

Article

# How KLFCM Works—Convergence and Parameter Analysis for KLFCM Clustering Algorithm

Chaomurilige <sup>1,2</sup>

<sup>1</sup> School of Information Engineering, Minzu University of China, Beijing 100081, China; chaomurilige@muc.edu.cn

<sup>2</sup> Key Laboratory of Ethnic Language Intelligent Analysis and Security Governance, Ministry of Education, Minzu University of China, Beijing 100081, China

**Abstract:** KLFCM is a clustering algorithm proposed by introducing K-L divergence into FCM, which has been widely used in the field of fuzzy clustering. Although many studies have focused on improving its accuracy and efficiency, little attention has been paid to its convergence properties and parameter selection. Like other fuzzy clustering algorithms, the output of the KLFCM algorithm is also affected by fuzzy parameters. Furthermore, some researchers have noted that the KLFCM algorithm is equivalent to the EM algorithm for Gaussian mixture models when the fuzzifier  $\lambda$  is equal to 2. In practical applications, the KLFCM algorithm may also exhibit self-annealing properties similar to the EM algorithm. To address these issues, this paper uses Jacobian matrix analysis to investigate the KLFCM algorithm's parameter selection and convergence properties. We first derive a formula for calculating the Jacobian matrix of the KLFCM with respect to the membership function. Then, we demonstrate the self-annealing behavior of this algorithm through theoretical analysis based on the Jacobian matrix. We also provide a reference strategy for determining the appropriate values of fuzzy parameters in the KLFCM algorithm. Finally, we use Jacobian matrix analysis to investigate the relationships between the convergence rate and different parameter values of the KLFCM algorithm. Our experimental results validate our theoretical findings, demonstrating that when selecting appropriate lambda parameter values, the KLFCM clustering algorithm exhibits self-annealing properties that reduce the impact of initial clustering centers on clustering results. Moreover, using our proposed strategy for selecting the fuzzy parameter lambda of the KLFCM algorithm effectively prevents coincident clustering results from being produced by the algorithm.



**Citation:** Chaomurilige. How KLFCM Works—Convergence and Parameter Analysis for KLFCM Clustering Algorithm. *Mathematics* **2023**, *11*, 2285. <https://doi.org/10.3390/math11102285>

Academic Editor: Javier Alcaraz

Received: 13 April 2023

Revised: 7 May 2023

Accepted: 8 May 2023

Published: 14 May 2023



**Copyright:** © 2023 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/>).

**Keywords:** KLFCM clustering algorithm; Jacobian matrix; convergence properties analysis; self-annealing

**MSC:** 68T10

## 1. Introduction

Clustering and fuzzy-based clustering have become popular techniques in data mining and machine learning due to their ability to identify patterns and group similar data points without the need for training data [1,2]. Clustering algorithms are a type of unsupervised approach where data are partitioned into subgroups based on their similarities or distances from each other [3]. The partition matrix represents the degree of membership of each data point in each cluster [4]. Since Zadeh [5] introduced the fuzzy set theory, the hard clustering algorithm was extended to the FCM (Fuzzy C-means) clustering algorithm in [6]. Clustering algorithms can be applied to a wide range of applications, such as image segmentation, customer segmentation, and anomaly detection.

In the literature, there are numerous improvements to the FCM algorithm that aim to address various clustering problems or enhance clustering performance [7–11]. The KLFCM (FCM with K-L information term) algorithm is one of the more well-known

methods. Honda and Ichihashi proposed a modified objective function for Fuzzy c-Means (FCM) clustering that includes a regularizer based on K-L information [7]. In KLFCM, a regularization term based on the K-L divergence metric is added to the objective function to encourage cluster centers to be spaced further apart. The K-L divergence is used to measure the difference between two probability distributions, however, it can be used to measure the degree of separation between two cluster centers in this case. By including the regularization term, the algorithm ensures that the distance between cluster centers is maximized, which can help prevent the formation of overlapping clusters and ensure the resulting clusters are well separated. This algorithm has such unique characteristics that it has inspired the development of many clustering algorithms based on its principles, which have been proposed in the research literature. Honda applied probabilistic principal component analysis (PCA) mixture models to linear clustering and proposed a constrained model KLFCV [12]. Gharieb and Gendy [13] modified the regularization term of the original KLFCM algorithm using the Kullback–Leibler (KL) divergence, which measures the proximity between a pixel membership and the local average of this membership in the immediate neighborhood. Zhang et al. [14] combined the benefits of KLFCM and Student's t-distribution to propose a new algorithm for image segmentation. A novel image segmentation algorithm based on KLFCM is proposed in [15] to increase the ability to overcome noise and describe the segmentation uncertainty. Amira et al. [16] incorporated conditional probability distributions and the probabilistic dissimilarity functional into the conventional KLFCM algorithm and proposed a new model called CKLFCM.

While numerous clustering methods based on the KLFCM algorithm have been proposed in the literature, few provide clear explanations for how and why this algorithm works. Furthermore, there is a lack of theoretical research into its convergence properties and optimal parameter selection. Similar to the FCM clustering algorithm, the degree of fuzziness in KLFCM's membership values is regulated by the fuzzifier parameter. Larger values lead to fuzzier memberships [17], which can result in coincident clustering when fuzziness approaches infinity. In our research, "coincident clustering result" refers to a specific type of coincident clustering where all cluster centers coincide with the dataset's mass center and merge into a single center, resulting in a loss of clustering information and decreased accuracy in data partitioning. Hence, selecting the proper fuzzifier value is crucial for obtaining accurate clustering results. Nevertheless, the use of KL divergence as a penalty term in the algorithm helps prevent the overlapping of the cluster centers by spreading them throughout the data space. Consequently, theoretically, the algorithm can eliminate overlap between the dataset's mass center and all cluster centers. We have addressed the parameter selection of the clustering algorithm by using Jacobian matrix analysis in previous papers [18–21]. In these papers, we revealed the relationship between the stable fixed points of the clustering algorithm and the datasets using Jacobian matrix analysis. In [18], we provided an explanation of the self-annealing behavior observed in the EM algorithm for Gaussian mixtures, along with the initialization lower bound of the temperature parameter in the DA-EM algorithm. In addition, Ref. [21] demonstrated that coincident clustering results are not stable fixed points of the GG clustering algorithm and discussed the correlation between the clustering algorithm's convergence rate and the fuzziness index. In this paper, we further analyze the parameter selection and convergence properties of the KLFCM clustering algorithm through Jacobian matrix analysis, building on our previous work.

The primary contributions of this paper can be summarized as follows:

- Firstly, we constructed the Jacobian matrix of the KLFCM algorithm regarding the membership function. Then, we provided theoretical proof for the self-annealing property of the KLFCM algorithm.
- We discussed the reference methods for selecting fuzzy parameters in practical applications of the KLFCM algorithm. Specifically, we talked about how to choose appropriate values for the parameters' lambda to ensure poor clustering results are avoided.

- Additionally, similar to the Hessian matrix, the Jacobian matrix can be utilized to estimate the convergence rate of an algorithm. Since computing the Jacobian matrix is simpler than computing the Hessian matrix, the third contribution of this paper is to estimate the convergence rate of the KLFCM algorithm under different parameter conditions using the Jacobian matrix.
- Finally, we conducted experiments to verify the accuracy and effectiveness of the theoretical derivation.

