executed only once, Algorithm 2 degenerates to the original PFCM method. For a detailed analysis, refer to Theorem 3.

Theorem 3. When $K = 1$ in Algorithm 2, Algorithm 2 is equivalent to the original PFCM method in this case.

Proof. If the inner loop is executed only once, then Algorithm 2 contains one loop, and its detailed steps are as follows: calculate s_j using Equation (15), calculate α_j using Equation (19), calculate u_{ij} using Equation (22), and calculate t_{ij} using Equation (24).

From Equation (19), we can further deduce the value of each item in matrix A as follows:

$$\alpha_{ij}^{(k)} = \frac{x_i^T X z_j^{(k)}}{\sqrt{\left(z_j^{(k)}\right)^T X^T X z_j^{(k)}}}. \quad (25)$$

Then, substituting Equations (15) and (25) into Equation (22), we have

$$u_{ij}^{(k+1)} = \frac{\left\|x_i - v_j^{(k)}\right\|^{\frac{2}{1-m}}}{\sum_{l=1}^c \left\|x_i - v_l^{(k)}\right\|^{\frac{2}{1-m}}},$$

which is the same as Equation (2). Similarly, substituting Equations (15) and (25) into Equation (24), we observe that the obtained $t_{ij}^{(k+1)}$ is similar to Equation (3). Therefore, based on the above analysis, when the inner loop is executed only once in Algorithm 2, Algorithm 2 degenerates to the original PFCM method. $\square$

In addition, Theorem 1 proves that Algorithm 1 is equivalent to the original PFCM method, so when $K = 1$ in Algorithm 2, Algorithm 2 is also equivalent to Algorithm 1.

5. Theoretical Analysis

5.1. Convergence Analysis

If we want to prove the convergence of Algorithm 2, it is essential to initially prove the convergence of the inner loop in Algorithm 2. Therefore, let us first direct our attention to proving the convergence of its inner loop.

Theorem 4. The inner loop in Algorithm 2 will decrease the objective value of problem (18) in each iteration until it converges.

Proof. In Equation (20), let

$$f(Z) = \sum_{i=1}^n \sum_{j=1}^c z_{ij} x_i^T x_i + \sum_{j=1}^c s_j^2 z_j^T e + \sum_{j=1}^c \gamma_j \sum_{i=1}^n (1 - t_{ij})^q.$$

Then, Equation (20) becomes $f(Z) - \sum_{j=1}^c 2s_j \left(\alpha_j^{(k)}\right)^T z_j$. Let $\bar{U}$ and $\bar{T}$ be the updated U and T in each iteration, and then we have

$$f(\bar{Z}) - \sum_{j=1}^c 2s_j \left(\alpha_j^{(k)}\right)^T \bar{z}_j \leq f(Z) - \sum_{j=1}^c 2s_j \left(\alpha_j^{(k)}\right)^T z_j. \quad (26)$$

Furthermore, according to $\sqrt{z_j^T X^T X z_j}$ is convex about z_j , and convexity supplies the linear majorization

$$\sqrt{\bar{z}_j^T X^T X \bar{z}_j} - \sqrt{z_j^T X^T X z_j} \geq \left(\alpha_j^{(k)}\right)^T \bar{z}_j - \left(\alpha_j^{(k)}\right)^T z_j,$$

Further, multiplying both sides by $\sum_j 2s_j$, we have

$$\sum_j 2s_j \left(\alpha_j^{(k)}\right)^T \bar{z}_j - \sum_j 2s_j \sqrt{\bar{z}_j^T X^T X \bar{z}_j} \leq \sum_j 2s_j \left(\alpha_j^{(k)}\right)^T z_j - \sum_j 2s_j \sqrt{z_j^T X^T X z_j}. \quad (27)$$

Adding both sides of Inequality (26) and (27) yields

$$f(\bar{Z}) - \sum_j 2s_j \sqrt{\bar{z}_j^T X^T X \bar{z}_j} \leq f(Z) - \sum_j 2s_j \sqrt{z_j^T X^T X z_j}.$$

Therefore, the inner loop in Algorithm 2 decreases the objective value of problem (18) in each iteration until it converges. $\square$

In conclusion, since the inner loop in Algorithm 2 converges, Algorithm 2 also converges.

5.2. Complexity Analysis

For the computational complexity of MMPFCM, since the multiplication operation of matrices generally requires more time than simple addition operations, we focus solely on the multiplication operation.

In Algorithm 2, for Step 4, computing s_j ($j = 1, \dots, c$) needs $O(np + pc + nc)$. For Step 6, we need $O(2np + pc)$ to calculate α_j ($j = 1, \dots, c$). For Step 7, computing U needs $O(np + nc)$. For Step 8, we need $O(np + nc)$ to calculate T . Therefore, the total computational complexity of MMPFCM is $O(((2np + 2nc + 2np + pc)t_1 + np + pc + nc)t_2)$, where t_1 and t_2 are defined as the number of iterations of the inner loop and the outer loop in Algorithm 2, respectively.

As a consequence, MMPFCM has the same linear complexity with respect to the number of samples, i.e., $O(npct_1t_2)$. Next, we verify the effectiveness of the proposed method through experiments.

6. Experiments

To verify the effectiveness and clustering performance of the proposed algorithm, experimental studies are conducted on twelve real-world datasets, all selected from the UCI Machine Learning Repository (<http://archive.ics.uci.edu/ml/index.php>, accessed on 26 February 2024). The specifics of these datasets are outlined in Table 1.

Table 1. Benchmark datasets.

Datasets	Instance	Feature	Class
SCADI	70	205	7
balance	625	4	3
glass	214	9	7
COIL20	1440	1024	20
ORL	400	1024	40
Yale32	165	1024	15
Yale64	165	4096	15
Isolet5	1559	617	26
Urban	168	147	9
Iris	150	4	3
Vehicle	846	18	4
USPS	9298	256	10

All the experiments are run on a personal computer with an Intel Core i5-6500 processor and a maximum memory of 16 GB for all processes. The computer operates on Windows 10 with MATLAB R2017b. The convergence criterion is set as $|obj(it) - obj(it - 1)| < 10^{-5}$, where $obj(it)$ represents the objective function value at the end of the it -th iteration. The results are provided in tables and figures to verify the superiority of the proposed method.

6.1. Evaluation Metrics

To evaluate the performance of the proposed algorithm, four external metrics, including the overall F-measure for the entire dataset (F^*), Normalized Mutual Information (NMI), Adjusted Rand Index (ARI), and purity, are used to measure the agreement between the ground truth and the clustering results produced by the algorithm [14,34,35]. For all four metrics, higher scores correspond to improved clustering quality.