The experimental results indicate that the fuzzy parameter lambda has a significant impact on the clustering outcome of the algorithm, and inappropriate parameter selection can result in poor clustering performance. The research also demonstrates that the coincident clustering solution is not a stable fixed point of the KLFCM algorithm. Therefore, under certain parameter conditions (i.e., where the chosen  $\lambda$  results in the spectral radius of the Jacobian matrix at the coincident clustering center being greater than 1), even if the initial clustering center selection is suboptimal, the algorithm may still produce good clustering results. Meanwhile, we used the spectral radius of the Jacobian matrix to estimate the convergence rate of the KLFCM algorithm under different parameter conditions in the experiment and further explained the relationship between the parameters and convergence rate.

In this research, we provide an introduction to the KLFCM clustering algorithm with a brief overview in Section 2. We then analyze the Jacobian matrix and discuss the theoretical behavior of the KLFCM algorithm in Section 3. To validate our theoretical findings, we present various experimental results in Section 4. Additionally, we include a discussion on the experimental outcomes in Section 5. Finally, we summarize our research in Section 6.

## 2. The KLFCM Clustering Algorithm

This section provides a concise overview of the KLFCM clustering algorithm.

Firstly, we focus on the original FCM clustering algorithm. Let  $X = \{x_1, \dots, x_k\} \in R^s$  be a dataset from an s-dimensional Euclidean space. The aim of clustering is to find structure in data and cluster  $n$  data points to  $c$  clusters. The assignment of all items to the clusters is determined by their membership values, indicating the degree to which each item belongs to each cluster. The membership matrix  $U = [u_{ik}]_{c \times n}$  represents these values, where  $u_{ik}$  denotes the membership value of the  $i$ th data sample for the  $j$ th cluster. It should be noted that all membership values must adhere to the following constraints:

$$u_{ik} \geq 0, \forall i, j \quad \text{and} \quad \sum_{i=1}^c u_{ik} = 1$$

We denote the set of fuzzy partition matrices as

$$M_{fc} = \left\{ U = [u_{ik}]_{c \times n} \mid \forall i, \forall k, u_{ik} \geq 0, n > \sum_{k=1}^n u_{ik} > 0, \sum_{i=1}^c u_{ik} = 1 \right\}.$$

The objective function of the FCM algorithm [6] is formulated as follows:

$$J_m(U, V) = \sum_{k=1}^n \sum_{i=1}^c u_{ik}^m d_{ik}^2 \tag{1}$$

where  $U \in M_{fc}$  and  $d_{ik} = \|x_k - v_i\| = ((x_k - v_i)^T(x_k - v_i))^{1/2}$  is the Euclidian distance from the  $k$ th object  $x_k$  to the  $i$ th cluster center  $v_i$ .  $m$  is the weighting exponent which

determines the degree of fuzziness and  $1 < m < +\infty$ . The necessary conditions for optimality of (1) are derived as follows:  $\forall i = 1, \dots, c$  and  $\forall k = 1, \dots, n$

$$v_i = \frac{\sum_{k=1}^n u_{ik}^m x_k}{\sum_{k=1}^n u_{ik}^m} \tag{2}$$

$$u_{ik} = \frac{\|x_k - v_i\|^{\frac{-2}{m-1}}}{\sum_{j=1}^c \|x_k - v_j\|^{\frac{-2}{m-1}}} \tag{3}$$

Miyamoto et al. [22] proposed the introduction of an entropy term and a positive parameter  $\lambda$ , resulting in the minimization of a new objective function  $J_\lambda(U, V)$  instead of  $J_m(U, V)$ . This approach is commonly known as entropy regularization.

$$J_\lambda(U, V) = \sum_{k=1}^n \sum_{i=1}^c u_{ik} d^2(x_k, v_i) + \lambda \sum_{k=1}^n \sum_{i=1}^c u_{ik} \log u_{ik} \tag{4}$$

The objective function of the FCM clustering method with regularization by K-L information (KLFCM) is obtained by substituting the entropy term in Equation (4) with K-L information. The objective function is given by the following equation:

$$J_{klfcm}(U, V) = \sum_{i=1}^c \sum_{k=1}^n u_{ik} d_{ik} + \lambda \sum_{i=1}^c \sum_{k=1}^n u_{ik} \log \frac{u_{ik}}{\alpha_i} + \sum_{i=1}^c \sum_{k=1}^n u_{ik} |F_i| \tag{5}$$

where  $\alpha_i$  represents the proportion of samples belonging to the  $i$ th cluster. In the KLFCM algorithm, the Mahalanobis distance is utilized to quantify the dissimilarity between each data point and the cluster centers during the clustering process. The Mahalanobis distance takes into account the covariance structure of the data, which makes it a more precise distance measure than Euclidean distance. By considering the distribution of the data and the correlation between variables, it can provide better estimates of similarity or dissimilarity between data points. The formula for the Mahalanobis distance is as follows:

$$d_{ik} = (\det F_i)^{-1} \exp \left[ -(x_k - v_i)^T F_i^{-1} (x_k - v_i) \right]$$

The objective function of KLFCM (5) is minimized under the condition that  $\alpha_i > 0$ ,  $\sum_{i=1}^c \alpha_i = 1$   $\sum_{i=1}^c \sum_{k=1}^n u_{ik} = 1$ , respectively. Then, the updating rules in the KLFCM clustering algorithm are as follows:

$$v_i = \frac{\sum_{k=1}^n u_{ik} x_k}{\sum_{k=1}^n u_{ik}} \tag{6}$$

$$u_{ik} = \frac{\alpha_i (d_{ik})^{\frac{1}{\lambda}}}{\sum_{i=1}^c \alpha_i (d_{ik})^{\frac{1}{\lambda}}} \tag{7}$$

$$\alpha_i = \frac{1}{n} \sum_{k=1}^n u_{ik} \tag{8}$$

$$F_i = \frac{\sum_{k=1}^n u_{ik} (x_k - v_i)(x_k - v_i)^T}{\sum_{k=1}^n u_{ik}} \tag{9}$$

where  $d_{ik} = (\det F_i)^{-1} \exp \left[ -(x_k - v_i)^T F_i^{-1} (x_k - v_i) \right]$ . The KLFCM clustering algorithm is equivalent to the Expectation-Maximization (EM) algorithm with Gaussian Mixture Models (GMMs) only when the value of  $\lambda$  is equal to 2. This relationship between the two algorithms is well established in the literature.

The KLFCM Algorithm is summarized in Algorithm 1:

---

**Algorithm 1:** KLFCM algorithm.

---

Step 1: Assuming a fixed number of clusters and parameter value, with  $2 \leq c \leq n$  and  $\lambda$  selected, an initial matrix  $u^{(0)} = (u_1^{(0)}, \dots, u_i^{(0)})$ ,  $i = 1, \dots, c \in M_{fc}$  is chosen. The algorithm starts at  $t = 1$ .

Step 2: Calculate the cluster centers  $v^{(t)}$  using  $u^{(t-1)}$  through the utilization of Equation (6). The notation  $v^{(t)}$  denotes the cluster centers obtained in the  $t$ th iteration, while  $u^{(t-1)}$  represents the membership matrix from the previous  $t-1$ th iteration of the clustering algorithm.

Step 3: Calculate the cluster covariance matrix  $F^{(t)}$  and the matrix  $\alpha^{(t)}$  by applying Equation (9) for  $F^{(t)}$  and Equation (8) for  $\alpha^{(t)}$  in the iterative process of the KLFCM algorithm.