Metrics that do not require the labels of data are used for the performance evaluation and are called internal metrics. Two internal validity metrics, including the DBI [36] and XB [37], are selected here. It is worth noting that for both metrics, smaller values indicate better clustering performance [5].

The objective function value, time, and number of iterations are the remaining three evaluation metrics used to indicate the efficiency of the algorithms.

6.2. Setting of the Iterations in the Inner Loop

Since the setting of the iterations for the inner loop in Algorithm 2 is the primary issue that needs to be addressed for the execution of this method, in this subsection, we investigate the impact of different K on the clustering performance of the algorithm in order to select the appropriate number of iterations for the subsequent comparative experiments.

To study the effect of K on MMPFCM, we set the values of K as 1, 2, 5, 8, 10, and 30, respectively. MMPFCM is executed 10 times under random initializations, with the clarification that the initializations are the same for different values of K . We record the mean and standard deviation of F^* , as well as the purity, across 10 experiments for different values of K . The results are shown in Figure 1.

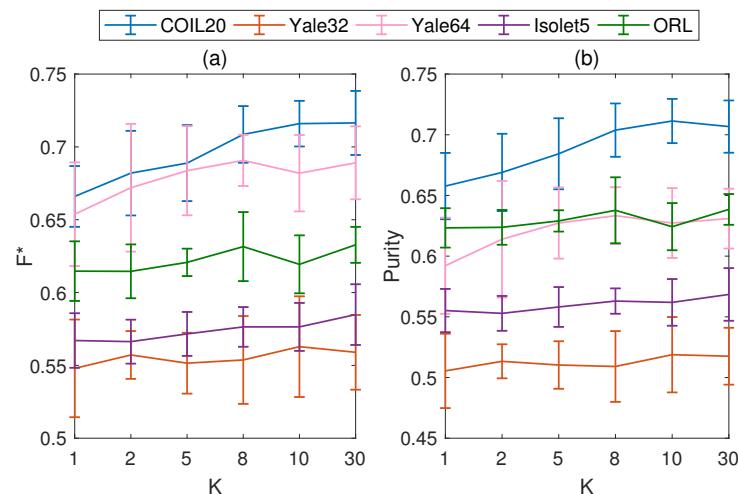


Figure 1. Mean and standard deviation of F^* , as well as Purity, for different values of K on five real-world datasets, where K is the number of iterations in the inner loop. (a) Mean and standard deviation of F^* . (b) Mean and standard deviation of Purity.

Although the curves in Figure 1 exhibit minor fluctuations under different values of K , they generally show an upward trend. Additionally, It can be observed that the clustering performance of the algorithm shows a significant improvement when the number of iterations in the inner loop is set within the range of 5 to 10. Therefore, we can choose any number within this range to execute MMPFCM. This new update method increases the flexibility of the original model, allowing for similar fine-tuning to achieve better

performance. It should be noted that for convenience, this paper selects $K = 5$ as the reference value for all subsequent comparative experiments.

6.3. Comparison between PFCM and MMPFCM

Nie et al. [38] mentioned that a bad local minimum makes the objective value not small enough, which limits the algorithm's performance. Based on this observation, the first set of experiments is carried out to evaluate the performance of MMPFCM, focusing on the objective $J_{PFCM}(U, T, V)$ in Equation (1). To more intuitively compare the performance of MMPFCM and PFCM, we run both algorithms 10 times under the same random initializations and record their objective function values. Then, we calculate the difference in the objective function values, defined as $J_{MMPFCM} - J_{PFCM}$. Figure 2 presents a box plot of these differences across twelve real-world datasets. The green plus sign on each box plot indicates the mean of the differences. The red plus sign indicates the outliers. The red horizontal line represents the median. The length of the box represents the interquartile range (IQR), which is the range from the first quartile (Q1) to the third quartile (Q3) and shows the middle 50% of the data distribution. The whiskers extend from the box to the highest and lowest values, excluding outliers. Typically, the length of the whiskers extends up to 1.5 times the IQR from the edges of the box.

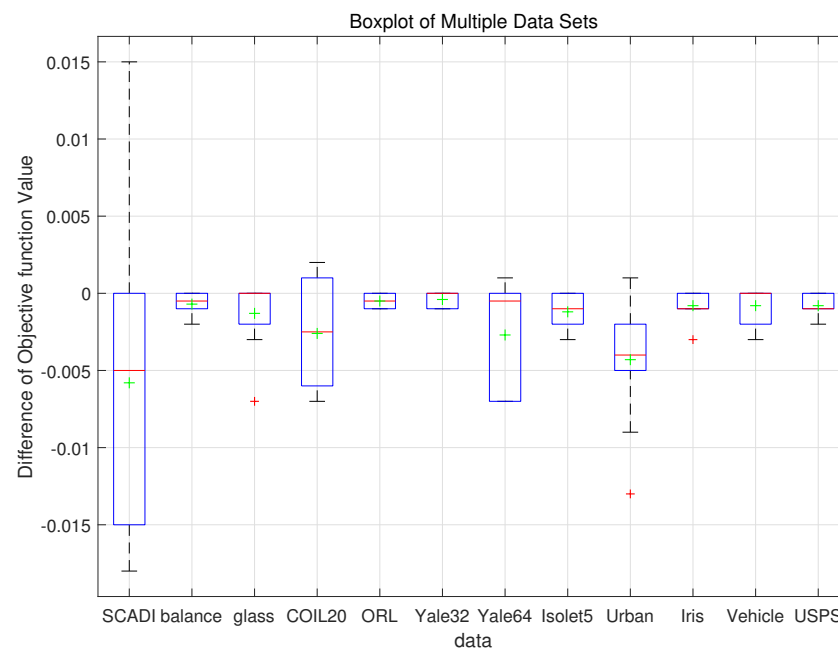


Figure 2. Box plot of the differences in objective function values across twelve real-world datasets under the same initialization conditions.

As shown in Figure 2, based on the same optimization objective function and the same random initialization conditions, the green plus signs are all below the horizontal 0 line, indicating that the mean difference in objective function values between MMPFCM and PFCM is less than 0 for these datasets. In other words, the average objective function value of MMPFCM is lower than that of PFCM. The red horizontal lines represent the median and are also below the horizontal 0 line for most datasets. This further demonstrates that the median objective function value of MMPFCM is lower than that of PFCM for most datasets, indicating better performance. Additionally, except for the COIL20 dataset, the boxes for the other datasets are below the horizontal 0 line. Since the difference is defined as $J_{MMPFCM} - J_{PFCM}$, the boxes being mostly below 0 indicates that the objective function value of MMPFCM is generally lower than that of PFCM. From this analysis, we can conclude that the proposed method achieves better local optimal solutions under the same initialization conditions.

The second set of experiments involves comparing the clustering performance of these two methods. Given that both the fuzzy membership U and the possibilistic membership T in PFCM can reflect the degree of membership of a point belonging to a particular cluster, we record the DBI and XB calculated by U and T for PFCM and MMPFCM in this set of experiments. The results are listed in Table 2 and Table 3, respectively.

Table 2. DBI and XB calculated by U for PFCM and MMPFCM on real-world datasets. The values are averaged over 10 trials with random initializations. The standard deviations are given below the means, and the best results are shown in bold.

	Datasets	SCADI	Balance	Glass	COIL20	ORL	Yale32	Yale64	Isolet5	Urban	Iris	Vehicle	USPS
DBI													
PFCM	mean	1.430	1.611	4.427	2.284	2.045	2.598	1.860	4.471	1.558	0.639	0.682	3.367
	std	(0.137)	(0.000)	(1.856)	(0.170)	(0.058)	(0.096)	(0.127)	(0.324)	(0.129)	(0.017)	(0.007)	(0.001)
MMPFCM	mean	1.275	1.610	4.305	2.258	2.037	2.561	1.806	4.448	1.507	0.634	0.682	3.366
	std	(0.097)	(0.000)	(1.707)	(0.119)	(0.075)	(0.100)	(0.137)	(0.329)	(0.078)	(0.017)	(0.007)	(0.001)
XB													
PFCM	mean	1.046	0.833	10.193	1.483	1.902	2.319	1.881	3.905	2.926	0.276	0.868	2.055
	std	(0.256)	(0.000)	(2.509)	(0.265)	(0.216)	(0.257)	(0.409)	(1.347)	(1.066)	(0.008)	(0.013)	(0.002)
MMPFCM	mean	0.808	0.833	10.092	1.350	1.811	2.317	1.706	3.871	2.071	0.274	0.867	2.054
	std	(0.168)	(0.000)	(2.202)	(0.125)	(0.205)	(0.281)	(0.429)	(1.281)	(0.110)	(0.008)	(0.013)	(0.002)

Table 3. DBI and XB calculated by T for PFCM and MMPFCM on real-world datasets. The values are averaged over 10 trials with random initializations. The standard deviations are given below the means, and the best results are shown in bold.

	Datasets	SCADI	Balance	Glass	COIL20	ORL	Yale32	Yale64	Isolet5	Urban	Iris	Vehicle	USPS
DBI													
PFCM	mean	1.433	1.617	3.415	2.286	2.048	2.596	1.858	4.469	1.609	0.639	0.682	3.366
	std	(0.138)	(0.008)	(1.127)	(0.172)	(0.057)	(0.096)	(0.135)	(0.322)	(0.130)	(0.015)	(0.011)	(0.002)
MMPFCM	mean	1.275	1.615	3.279	2.258	2.041	2.564	1.805	4.447	1.562	0.635	0.681	3.365
	std	(0.097)	(0.008)	(0.888)	(0.118)	(0.077)	(0.099)	(0.135)	(0.329)	(0.088)	(0.016)	(0.011)	(0.003)
XB													
PFCM	mean	0.155	0.059	3.764	1.053	0.497	0.457	0.417	1.321	1.668	0.178	0.797	0.905
	std	(0.033)	(0.000)	(0.327)	(0.172)	(0.052)	(0.050)	(0.091)	(0.451)	(0.371)	(0.005)	(0.010)	(0.001)
MMPFCM	mean	0.125	0.059	3.747	0.973	0.470	0.454	0.374	1.309	1.344	0.177	0.796	0.905
	std	(0.022)	(0.000)	(0.254)	(0.083)	(0.058)	(0.057)	(0.089)	(0.426)	(0.079)	(0.005)	(0.010)	(0.001)

From Tables 2 and 3, it is evident that regardless of whether the experimental results are obtained through U or T , the DBI and XB of MMPFCM are either less than or equal to those of PFCM, indicating that the proposed method exhibits better clustering performance on these twelve datasets. Moreover, the smaller standard deviation of MMPFCM also demonstrates its stability under the same initialization conditions.

The convergence curves of PFCM and MMPFCM on twelve real-world datasets are shown in Figure 3. The figures illustrate that both algorithms exhibit monotonically decreasing objective values over time. However, MMPFCM obtains better local minima on the SCADI, COIL20, ORL, Yale64, Isolet5, and Urban datasets. Furthermore, the time taken by both methods is at the same linear scale due to their identical linear complexities.

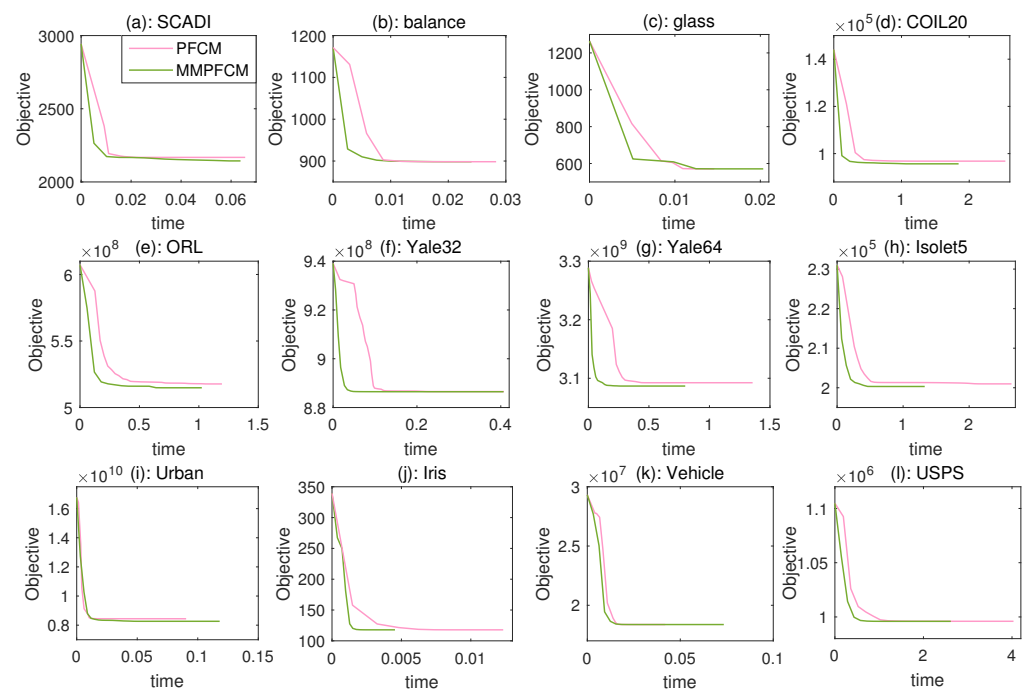


Figure 3. Convergence curves of PFCM and MMPFCM on twelve real-world datasets, where the two methods share the same initialization.