Step 4: Using Equation (7), revise the membership matrix  $u^{(t)}$  by incorporating the current cluster centers  $v^{(t)}$  in the iterative procedure of the KLFCM algorithm.

Step 5: if  $\|u^{(t)} - u^{(t-1)}\| < \varepsilon$  then stop, otherwise return to Step 2.

---

For a better understanding of the impact of the fuzzy parameter  $\lambda$  on the output results of the KLFCM algorithm, we conducted experiments using the Iris dataset as the experimental object to observe the clustering results obtained by selecting different fuzzy parameters. In the clustering results, we used \* (green asterisks),  $\Delta$  (red triangles), and  $\star$  (blue pentagrams) to represent sample points that belong to different clusters. The sample cluster centers were represented by black circles  $\bullet$ . In the use of the KLFCM clustering algorithm, it is generally necessary to initialize the cluster centers and membership matrix of the algorithm. To accomplish this task, we utilize the K-means clustering algorithm for initialization. Specifically, we use the K-means clustering algorithm to divide the sample data into  $k$  clusters and use each cluster's centroid as the initial cluster center in the KLFCM algorithm. Additionally, based on the K-means clustering results, we calculate the distance between each sample point and the various cluster centers, which allows us to establish the initial membership matrix in the KLFCM algorithm.

The simulation shown in Figure 1 highlights the importance of choosing an appropriate value for the parameter  $\lambda$  in the KLFCM clustering algorithm. The results demonstrate that different values of  $\lambda$  can lead to significantly different clustering outcomes, and a poor choice of  $\lambda$  can result in an invalid or uninformative clustering solution. When  $\lambda$  is set to 2 or 4, reasonable clustering outcomes are obtained with low error counts of 5 and 44, respectively. This indicates that the algorithm was able to produce meaningful clusters with acceptable levels of misclassification. However, when  $\lambda$  is improperly initialized to 8 or 36, as illustrated in Figure 1e,f, the clustering algorithm fails to produce informative results. Specifically, the algorithm outputs a single cluster, which indicates that the clustering solution is invalid and uninformative.

Following that, we manually designate the initial class center of the KLFCM algorithm under identical conditions regarding the fuzzy parameter. The initial cluster centers are closely situated to each other, but there is no complete overlap. Then we will apply the KLFCM algorithm to cluster the Iris dataset and display the clustering results in Figure 2. We use  $\circ$  (magenta circles) to represent the initial cluster centers.

The initial cluster centers are already tightly close to each other (however, it is not completely overlapping). Even with poor initial clustering center selection, the algorithm can still prevent convergence towards overlapping clustering centers as long as the fuzzy parameter  $\lambda$  is appropriately chosen. That means the KLFCM clustering algorithm possesses the capability of evading these kinds of erroneous clustering outcomes, which highlights its potential self-annealing properties. When the fuzzy parameter  $\lambda$  is set to 2, the KLFCM clustering algorithm delivers satisfactory clustering results despite the less-than-ideal initial clustering center. However, when the fuzzy parameter  $\lambda$  is set to 5, the KLFCM clustering

algorithm fails to prevent producing clustering results in which all samples are assigned to a single class. The self-annealing property refers to the ability of an algorithm to adapt and improve its performance without explicit external intervention. In the context of the KLFCM clustering algorithm, it means that the algorithm has the ability to adjust its parameters during the iterative process to achieve better clustering results. The algorithm seems to “self-anneal” toward a more meaningful clustering outcome, even when the initial parameter selection may be inappropriate.

In the upcoming section, we will perform a theoretical analysis of the KLFCM clustering algorithm using Jacobian matrix analysis.

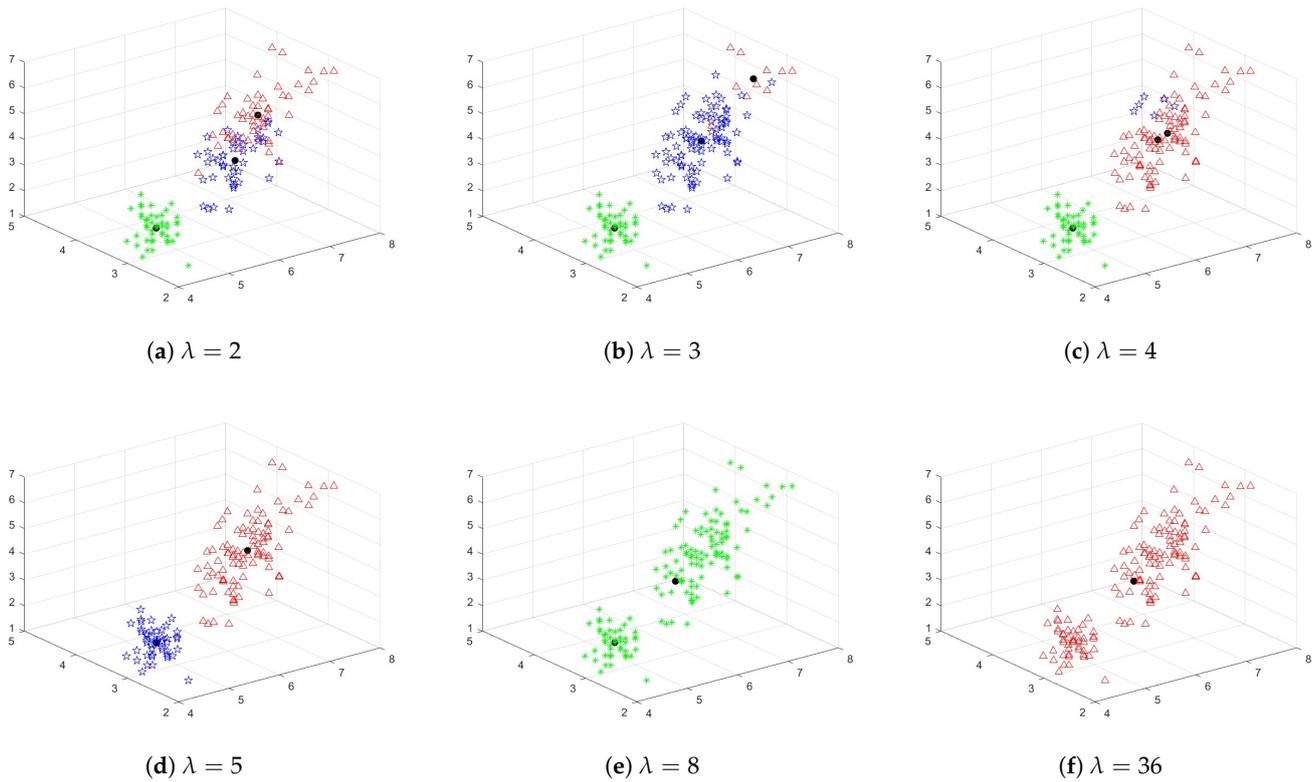


Figure 1. KLFCM Clustering Results with Different  $\lambda$  for Iris Dataset.