6.4. Comparison between MMPFCM and Other Methods

In order to demonstrate the performance of MMPFCM more comprehensively, MMPFCM is compared with six clustering algorithms, which include the following:

1. Fuzzy c-means (FCM) [11];
2. Iteratively re-weighted algorithm for fuzzy c-means (IRWFCM) [14];
3. An effective optimization method For fuzzy c-means with entropy regularization (IRWERFCM) [16];
4. A possibilistic fuzzy c-means clustering algorithm (PFCM) [20];
5. Generalized entropy-based possibilistic fuzzy c-means for clustering noisy data and its convergence proof (EPFCMR) [23];
6. A feature-weighted suppressed possibilistic fuzzy c-means clustering algorithm and its application to color image segmentation (FW-S-PFCM) [24].

IRWFCM and IRWERFCM are chosen because they utilize a novel method, namely the iteratively re-weighted method, to optimize FCM-type problems, and their advantages lie in their ability to achieve better local minima with fewer iterations. EPFCMR is a generalized entropy-based PFCM, which utilizes functions of distance, not the distance itself, to decrease noise contributions on the cluster centers. FW-S-PFCM introduces a feature-weighted method and a “suppressed competitive learning” strategy into the PFCM model, resulting in improved clustering performance. Additionally, it reduces the number of iterations, sensitivity to membership weights, and initializations of PFCM.

To facilitate a comprehensive comparison between MMPFCM and other algorithms, four external metrics on twelve real-world datasets are presented in Table 4. The values are averaged over 10 trials with random initializations. The standard deviations are given after the means, and the best results are shown in bold. Note that all experimental results are calculated by the obtained U for different methods. In addition, the corresponding running times of the PFCM-type algorithms are illustrated in Figure 4.

Table 4. Experimental results of different algorithms on real-world datasets. The values are averaged over 10 trials with random initializations. The standard deviations are given after the means, and the best results are shown in bold.

Datasets	Metrics	FCM	IRWFCM	IRWERFCM	PFCM	EPFCMR	FW-S-PFCM	MMPFCM
SCADI	ARI	0.267 (0.064)	0.300 (0.001)	0.327 (0.057)	0.429 (0.064)	0.434 (0.072)	0.261 (0.042)	0.436 (0.080)
	F*	0.535 (0.052)	0.560 (0.003)	0.581 (0.050)	0.630 (0.026)	0.632 (0.028)	0.528 (0.031)	0.637 (0.032)
	NMI	0.492 (0.034)	0.505 (0.000)	0.518 (0.028)	0.524 (0.019)	0.523 (0.019)	0.488 (0.029)	0.531 (0.026)
	Purity	0.734 (0.024)	0.759 (0.005)	0.753 (0.022)	0.631 (0.009)	0.636 (0.018)	0.736 (0.024)	0.637 (0.018)
balance	ARI	0.116 (0.131)	0.114 (0.095)	0.108 (0.077)	0.128 (0.119)	0.130 (0.123)	0.130 (0.011)	0.131 (0.125)
	F*	0.529 (0.119)	0.543 (0.078)	0.539 (0.056)	0.552 (0.089)	0.551 (0.093)	0.577 (0.009)	0.555 (0.088)
	NMI	0.106 (0.119)	0.098 (0.086)	0.102 (0.068)	0.112 (0.100)	0.113 (0.104)	0.112 (0.009)	0.114 (0.105)
	Purity	0.605 (0.102)	0.629 (0.067)	0.635 (0.060)	0.640 (0.075)	0.639 (0.079)	0.656 (0.016)	0.642 (0.077)
glass	ARI	0.235 (0.058)	0.243 (0.056)	0.241 (0.000)	0.221 (0.042)	0.175 (0.005)	0.234 (0.054)	0.246 (0.064)
	F*	0.549 (0.017)	0.552 (0.005)	0.536 (0.000)	0.527 (0.022)	0.457 (0.015)	0.556 (0.014)	0.558 (0.025)
	NMI	0.425 (0.032)	0.426 (0.003)	0.349 (0.000)	0.409 (0.019)	0.322 (0.005)	0.431 (0.023)	0.435 (0.036)
	Purity	0.591 (0.009)	0.592 (0.002)	0.565 (0.000)	0.587 (0.008)	0.538 (0.017)	0.593 (0.008)	0.594 (0.008)
COIL20	ARI	0.593 (0.026)	0.601 (0.018)	0.590 (0.026)	0.573 (0.012)	0.575 (0.007)	0.554 (0.023)	0.593 (0.030)
	F*	0.682 (0.025)	0.695 (0.019)	0.686 (0.018)	0.659 (0.011)	0.664 (0.006)	0.650 (0.022)	0.695 (0.029)
	NMI	0.773 (0.012)	0.782 (0.008)	0.782 (0.015)	0.772 (0.005)	0.778 (0.005)	0.766 (0.010)	0.783 (0.013)
	Purity	0.683 (0.030)	0.691 (0.020)	0.679 (0.022)	0.640 (0.013)	0.640 (0.005)	0.642 (0.018)	0.689 (0.029)
ORL	ARI	0.445 (0.021)	0.459 (0.024)	0.438 (0.028)	0.456 (0.023)	0.450 (0.017)	0.427 (0.015)	0.466 (0.013)
	F*	0.620 (0.016)	0.630 (0.022)	0.609 (0.019)	0.622 (0.022)	0.621 (0.020)	0.603 (0.013)	0.637 (0.010)
	NMI	0.784 (0.006)	0.788 (0.009)	0.776 (0.012)	0.786 (0.010)	0.785 (0.008)	0.777 (0.006)	0.784 (0.006)
	Purity	0.626 (0.020)	0.647 (0.019)	0.610 (0.019)	0.629 (0.020)	0.616 (0.017)	0.607 (0.011)	0.650 (0.012)
Yale32	ARI	0.278 (0.011)	0.276 (0.009)	0.253 (0.029)	0.295 (0.027)	0.289 (0.023)	0.250 (0.024)	0.304 (0.019)
	F*	0.528 (0.014)	0.524 (0.010)	0.500 (0.032)	0.537 (0.032)	0.535 (0.024)	0.496 (0.026)	0.559 (0.025)
	NMI	0.536 (0.006)	0.538 (0.007)	0.524 (0.020)	0.553 (0.020)	0.551 (0.019)	0.521 (0.018)	0.553 (0.013)
	Purity	0.479 (0.023)	0.468 (0.017)	0.453 (0.023)	0.495 (0.032)	0.485 (0.022)	0.453 (0.024)	0.518 (0.023)
Yale64	ARI	0.367 (0.023)	0.375 (0.021)	0.347 (0.043)	0.396 (0.020)	0.402 (0.025)	0.350 (0.016)	0.410 (0.017)
	F*	0.619 (0.028)	0.635 (0.015)	0.606 (0.040)	0.654 (0.021)	0.670 (0.023)	0.602 (0.018)	0.674 (0.019)
	NMI	0.603 (0.019)	0.615 (0.017)	0.594 (0.032)	0.623 (0.014)	0.632 (0.018)	0.599 (0.011)	0.634 (0.011)
	Purity	0.575 (0.021)	0.592 (0.008)	0.550 (0.033)	0.605 (0.021)	0.619 (0.025)	0.556 (0.017)	0.619 (0.021)
Isolet5	ARI	0.434 (0.019)	0.457 (0.023)	0.440 (0.020)	0.449 (0.019)	0.451 (0.019)	0.454 (0.014)	0.460 (0.015)
	F*	0.559 (0.015)	0.578 (0.025)	0.552 (0.016)	0.563 (0.016)	0.570 (0.020)	0.573 (0.015)	0.579 (0.012)
	NMI	0.704 (0.006)	0.711 (0.010)	0.708 (0.013)	0.706 (0.008)	0.708 (0.008)	0.710 (0.013)	0.712 (0.006)
	Purity	0.547 (0.011)	0.565 (0.025)	0.536 (0.023)	0.536 (0.016)	0.549 (0.022)	0.567 (0.019)	0.561 (0.013)
Urban	ARI	0.115 (0.014)	0.121 (0.005)	0.117 (0.002)	0.109 (0.003)	0.120 (0.012)	0.107 (0.030)	0.129 (0.000)
	F*	0.353 (0.011)	0.358 (0.002)	0.364 (0.005)	0.365 (0.006)	0.367 (0.014)	0.346 (0.033)	0.368 (0.000)
	NMI	0.278 (0.017)	0.286 (0.010)	0.271 (0.009)	0.273 (0.003)	0.280 (0.013)	0.266 (0.037)	0.303 (0.000)
	Purity	0.371 (0.013)	0.375 (0.000)	0.373 (0.000)	0.354 (0.003)	0.380 (0.013)	0.362 (0.036)	0.369 (0.000)
Iris	ARI	0.743 (0.000)	0.743 (0.000)	0.667 (0.129)	0.780 (0.007)	0.781 (0.006)	0.716 (0.000)	0.783 (0.005)
	F*	0.899 (0.000)	0.899 (0.000)	0.850 (0.090)	0.917 (0.003)	0.918 (0.003)	0.885 (0.000)	0.919 (0.002)
	NMI	0.758 (0.000)	0.758 (0.000)	0.709 (0.062)	0.763 (0.006)	0.765 (0.004)	0.742 (0.000)	0.767 (0.008)
	Purity	0.900 (0.000)	0.900 (0.000)	0.848 (0.096)	0.918 (0.003)	0.919 (0.003)	0.887 (0.000)	0.919 (0.002)
Vehicle	ARI	0.123 (0.000)	0.123 (0.000)	0.117 (0.022)	0.117 (0.003)	0.129 (0.004)	0.123 (0.000)	0.141 (0.002)
	F*	0.451 (0.000)	0.451 (0.000)	0.462 (0.033)	0.461 (0.003)	0.471 (0.003)	0.455 (0.000)	0.479 (0.002)
	NMI	0.172 (0.000)	0.172 (0.000)	0.159 (0.017)	0.152 (0.003)	0.166 (0.005)	0.187 (0.000)	0.184 (0.003)
	Purity	0.453 (0.000)	0.453 (0.000)	0.439 (0.022)	0.437 (0.004)	0.450 (0.003)	0.453 (0.000)	0.458 (0.002)
USPS	ARI	0.521 (0.014)	0.527 (0.010)	0.536 (0.000)	0.562 (0.005)	0.555 (0.001)	0.526 (0.000)	0.571 (0.000)
	F*	0.682 (0.016)	0.688 (0.012)	0.697 (0.000)	0.720 (0.004)	0.714 (0.002)	0.689 (0.000)	0.725 (0.000)
	NMI	0.605 (0.006)	0.607 (0.004)	0.613 (0.000)	0.611 (0.001)	0.612 (0.001)	0.605 (0.000)	0.610 (0.000)
	Purity	0.718 (0.017)	0.725 (0.012)	0.734 (0.000)	0.724 (0.017)	0.732 (0.020)	0.725 (0.000)	0.718 (0.000)

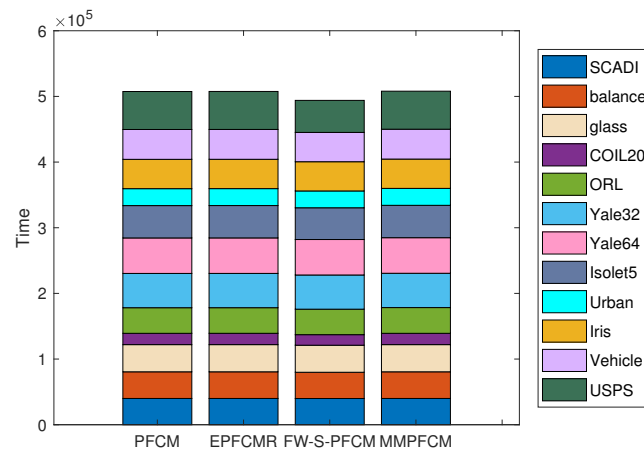


Figure 4. Plot of the corresponding times of the different algorithms on twelve real-world datasets.

From the experimental results in Table 4 and Figure 4, the following conclusions can be made:

1. Comparing the fourth and last columns of each dataset, MMPFCM consistently outperforms PFCM across all four clustering evaluation metrics on ten datasets. In addition, MMPFCM outperforms PFCM in terms of the ARI and FM on the ORL and USPS datasets, and MMPFCM outperforms PFCM in terms of purity on the ORL dataset. These results indicate the superiority of the proposed method under the same initialization conditions.
2. PFCM-type clustering algorithms have better clustering results than FCM-type clustering algorithms on the SCADI, balance, Yale32, Yale64, Iris, and USPS datasets. This is because PFCM-type clustering algorithms are better equipped to handle data with noise and outliers.
3. The total running time of FW-S-PFCM is the lowest on these twelve datasets, but this is achieved under the condition of tuning more hyperparameters. The time taken by MMPFCM and PFCM is at the same linear level, which confirms that their time complexities have the same linear relationship.