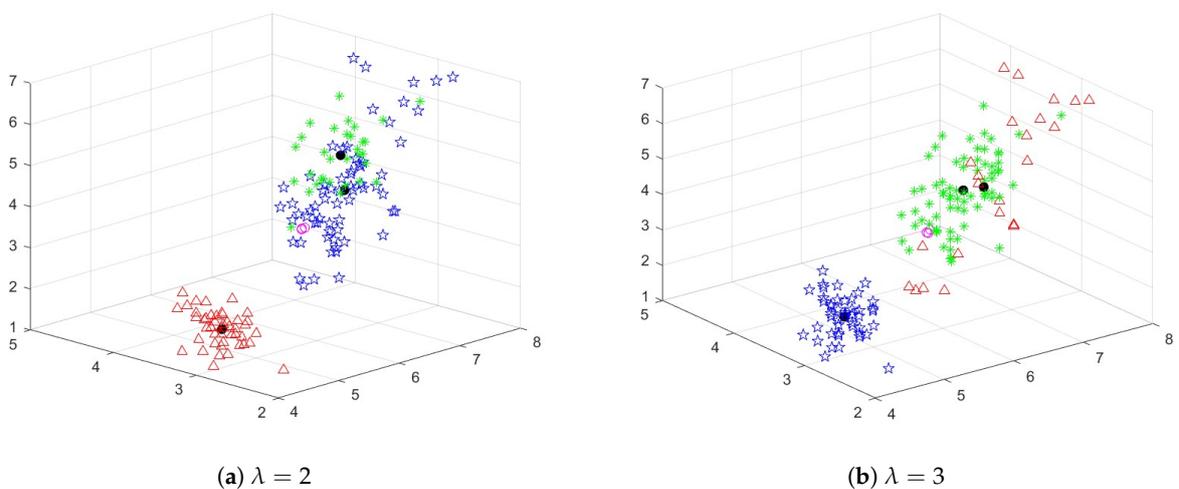


Figure 2. Cont.

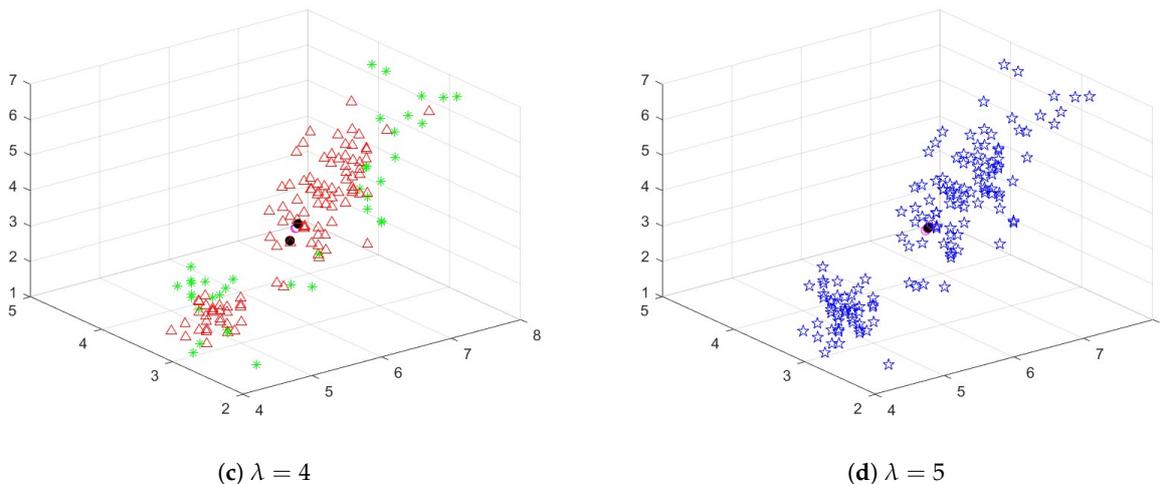


Figure 2. KLFCM Clustering Results with Inappropriate Cluster Center Initialization for Iris Dataset.

### 3. Convergence and Parameter Analysis Based on Jacobian Matrix

It is a well-known fact that when partitioning a dataset into clusters, each cluster should have distinct centers from the others. Otherwise, if all degrees of membership between samples and any clustering center are equal, it would imply that we cannot meaningfully divide the dataset into subsets based on the membership matrix. Similarly, in the case of the KLFCM algorithm, we would expect it to circumvent this potential drawback; otherwise, it cannot be considered successful as a clustering algorithm.

As we mentioned in Section 2, the KLFCM cluster centers and membership values of the data points with them are updated through the following iterations.

$$v_i^{(t)} = \frac{\sum_{k=1}^n u_{ik}^{(t-1)} x_k}{\sum_{k=1}^n u_{ik}^{(t-1)}} \tag{10}$$

$$\alpha_i^{(t)} = \frac{1}{n} \sum_{k=1}^n u_{ik}^{(t-1)} \tag{11}$$

$$F_i^{(t)} = \frac{\sum_{k=1}^n u_{ik}^{(t-1)} (x_k - v_i^{(t)}) (x_k - v_i^{(t)})^T}{\sum_{k=1}^n u_{ik}^{(t-1)}} \tag{12}$$

$$u_{ik}^{(t)} = \frac{\alpha_i (d_{ik}^{(t)})^{\frac{1}{\lambda}}}{\sum_{i=1}^c \alpha_i (d_{ik}^{(t)})^{\frac{1}{\lambda}}} \tag{13}$$

where  $d_{ik}^{(t)} = (\det F_i^{(t)})^{-1} \exp \left[ - (x_k - v_i^{(t)})^T (F_i^{(t)})^{-1} (x_k - v_i^{(t)}) \right]$ .  $v_i^{(t)}$  and  $u_{ik}^{(t)}$  are the  $i$ th cluster center and membership value of  $k$ th sample for the  $i$ th cluster obtained in the  $t$ th iteration, and so on.

We considering the KLFCM clustering algorithm as a map  $U^{(t)} = \theta(U^{(t-1)}) = H(G(U^{(t-1)}))$ , where the mapping function  $G : U = [u_{ik}]_{c \times n} \in M_{fc} \mapsto V = (v_1, v_2, \dots, v_c)^T \in R^{c \times s}$  and  $F : V = (v_1, v_2, \dots, v_c)^T \in R^{c \times s} \mapsto U = [u_{ik}]_{c \times n} \in M_{fc}$  satisfy  $V^{(t)} = G(U^{(t-1)})$  and  $U^{(t)} = H(V^{(t)})$ . Then  $U^{(t)}, t = 1, \dots, n, \dots$  is called the iteration sequence or convergent sequence of the KLFCM algorithm. If the iteration sequence converges to a point  $U^* \in \Omega$ , this

point should be a fixed point of the algorithm which satisfies  $U^* = \theta(U^*)$ . Set the convergence domain of the KLFM clustering algorithm as

$$\Omega = \left\{ U^* \in M_{fc} \mid J_{klfcm}(U^*, G(U^*)) \leq J_{klfcm}(U, G(U^*)), \forall U \in M_{fc}, U \neq U^* \right. \\ \left. \text{and } (U^*, G(U^*)) < J_{klfcm}(U^*, G(U)), \forall G(U) \in R^{c \times s}, G(U) \neq G(U^*) \right\}$$

Clearly, if the iteration process is starting from a point  $U^{(0)} \in M_{fc}$ , then the iteration process will terminate at a point in the convergence domain, or there is a subsequence converges to a point in  $\Omega$ .