7. Conclusions

This article mainly revisits PFCM using the MM method. By eliminating the variable V , we obtain a simplified model with fewer variables, and then we provide a solution using the MM method by proving the convexity of the proportional term in this model. Through analysis, the new derivation method is shown to be equivalent to the original PFCM. In addition, we introduce a new intermediate variable s to transform the simplified model with a proportional term into an easily solvable equivalent form. Then, we design an iterative sub-problem using the MM method. For convenience, we refer to this method as MMPFCM. The complexity analysis indicates that MMPFCM and PFCM share the same computational complexity. However, MMPFCM uses the intermediate variables s of $c \times 1$ size instead of the variable V of $p \times c$ size to update U and T , resulting in smaller space complexity. Extensive experiments have shown that MMPFCM converges to a better local minimum compared to PFCM. In addition, this new updating approach enhances the flexibility of the original model, allowing for fine-tuning the number of iterations in the inner loop to achieve better performance.

In future work, we will try to apply this strategy to other PFCM-type clustering algorithms for the purpose of obtaining a better local minimum. In addition, since this method has the same computational complexity as the original PFCM, we are also committed to researching an accelerated version of MMPFCM to reduce its running time.

Author Contributions: Conceptualization, Y.C. and S.Z.; methodology, Y.C.; software, Y.C.; validation, Y.C. and S.Z.; writing—original draft preparation, Y.C.; writing—review and editing, Y.C. and S.Z.; supervision, S.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China [Grants numbers 61772020] and the Natural Science Basic Research Program of Shaanxi [Program No. 2024JC-YBMS-051].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Acknowledgments: The authors would like to thank the editors and the anonymous reviewers for their constructive comments and suggestions.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Proof of Lemma

Lemma A1. If A is a positive semidefinite matrix, and $Z^T e \geq 0$, $\phi(Z) = \frac{Z^T A Z}{Z^T e}$ is a convex function.

Proof. For any Z_1 and Z_2 both in $\text{dom } \phi$, and $\lambda \in (0, 1)$, we have

$$\begin{aligned}
 & \phi[\lambda Z_1 + (1 - \lambda)Z_2] - \lambda\phi(Z_1) - (1 - \lambda)\phi(Z_2) \\
 &= \frac{[\lambda Z_1 + (1 - \lambda)Z_2]^T A [\lambda Z_1 + (1 - \lambda)Z_2]}{[\lambda Z_1 + (1 - \lambda)Z_2]^T e} - \lambda \frac{Z_1^T A Z_1}{Z_1^T e} - (1 - \lambda) \frac{Z_2^T A Z_2}{Z_2^T e} \\
 &= \frac{\lambda^2 Z_1^T A Z_1 + (1 - \lambda)^2 Z_2^T A Z_2 + 2\lambda(1 - \lambda)Z_1^T A Z_2}{\lambda Z_1^T e + (1 - \lambda)Z_2^T e} - \lambda \frac{Z_1^T A Z_1}{Z_1^T e} - (1 - \lambda) \frac{Z_2^T A Z_2}{Z_2^T e} \\
 &= \frac{1}{\pi} \left\{ Z_1^T A Z_1 \left[\lambda(\lambda - 1) \left(Z_2^T e \right)^2 \right] + Z_2^T A Z_2 \left[\lambda(\lambda - 1) \left(Z_1^T e \right)^2 \right] \right. \\
 &\quad \left. + 2\lambda(1 - \lambda)Z_1^T A Z_2 \left(Z_1^T e \right) \left(Z_2^T e \right) \right\} \\
 &= \frac{1}{\pi} \left\{ \lambda(\lambda - 1) \left[Z_1^T A Z_1 \left(Z_2^T e \right)^2 + Z_2^T A Z_2 \left(Z_1^T e \right)^2 - 2Z_1^T A Z_2 \left(Z_1^T e \right) \left(Z_2^T e \right) \right] \right\} \\
 &= \frac{1}{\pi} \left\{ \lambda(\lambda - 1) \left[(Y_1 - Y_2)^T A (Y_1 - Y_2) \right] \right\}
 \end{aligned} \tag{A1}$$

with $\pi = [\lambda Z_1^T e + (1 - \lambda)Z_2^T e] (Z_1^T e) (Z_2^T e)$, $Y_1 = Z_2^T e Z_1$, and $Y_2 = Z_1^T e Z_2$. Since $Z^T e \geq 0$, it can be immediately concluded that

$$\phi[\lambda Z_1 + (1 - \lambda)Z_2] \leq \lambda\phi(Z_1) + (1 - \lambda)\phi(Z_2).$$

Based on the definition of convex functions, we can conclude that $\phi(Z) = \frac{Z^T A Z}{e^T Z}$ is a convex function. $\square$

Appendix B. Supplementary Experiments

When $a = 0$ in MMPFCM, we realize MMPCM in the MMPFCM framework. The algorithmic process of MMPCM can be summarized in Algorithm 2, excluding Step 7. In addition, when $b = 0$ in MMPFCM, we realize MMFCM in the MMPFCM framework. The algorithmic process of MMFCM can be summarized in Algorithm 2, excluding Step 8. In supplementary experiments, we primarily investigate the clustering effectiveness and performance of MMPCM and MMFCM.

The first part of the experiments involves comparing PCM and MMPCM. Firstly, to assess the performance of both PCM and MMPCM in finding local minima, we measure

the objective function values of both algorithms under the same random initializations. A smaller value indicates that the corresponding method achieves a better local minimum. We run each method 10 times and record the Mean_obj and Std_obj, as shown in Table A1. Note that we use the terminal outputs of FCM as the initialization of PCM and MMPCM here.

Table A1. Comparison of objective values. The values are averaged over 10 trials with random initializations. The standard deviations are given below the means, and the best results are shown in bold.

	Datasets	SCADI	Balance	Glass	COIL20	ORL	Yale32	Yale64
PCM	mean	5.7170×10^3	2.2766×10^3	3.1379×10^3	5.4845×10^5	5.5244×10^9	2.9639×10^9	9.4645×10^9
	std	(98.484)	(0.400)	(499.729)	(1.5326×10^4)	(2.9492×10^7)	(2.4593×10^7)	(1.1713×10^8)
MMPCM	mean	5.7159×10^3	2.2761×10^3	3.1379×10^3	5.4832×10^5	5.5243×10^9	2.9632×10^9	9.4636×10^9
	std	(95.202)	(0.290)	(499.729)	(1.5683×10^4)	(2.9515×10^7)	(2.5872×10^7)	(1.1731×10^8)

In Table A1, it can be observed that the objective values of MMPCM on the seven real-world datasets are all less than or equal to those of PCM, which leads to the conclusion that MMPCM achieves a better local minimum compared to PCM.

Then, we compare their clustering performance under the same random initializations. Table A2 lists the DBI on the real-world datasets for PCM and MMPCM. As shown in Table A2, the clustering results of MMPCM in terms of the DBI are either superior to or equal to those of PCM, validating that MMPCM has better clustering performance.

Table A2. DBI for PCM and MMPCM on real-world datasets. The values are averaged over 10 trials with random initializations. The standard deviations are given after the means, and the best results are shown in bold.

Datasets	SCADI	Balance	Glass	COIL20	ORL	Yale32	Yale64
PCM	1.135 (0.168)	14.651 (1.271)	14.038 (11.615)	2.978 (0.423)	17.458 (44.440)	410.359 (676.450)	41.938 (39.779)
MMPCM	1.126 (0.140)	13.184 (0.852)	12.543 (12.909)	2.945 (0.372)	16.418 (44.697)	353.354 (681.000)	41.748 (39.996)

The second part of the experiments involves comparing FCM and MMFCM. To assess the performance of both FCM and MMFCM in finding local minima, we measure the objective function values of both algorithms under the same random initializations. The experimental results are shown in Table A3.

Table A3. Comparison of objective values. The values are averaged over 10 trials with random initializations. The standard deviations are given below the means, and the best results are shown in bold.

	Datasets	SCADI	Balance	Glass	COIL20	ORL	Yale32	Yale64	Urban
FCM	mean	919.448	516.059	300.328	2.9395×10^4	1.5808×10^8	1.9207×10^8	6.1863×10^8	2.3351×10^9
	std	(26.876)	(0.001)	(4.875)	(304.055)	(6.3947×10^5)	(3.4124×10^5)	(1.5950×10^6)	(6.9282×10^7)
MMFCM	mean	910.441	516.059	295.703	2.9098×10^4	1.5856×10^8	1.9206×10^8	6.1835×10^8	2.2868×10^9
	std	(25.466)	(0.001)	(7.960)	(261.530)	(4.4893×10^5)	(2.5943×10^5)	(1.7965×10^6)	(4.1709×10^7)

As shown in Table A3, the objective values of MMFCM on the eight real-world datasets are all less than or equal to those of FCM. Therefore, we can conclude that MMFCM converges to a better local optimum compared to FCM. Moreover, the smaller standard deviation of MMFCM also demonstrates its stability under the same initialization conditions.

Table A4 lists the DBI and XB for FCM and MMFCM on the real-world datasets. As illustrated in the table, the clustering results of MMFCM in terms of the DBI and XB are all superior to those of FCM, validating that MMFCM has better clustering performance compared to FCM.

Table A4. DBI and XB for FCM and MMFCM on real-world datasets. The values are averaged over 10 trials with random initializations. The standard deviations are given after the means, and the best results are shown in bold.

Datasets	SCADI	Balance	Glass	COIL20	ORL	Yale32	Yale64	Urban
DBI								
FCM	1.168 (0.123)	5.657 (0.015)	0.629 (0.041)	1.925 (0.072)	2.206 (0.149)	3.499 (0.293)	2.367 (0.353)	0.924 (0.049)
MMFCM	1.113 (0.084)	5.656 (0.009)	0.591 (0.066)	1.896 (0.079)	2.181 (0.173)	3.470 (0.210)	2.123 (0.299)	0.904 (0.060)
XB								
FCM	0.806 (0.162)	1.513 (0.002)	0.489 (0.015)	1.154 (0.064)	2.338 (0.459)	3.125 (0.377)	3.302 (1.042)	1.082 (0.052)
MMFCM	0.694 (0.063)	1.512 (0.002)	0.475 (0.024)	1.130 (0.068)	2.231 (0.565)	3.111 (0.406)	2.611 (0.958)	1.075 (0.163)