If the initial membership matrix is  $U = U^* = [c^{-1}]_{c \times n} \in M_{fc}$ , then the KLFM clustering centers are equal to the mass center of the dataset  $v_i = \bar{x} = \sum_{k=1}^n x_k/n, \forall i$ . In next iteration, we still get  $\bar{x}$  and  $U = [c^{-1}]_{c \times n}$ . That is,  $(\bar{x}, U = [c^{-1}]_{c \times n})$  is actually the fixed point of the KLFM algorithm. If the KLFM algorithm converges to this point, the algorithm will fail to produce meaningful clusters. Moreover, if  $(\bar{x}, U = [c^{-1}]_{c \times n})$  is a stable fixed point of KLFM clustering algorithm, this clustering algorithm will not escape from this point. Of course, this kind of situation should be avoided.

The KLFM clustering result may be heavily influenced by the parameter value of  $\lambda$ , such as shown in Figure 1. However, the KLFM clustering algorithm can avoid outputting the coincident clustering result in  $(\bar{x}, U = [c^{-1}]_{c \times n})$ , which means it is not a stable fixed point of the algorithm. Next, we address the convergence and parameter analysis of the KLFM clustering algorithm using the Jacobian matrix. Our theoretical analysis is based on Olver’s Corollary [23]. According to Olver ([23], p. 143), for Jacobian matrix  $g'(\mu^*) = \frac{dg(\mu)}{d\mu} \Big|_{\mu=\mu^*}$ , if the spectral radius (i.e., the maximum of absolute eigenvalues of the matrix)  $r(g'(\mu^*))$  is less than one, then the fixed point  $\mu^*$  is asymptotically stable. That is, for KLFM, if spectral radius of the Jacobian matrix  $\frac{\partial \theta(U)}{\partial U}$  at point  $U = [c^{-1}]_{c \times n}$  is not less than 1, then  $(\bar{x}, U = [c^{-1}]_{c \times n})$  is not a stable fixed point of the clustering algorithm.

Next, we construct the formula for the element of the Jacobian matrix. The element  $\frac{\partial \theta_{ik}}{\partial u_{jr}}, i = 1, \dots, c, k = 1, \dots, n, j = 1, \dots, c - 1, r = 1, \dots, n$  of Jacobian matrix is obtained by taking the derivations of  $\theta_{ik}$  with respect to  $u_{jr}$ .

**Theorem 1.** For  $i = 1, \dots, c, k = 1, \dots, n$  and  $j = 1, \dots, c - 1, r = 1, \dots, n$ , each element of Jacobian matrix  $\frac{\partial \theta_{ik}}{\partial u_{jr}}$  is

$$\frac{\partial \theta_{ik}}{\partial u_{jr}} = -\frac{u_{ik}u_{jk}}{\lambda n \alpha_j} (H_j)_{kr} + \frac{u_{ik}u_{ck}}{\lambda n \alpha_c} (H_c)_{kr} + \frac{\delta_{ij}u_{ik}}{\lambda n \alpha_i} (H_i)_{kr} \tag{14}$$

where  $(H_j)_{kr} = \lambda + s - (x_r - v_j)^T F_j^{-1} (x_r - v_j) + [(x_k - v_j)^T F_j^{-1} (x_r - v_j)]^2 - (x_k - v_j)^T F_j^{-1} (x_k - v_j) + 2(x_r - v_j)^T F_j^{-1} (x_k - v_j)$  and  $\delta_{ij} = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases}$  is the Kronecker delta function.

**Proof.** Each element of the Jacobian matrix is obtained as follows:

$$\frac{\partial \theta_{ik}}{\partial u_{jr}} = \frac{(\sum_{i=1}^c \alpha_i (d_{ik})^{\frac{1}{\lambda}}) \frac{\partial \alpha_i (d_{ik})^{\frac{1}{\lambda}}}{\partial u_{jr}}}{\left(\sum_{i=1}^c \alpha_i (d_{ik})^{\frac{1}{\lambda}}\right)^2} - \frac{\alpha_i (d_{ik})^{\frac{1}{\lambda}} \left(\frac{\partial \sum_{i=1}^c \alpha_i (d_{ik})^{\frac{1}{\lambda}}}{\partial u_{jr}}\right)}{\left(\sum_{i=1}^c \alpha_i (d_{ik})^{\frac{1}{\lambda}}\right)^2} \tag{15}$$

Recall that the membership matrix of the KLFCM clustering algorithm satisfies  $\sum_{i=1}^c u_{ik} = 1$ , thus, we have:

$$\begin{aligned} \frac{\partial \theta_{ik}}{\partial u_{jr}} &= \frac{\frac{\partial \alpha_i(d_{ik})^{\frac{1}{\lambda}}}{\partial u_{jr}}}{\sum_{i=1}^c \alpha_i(d_{ik})^{\frac{1}{\lambda}}} - \frac{\alpha_i(d_{ik})^{\frac{1}{\lambda}} \left( \frac{\partial \left( \sum_{i=1}^{c-1} \alpha_i(d_{ik})^{\frac{1}{\lambda}} \right)}{\partial u_{jr}} + \frac{\partial \left( \alpha_i(d_{ck})^{\frac{1}{\lambda}} \right)}{\partial u_{jr}} \right)}{\left( \sum_{i=1}^c \alpha_i(d_{ik})^{\frac{1}{\lambda}} \right)^2} \\ &= \frac{\delta_{ij} \frac{\partial \alpha_i(d_{ik})^{\frac{1}{\lambda}}}{\partial u_{jr}}}{\sum_{i=1}^c \alpha_i(d_{ik})^{\frac{1}{\lambda}}} + \frac{\delta_{ij} \alpha_i \frac{\partial (d_{ik})^{\frac{1}{\lambda}}}{\partial u_{jr}}}{\sum_{i=1}^c \alpha_i(d_{ik})^{\frac{1}{\lambda}}} - \frac{\alpha_i(d_{ik})^{\frac{1}{\lambda}} (d_{jk})^{\frac{1}{\lambda}}}{n \left( \sum_{i=1}^c \alpha_i(d_{ik})^{\frac{1}{\lambda}} \right)^2} \\ &\quad - \frac{\alpha_i \alpha_j (d_{ik})^{\frac{1}{\lambda}} \frac{\partial (d_{jk})^{\frac{1}{\lambda}}}{\partial u_{jr}}}{\left( \sum_{i=1}^c \alpha_i(d_{ik})^{\frac{1}{\lambda}} \right)^2} + \frac{\alpha_i(d_{ik})^{\frac{1}{\lambda}} (d_{ck})^{\frac{1}{\lambda}}}{n \left( \sum_{i=1}^c \alpha_i(d_{ik})^{\frac{1}{\lambda}} \right)^2} + \frac{\alpha_i \alpha_c (d_{ik})^{\frac{1}{\lambda}} \frac{\partial (d_{ck})^{\frac{1}{\lambda}}}{\partial u_{cr}}}{\left( \sum_{i=1}^c \alpha_i(d_{ik})^{\frac{1}{\lambda}} \right)^2} \end{aligned} \tag{16}$$