References

- Jordan, M.I.; Mitchell, T.M. Machine learning: Trends, perspectives, and prospects. *Science* **2015**, *349*, 255–260. [\[CrossRef\]](#) [\[PubMed\]](#)
- Döring, C.; Lesot, M.J.; Kruse, R. Data analysis with fuzzy clustering methods. *Comput. Stat. Data Anal.* **2006**, *51*, 192–214. [\[CrossRef\]](#)
- Rezaee, M.R.; Van der Zwet, P.M.; Lelieveldt, B.; Van der Geest, R.J.; Reiber, J.H. A multiresolution image segmentation technique based on pyramidal segmentation and fuzzy clustering. *IEEE Trans. Image Process.* **2000**, *9*, 1238–1248. [\[CrossRef\]](#) [\[PubMed\]](#)
- Horn, D.; Gottlieb, A. Algorithm for data clustering in pattern recognition problems based on quantum mechanics. *Phys. Rev. Lett.* **2001**, *88*, 018702. [\[CrossRef\]](#) [\[PubMed\]](#)
- Li, D.; Zhou, S.; Pedrycz, W. Accelerated fuzzy c-means clustering based on new affinity filtering and membership scaling. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 12337–12349. [\[CrossRef\]](#)
- Chen, Y.; Zhou, S.; Zhang, X.; Li, D.; Fu, C. Improved fuzzy c-means clustering by varying the fuzziness parameter. *Pattern Recognit. Lett.* **2022**, *157*, 60–66. [\[CrossRef\]](#)
- Saxena, A.; Prasad, M.; Gupta, A.; Bharill, N.; Patel, O.P.; Tiwari, A.; Er, M.J.; Ding, W.; Lin, C.T. A review of clustering techniques and developments. *Neurocomputing* **2017**, *267*, 664–681. [\[CrossRef\]](#)
- Lloyd, S. Least squares quantization in PCM. *IEEE Trans. Inf. Theory* **1982**, *28*, 129–137. [\[CrossRef\]](#)
- Xu, J.; Lange, K. Power k-means clustering. In Proceedings of the International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 6921–6931.
- Ruspini, E.H. Numerical methods for fuzzy clustering. *Inf. Sci.* **1970**, *2*, 319–350. [\[CrossRef\]](#)
- Bezdek, J.C.; Ehrlich, R.; Full, W. FCM: The fuzzy c-means clustering algorithm. *Comput. Geosci.* **1984**, *10*, 191–203. [\[CrossRef\]](#)
- Jain, A.K. Data clustering: 50 years beyond k-means. *Pattern Recognit. Lett.* **2010**, *31*, 651–666. [\[CrossRef\]](#)
- Bezdek, J.C. *Pattern Recognition with Fuzzy Objective Function Algorithms*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2013.
- Xue, J.; Nie, F.; Wang, R.; Li, X. Iteratively reweighted algorithm for fuzzy c-means. *IEEE Trans. Fuzzy Syst.* **2022**, *30*, 4310–4321. [\[CrossRef\]](#)
- Miyamoto, S.; Mukaidono, M. Fuzzy c-means as a regularization and maximum entropy approach. In Proceedings of the IFSA'97 Prague: Proceedings of the Seventh International Fuzzy Systems Association World Congress, Prague, Czech Republic, 25–30 June 1997; Volume 2, pp. 86–92.
- Liang, Y.; Chen, Y.; Huang, Q.; Chen, H.; Nie, F. An effective optimization method for fuzzy k-means with entropy regularization. *IEEE Trans. Knowl. Data Eng.* **2024**, *36*, 2846–2861. [\[CrossRef\]](#)
- Chen, J.; Zhang, H.; Pi, D.; Kantardzic, M.; Yin, Q.; Liu, X. A weight possibilistic fuzzy c-means clustering algorithm. *Sci. Program.* **2021**, *2021*, 9965813. [\[CrossRef\]](#)
- Krishnapuram, R.; Keller, J.M. A possibilistic approach to clustering. *IEEE Trans. Fuzzy Syst.* **1993**, *1*, 98–110. [\[CrossRef\]](#)
- Benjamin, J.B.M.; Yang, M.S. Weighted multiview possibilistic c-means clustering with L2 regularization. *IEEE Trans. Fuzzy Syst.* **2021**, *30*, 1357–1370. [\[CrossRef\]](#)
- Pal, N.R.; Pal, K.; Keller, J.M.; Bezdek, J.C. A possibilistic fuzzy c-means clustering algorithm. *IEEE Trans. Fuzzy Syst.* **2005**, *13*, 517–530. [\[CrossRef\]](#)
- Yu, H.; Jiang, L.; Fan, J.; Lan, R. Double-suppressed possibilistic fuzzy Gustafson-Kessel clustering algorithm. *Knowl.-Based Syst.* **2023**, *276*, 110736. [\[CrossRef\]](#)
- Mai, D.S.; Ngo, L.T.; Trinh, L.H.; Hagra, H. A hybrid interval type-2 semi-supervised possibilistic fuzzy c-means clustering and particle swarm optimization for satellite image analysis. *Inf. Sci.* **2021**, *548*, 398–422. [\[CrossRef\]](#)
- Askari, S.; Montazerin, N.; Zarandi, M.F.; Hakimi, E. Generalized entropy based possibilistic fuzzy c-means for clustering noisy data and its convergence proof. *Neurocomputing* **2017**, *219*, 186–202. [\[CrossRef\]](#)
- Yu, H.; Jiang, L.; Fan, J.; Xie, S.; Lan, R. A feature-weighted suppressed possibilistic fuzzy c-means clustering algorithm and its application on color image segmentation. *Expert Syst. Appl.* **2024**, *241*, 122270. [\[CrossRef\]](#)
- Wu, C.; Zhang, X. A self-learning iterative weighted possibilistic fuzzy c-means clustering via adaptive fusion. *Expert Syst. Appl.* **2022**, *209*, 118280. [\[CrossRef\]](#)

26. Kuo, R.; Hsu, C.; Nguyen, T.P.Q.; Tsai, C. Hybrid multi-objective metaheuristic and possibilistic intuitionistic fuzzy c-means algorithms for cluster analysis. *Soft Comput.* **2024**, *28*, 991–1008. [\[CrossRef\]](#)
27. Ji, Z.; Xia, Y.; Sun, Q.; Cao, G. Interval-valued possibilistic fuzzy c-means clustering algorithm. *Fuzzy Sets Syst.* **2014**, *253*, 138–156. [\[CrossRef\]](#)
28. Tang, Y.; Hu, X.; Pedrycz, W.; Song, X. Possibilistic fuzzy clustering with high-density viewpoint. *Neurocomputing* **2019**, *329*, 407–423. [\[CrossRef\]](#)
29. Antoine, V.; Guerrero, J.A.; Romero, G. Possibilistic fuzzy c-means with partial supervision. *Fuzzy Sets Syst.* **2022**, *449*, 162–186. [\[CrossRef\]](#)
30. Yu, H.; Xu, X.; Li, H.; Wu, Y.; Lei, B. Semi-supervised possibilistic c-means clustering algorithm based on feature weights for imbalanced data. *Knowl.-Based Syst.* **2024**, *286*, 111388. [\[CrossRef\]](#)
31. Ortega, J.M.; Rheinboldt, W.C. *Iterative Solution of Nonlinear Equations in Several Variables*; SIAM: Philadelphia, PA, USA, 2000.
32. Sun, Y.; Babu, P.; Palomar, D.P. Majorization-Minimization algorithms in signal processing, communications, and machine learning. *IEEE Trans. Signal Process.* **2016**, *65*, 794–816. [\[CrossRef\]](#)
33. Ren, J.; Zhang, T.; Li, J.; Stoica, P. Sinusoidal parameter estimation from signed measurements via majorization–minimization based RELAX. *IEEE Trans. Signal Process.* **2019**, *67*, 2173–2186. [\[CrossRef\]](#)
34. Hubert, L.; Arabie, P. Comparing partitions. *J. Classif.* **1985**, *2*, 193–218. [\[CrossRef\]](#)
35. Mei, J.P.; Wang, Y.; Chen, L.; Miao, C. Large scale document categorization with fuzzy clustering. *IEEE Trans. Fuzzy Syst.* **2016**, *25*, 1239–1251. [\[CrossRef\]](#)
36. Davies, D.L.; Bouldin, D.W. A cluster separation measure. *IEEE Trans. Pattern Anal. Mach. Intell.* **1979**, *2*, 224–227. [\[CrossRef\]](#)
37. Xie, X.L.; Beni, G. A validity measure for fuzzy clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* **1991**, *13*, 841–847. [\[CrossRef\]](#)
38. Nie, F.; Xue, J.; Wu, D.; Wang, R.; Li, H.; Li, X. Coordinate descent method for k-means. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 2371–2385. [\[CrossRef\]](#) [\[PubMed\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.