For each element in Equation (16), we can get following result by simple computation:

$$\begin{aligned} \frac{\delta_{ij} \frac{\partial \alpha_i(d_{ik})^{\frac{1}{\lambda}}}{\partial u_{jr}}}{\sum_{i=1}^c \alpha_i(d_{ik})^{\frac{1}{\lambda}}} + \frac{\delta_{ij} \alpha_i \frac{\partial (d_{ik})^{\frac{1}{\lambda}}}{\partial u_{jr}}}{\sum_{i=1}^c \alpha_i(d_{ik})^{\frac{1}{\lambda}}} &= \frac{\delta_{ij} (d_{ik})^{\frac{1}{\lambda}}}{n \sum_{i=1}^c \alpha_i(d_{ik})^{\frac{1}{\lambda}}} - \frac{\delta_{ij} \alpha_i \frac{1}{\lambda} (d_{ik})^{\frac{1}{\lambda}}}{\sum_{i=1}^c \alpha_i(d_{ik})^{\frac{1}{\lambda}}} \frac{\partial (\det F_i)}{\det F_i \partial u_{jr}} \\ &\quad - \frac{\frac{1}{\lambda} \delta_{ij} \alpha_i (d_{ik})^{\frac{1}{\lambda}}}{\sum_{i=1}^c \alpha_i(d_{ik})^{\frac{1}{\lambda}}} (x_k - v_i)^T \frac{\partial (F_i^{-1})}{\partial u_{jr}} (x_k - v_i) \\ &\quad + \frac{2 \delta_{ij} \alpha_i \frac{1}{\lambda} (d_{ik})^{\frac{1}{\lambda}}}{\sum_{i=1}^c \alpha_i(d_{ik})^{\frac{1}{\lambda}} \sum_{k=1}^n u_{ik}} (x_r - v_i)^T F_i^{-1} (x_k - v_i) \\ \frac{\alpha_i \alpha_j (d_{ik})^{\frac{1}{\lambda}} \frac{\partial (d_{jk})^{\frac{1}{\lambda}}}{\partial u_{jr}}}{\left( \sum_{i=1}^c \alpha_i(d_{ik})^{\frac{1}{\lambda}} \right)^2} &= \frac{2 \frac{1}{\lambda} \alpha_i \alpha_j (d_{ik})^{\frac{1}{\lambda}} (d_{jk})^{\frac{1}{\lambda}}}{\left( \sum_{i=1}^c \alpha_i(d_{ik})^{\frac{1}{\lambda}} \right)^2 \sum_{k=1}^n u_{jk}} (x_r - v_j)^T F_j^{-1} (x_k - v_j) \\ &\quad - \frac{\frac{1}{\lambda} \alpha_i \alpha_j (d_{ik})^{\frac{1}{\lambda}} (d_{jk})^{\frac{1}{\lambda}}}{\left( \sum_{i=1}^c \alpha_i(d_{ik})^{\frac{1}{\lambda}} \right)^2} (x_k - v_j)^T \frac{\partial F_j^{-1}}{\partial u_{jr}} (x_k - v_j) - \frac{\frac{1}{\lambda} \alpha_i \alpha_j (d_{ik})^{\frac{1}{\lambda}} (d_{jk})^{\frac{1}{\lambda}}}{\left( \sum_{i=1}^c \alpha_i(d_{ik})^{\frac{1}{\lambda}} \right)^2} (\det F_j)^{-1} \frac{\partial \det F_j}{\partial u_{jr}} \\ \frac{\alpha_i \alpha_c (d_{ik})^{\frac{1}{\lambda}} \frac{\partial (d_{ck})^{\frac{1}{\lambda}}}{\partial u_{cr}}}{\left( \sum_{i=1}^c \alpha_i(d_{ik})^{\frac{1}{\lambda}} \right)^2} &= \frac{2 \frac{1}{\lambda} \alpha_i \alpha_c (d_{ik})^{\frac{1}{\lambda}} (d_{ck})^{\frac{1}{\lambda}}}{\left( \sum_{i=1}^c \alpha_i(d_{ik})^{\frac{1}{\lambda}} \right)^2 \sum_{k=1}^n u_{ck}} (x_r - v_c)^T F_c^{-1} (x_k - v_c) \\ &\quad - \frac{\frac{1}{\lambda} \alpha_i \alpha_c (d_{ik})^{\frac{1}{\lambda}} (d_{ck})^{\frac{1}{\lambda}}}{\left( \sum_{i=1}^c \alpha_i(d_{ik})^{\frac{1}{\lambda}} \right)^2} (x_k - v_c)^T \frac{\partial F_c^{-1}}{\partial u_{cr}} (x_k - v_c) - \frac{\frac{1}{\lambda} \alpha_i \alpha_c (d_{ik})^{\frac{1}{\lambda}} (d_{ck})^{\frac{1}{\lambda}}}{\left( \sum_{i=1}^c \alpha_i(d_{ik})^{\frac{1}{\lambda}} \right)^2} (\det F_c)^{-1} \frac{\partial \det F_c}{\partial u_{cr}} \end{aligned}$$

We have that  $F_i = \frac{\sum_{k=1}^n u_{ik}(x_k - v_i)(x_k - v_i)^T}{\sum_{k=1}^n u_{ik}}$ , so

$$\frac{\partial F_i^{-1}}{\partial u_{ir}} = -F_i^{-1} \frac{\partial F_i}{\partial u_{ir}} F_i^{-1} = -F_i^{-1} \frac{(x_r - v_i)(x_r - v_i)^T}{\sum_{k=1}^n u_{ik}} F_i^{-1} + \frac{F_i^{-1}}{\sum_{k=1}^n u_{ik}}$$

$$\frac{\partial \det F_i}{\partial u_{ir}} = \det F_i \left( \frac{(x_r - v_i)^T F_i^{-1} (x_r - v_i)}{\sum_{k=1}^n u_{ik}} - \frac{s}{\sum_{k=1}^n u_{ik}} \right)$$

Finally, we substitute the above equations into Equation (16). Then the element of the KLFCM Jacobian matrix can be rewritten as

$$\frac{\partial \theta_{ik}}{\partial u_{jr}}$$

$$= \frac{\delta_{ij}(d_{ik})^{\frac{1}{\lambda}}}{n \sum_{i=1}^c \alpha_i (d_{ik})^{\frac{1}{\lambda}}} - \frac{\delta_{ij} \alpha_i \frac{1}{\lambda} (d_{ik})^{\frac{1}{\lambda}}}{\det F_i \sum_{i=1}^c \alpha_i (d_{ik})^{\frac{1}{\lambda}}} \left( \det F_i \left( \frac{(x_r - v_i)^T F_i^{-1} (x_r - v_i)}{\sum_{k=1}^n u_{ik}} - \frac{s}{\sum_{k=1}^n u_{ik}} \right) \right)$$

$$- \frac{\frac{1}{\lambda} \delta_{ij} \alpha_i (d_{ik})^{\frac{1}{\lambda}}}{\sum_{i=1}^c \alpha_i (d_{ik})^{\frac{1}{\lambda}}} (x_k - v_i)^T \left( -F_i^{-1} \frac{(x_r - v_i)(x_r - v_i)^T}{\sum_{k=1}^n u_{ik}} F_i^{-1} + \frac{F_i^{-1}}{\sum_{k=1}^n u_{ik}} \right) (x_k - v_i)$$

$$+ \frac{2\delta_{ij} \alpha_i \frac{1}{\lambda} (d_{ik})^{\frac{1}{\lambda}}}{\sum_{i=1}^c \alpha_i (d_{ik})^{\frac{1}{\lambda}} \sum_{k=1}^n u_{ik}} (x_r - v_i)^T F_i^{-1} (x_k - v_i)$$

$$- \frac{\alpha_i (d_{ik})^{\frac{1}{\lambda}} (d_{jk})^{\frac{1}{\lambda}}}{n \left( \sum_{i=1}^c \alpha_i (d_{ik})^{\frac{1}{\lambda}} \right)^2} - \frac{2\frac{1}{\lambda} \alpha_i \alpha_j (d_{ik})^{\frac{1}{\lambda}} (d_{jk})^{\frac{1}{\lambda}}}{\left( \sum_{i=1}^c \alpha_i (d_{ik})^{\frac{1}{\lambda}} \right)^2 \sum_{k=1}^n u_{jk}} (x_r - v_j)^T F_j^{-1} (x_k - v_j)$$

$$+ \frac{\frac{1}{\lambda} \alpha_i \alpha_j (d_{ik})^{\frac{1}{\lambda}} (d_{jk})^{\frac{1}{\lambda}}}{\left( \sum_{i=1}^c \alpha_i (d_{ik})^{\frac{1}{\lambda}} \right)^2} (x_k - v_j)^T \left( -F_j^{-1} \frac{(x_r - v_j)(x_r - v_j)^T}{\sum_{k=1}^n u_{jk}} F_j^{-1} + \frac{F_j^{-1}}{\sum_{k=1}^n u_{jk}} \right) (x_k - v_j)$$

$$+ \frac{\frac{1}{\lambda} \alpha_i \alpha_j (d_{ik})^{\frac{1}{\lambda}} (d_{jk})^{\frac{1}{\lambda}}}{\det F_j \left( \sum_{i=1}^c \alpha_i (d_{ik})^{\frac{1}{\lambda}} \right)^2} \left( \det F_j \left( \frac{(x_r - v_j)^T F_j^{-1} (x_r - v_j)}{\sum_{k=1}^n u_{jk}} - \frac{s}{\sum_{k=1}^n u_{jk}} \right) \right)$$

$$+ \frac{\alpha_i (d_{ik})^{\frac{1}{\lambda}} (d_{ck})^{\frac{1}{\lambda}}}{n \left( \sum_{i=1}^c \alpha_i (d_{ik})^{\frac{1}{\lambda}} \right)^2} + \frac{2\frac{1}{\lambda} \alpha_i \alpha_c (d_{ik})^{\frac{1}{\lambda}} (d_{ck})^{\frac{1}{\lambda}}}{\left( \sum_{i=1}^c \alpha_i (d_{ik})^{\frac{1}{\lambda}} \right)^2 \sum_{k=1}^n u_{ck}} (x_r - v_c)^T F_c^{-1} (x_k - v_c)$$

$$- \frac{\frac{1}{\lambda} \alpha_i \alpha_c (d_{ik})^{\frac{1}{\lambda}} (d_{ck})^{\frac{1}{\lambda}}}{\left( \sum_{i=1}^c \alpha_i (d_{ik})^{\frac{1}{\lambda}} \right)^2} (x_k - v_c)^T \left( -F_c^{-1} \frac{(x_r - v_c)(x_r - v_c)^T}{\sum_{k=1}^n u_{ck}} F_c^{-1} + \frac{F_c^{-1}}{\sum_{k=1}^n u_{ck}} \right) (x_k - v_c)$$

$$- \frac{\frac{1}{\lambda} \alpha_i \alpha_c (d_{ik})^{\frac{1}{\lambda}} (d_{ck})^{\frac{1}{\lambda}}}{\det F_c \left( \sum_{i=1}^c \alpha_i (d_{ik})^{\frac{1}{\lambda}} \right)^2} \left( \det F_c \left( \frac{(x_r - v_c)^T F_c^{-1} (x_r - v_c)}{\sum_{k=1}^n u_{ck}} - \frac{s}{\sum_{k=1}^n u_{ck}} \right) \right)$$

We further simplify the formula and get

$$\begin{aligned} \frac{\partial \theta_{ik}}{\partial u_{jr}} &= \frac{\delta_{ij} u_{ik}}{n \alpha_i} - \frac{\delta_{ij} u_{ik}}{\lambda} \left( \frac{(x_r - v_i)^T F_i^{-1} (x_r - v_i)}{\sum_{k=1}^n u_{ik}} - \frac{s}{\sum_{k=1}^n u_{ik}} \right) \\ &\quad - \frac{\delta_{ij} u_{ik}}{\lambda} (x_k - v_i)^T \left( -F_i^{-1} \frac{(x_r - v_i)(x_r - v_i)^T}{\sum_{k=1}^n u_{ik}} F_i^{-1} + \frac{F_i^{-1}}{\sum_{k=1}^n u_{ik}} \right) (x_k - v_i) \\ &\quad + \frac{2 \delta_{ij} u_{ik}}{\lambda \sum_{k=1}^n u_{ik}} (x_r - v_i)^T F_i^{-1} (x_k - v_i) - \frac{u_{ik} u_{jk}}{n \alpha_j} - \frac{2 u_{ik} u_{jk}}{\lambda \sum_{k=1}^n u_{jk}} (x_r - v_j)^T F_j^{-1} (x_k - v_j) \\ &\quad + \frac{u_{ik} u_{jk}}{\lambda} (x_k - v_j)^T \left( -F_j^{-1} \frac{(x_r - v_j)(x_r - v_j)^T}{\sum_{k=1}^n u_{jk}} F_j^{-1} + \frac{F_j^{-1}}{\sum_{k=1}^n u_{jk}} \right) (x_k - v_j) \\ &\quad + \frac{u_{ik} u_{jk}}{\lambda} \left( \frac{(x_r - v_j)^T F_j^{-1} (x_r - v_j)}{\sum_{k=1}^n u_{jk}} - \frac{s}{\sum_{k=1}^n u_{jk}} \right) \\ &\quad + \frac{u_{ik} u_{jk}}{n \alpha_c} + \frac{2 u_{ik} u_{ck}}{\lambda \sum_{k=1}^n u_{ck}} (x_r - v_c)^T F_c^{-1} (x_k - v_c) \\ &\quad - \frac{u_{ik} u_{ck}}{\lambda} (x_k - v_c)^T \left( -F_c^{-1} \frac{(x_r - v_c)(x_r - v_c)^T}{\sum_{k=1}^n u_{ck}} F_c^{-1} + \frac{F_c^{-1}}{\sum_{k=1}^n u_{ck}} \right) (x_k - v_c) \\ &\quad - \frac{u_{ik} u_{ck}}{\lambda} \left( \frac{(x_r - v_c)^T F_c^{-1} (x_r - v_c)}{\sum_{k=1}^n u_{ck}} - \frac{s}{\sum_{k=1}^n u_{ck}} \right) \\ &= \frac{\delta_{ij} u_{ik}}{\lambda n \alpha_i} \left( \lambda + s - (x_r - v_i)^T F_i^{-1} (x_r - v_i) + 2(x_r - v_i)^T F_i^{-1} (x_k - v_i) \right. \\ &\quad \left. - (x_k - v_i)^T \left( -F_i^{-1} (x_r - v_i)(x_r - v_i)^T F_i^{-1} + F_i^{-1} \right) (x_k - v_i) \right) \\ &\quad - \frac{u_{ik} u_{jk}}{\lambda n \alpha_j} \left( \lambda + s - (x_r - v_j)^T F_j^{-1} (x_r - v_j) + 2(x_r - v_j)^T F_j^{-1} (x_k - v_j) \right. \\ &\quad \left. - (x_k - v_j)^T \left( -F_j^{-1} (x_r - v_j)(x_r - v_j)^T F_j^{-1} + F_j^{-1} \right) (x_k - v_j) \right) \\ &\quad + \frac{u_{ik} u_{ck}}{\lambda n \alpha_c} \left( \lambda + s - (x_r - v_c)^T F_c^{-1} (x_r - v_c) + 2(x_r - v_c)^T F_c^{-1} (x_k - v_c) \right. \\ &\quad \left. - (x_k - v_c)^T \left( -F_c^{-1} (x_r - v_c)(x_r - v_c)^T F_c^{-1} + F_c^{-1} \right) (x_k - v_c) \right) \end{aligned}$$

Set  $(H_j)_{kr} = \lambda + s - (x_r - v_j)^T F_j^{-1} (x_r - v_j) + [(x_k - v_j)^T F_j^{-1} (x_r - v_j)]^2 - (x_k - v_j)^T F_j^{-1} (x_k - v_j) + 2(x_r - v_j)^T F_j^{-1} (x_k - v_j)$ . Then each element in the Jacobian matrix is:

$$\frac{\partial \theta_{ik}}{\partial u_{jr}} = -\frac{u_{ik} u_{jk}}{\lambda n \alpha_j} (H_j)_{kr} + \frac{u_{ik} u_{ck}}{\lambda n \alpha_c} (H_c)_{kr} + \frac{\delta_{ij} u_{ik}}{\lambda n \alpha_i} (H_i)_{kr}$$

The proof is completed. □

Now, we get a general form for the Jacobian matrix. To discuss the theoretical behavior of the KLFCM clustering algorithm, we should consider the Jacobian matrix  $\frac{\partial \theta(U)}{\partial U}$  at the special point  $U = [c^{-1}]_{c \times n}$ . We define a notation as follows: For any matrix  $M = (m_1, \dots, m_q)_{p \times q}$ ,  $vec(M) = (m_1^T, \dots, m_q^T)^T$ .

**Theorem 2.** Each element of Jacobian matrix  $\frac{\partial \theta(U)}{\partial U}$  at the special point  $U = [c^{-1}]_{c \times n}$  is

$$\frac{\partial \theta_{ik}}{\partial u_{jr}} \Big|_{\forall i,k,u_{ik}=c^{-1}} = \frac{\delta_{ij}}{n \lambda} (A^T A)_{kr} \tag{17}$$

where

$$A = \begin{bmatrix} \sqrt{\lambda} & \dots & \sqrt{\lambda} \\ \sqrt{2}\sigma_X^{-1/2}(x_1 - \bar{x}) & \dots & \sqrt{2}\sigma_X^{-1/2}(x_n - \bar{x}) \\ \text{vec}\left(\sigma_X^{-1/2}(x_1 - \bar{x})(x_1 - \bar{x})^T \sigma_X^{-1/2} - I_s\right) & \dots & \text{vec}\left(\sigma_X^{-1/2}(x_n - \bar{x})(x_n - \bar{x})^T \sigma_X^{-1/2} - I_s\right) \end{bmatrix}$$

and  $\bar{x} = \sum_{k=1}^n x_k/n, \sigma_x = n^{-1} \sum_{k=1}^n (x_k - \bar{x})(x_k - \bar{x})^T$ ,  $\delta_{ij}$  is the Kronecker delta function.

**Proof.** If  $U = [c^{-1}]_{c \times n}$ , then  $u_{ik} = \frac{1}{c}$ ,  $v_i = \frac{\sum_{k=1}^n x_k}{n} = \bar{x}$  and  $F_i = \frac{\sum_{k=1}^n (x_k - \bar{x})(x_k - \bar{x})^T}{n} = \sigma_x$ . Thus, the Jacobian matrix at this special point becomes

$$\begin{aligned} & \frac{\partial \theta_{ik}}{\partial u_{jr}} \Big|_{v_i, k, u_{ik} = c^{-1}} \\ &= \frac{\delta_{ij}\lambda}{n\lambda} - \frac{\delta_{ij}}{\lambda n} (x_r - \bar{x})^T \sigma_X^{-1} (x_r - \bar{x}) + \frac{\delta_{ij}s}{\lambda n} + \frac{\delta_{ij}}{n\lambda} \left[ (x_k - \bar{x})^T \sigma_X^{-1} (x_r - \bar{x}) \right]^2 \\ & - \frac{\delta_{ij}}{n\lambda} (x_k - \bar{x})^T \sigma_X^{-1} (x_k - \bar{x}) + \frac{2\delta_{ij}}{\lambda n} (x_r - \bar{x})^T F_i^{-1} (x_k - \bar{x}) \\ &= \frac{\delta_{ij}}{n\lambda} \left( \begin{aligned} & -(x_r - \bar{x})^T \sigma_X^{-1} (x_r - \bar{x}) + \lambda + s + 2(x_r - \bar{x})^T F_i^{-1} (x_k - \bar{x}) \\ & + \left[ (x_k - \bar{x})^T \sigma_X^{-1} (x_r - \bar{x}) \right]^2 - (x_k - \bar{x})^T \sigma_X^{-1} (x_k - \bar{x}) \end{aligned} \right) \\ &= \frac{\delta_{ij}}{n\lambda} \left( \begin{aligned} & \lambda + 2(x_r - \bar{x})^T F_i^{-1} (x_k - \bar{x}) - (x_r - \bar{x})^T \sigma_X^{-1} (x_r - \bar{x}) + s \\ & + \left[ (x_k - \bar{x})^T \sigma_X^{-1} (x_r - \bar{x}) \right]^2 - (x_k - \bar{x})^T \sigma_X^{-1} (x_k - \bar{x}) \end{aligned} \right) \end{aligned} \tag{18}$$

Consider that  $tr(a^T b) = tr(b^T a)$  where  $a$  and  $b$  are column vectos, also we have  $tr(ABC) = tr(BCA) = tr(CAB)$  for matrices  $A, B$ , and  $C$ , the following equations can be obtained by simple computation:

$$\begin{aligned} (x_r - \bar{x})^T \sigma_X^{-1} (x_r - \bar{x}) &= tr \left( \sigma_X^{-\frac{1}{2}} (x_r - \bar{x})(x_r - \bar{x})^T \sigma_X^{-\frac{1}{2}} \right) \\ (x_k - \bar{x})^T \sigma_X^{-1} (x_k - \bar{x}) &= tr \left( \sigma_X^{-\frac{1}{2}} (x_k - \bar{x})(x_k - \bar{x})^T \sigma_X^{-\frac{1}{2}} \right) \\ \left[ (x_k - \bar{x})^T \sigma_X^{-1} (x_r - \bar{x}) \right]^2 &= (x_r - \bar{x})^T \sigma_X^{-1} (x_k - \bar{x})(x_k - \bar{x})^T \sigma_X^{-1} (x_r - \bar{x}) \\ &= \left( (x_r - \bar{x})^T \sigma_X^{-\frac{1}{2}} \right) \left( \sigma_X^{-\frac{1}{2}} (x_k - \bar{x})(x_k - \bar{x})^T \sigma_X^{-\frac{1}{2}} \right) \left( \sigma_X^{-\frac{1}{2}} (x_r - \bar{x}) \right) \\ &= tr \left( \sigma_X^{-\frac{1}{2}} (x_k - \bar{x})(x_k - \bar{x})^T \sigma_X^{-\frac{1}{2}} \sigma_X^{-\frac{1}{2}} (x_r - \bar{x})(x_r - \bar{x})^T \sigma_X^{-\frac{1}{2}} \right) \end{aligned}$$

Equation (18) can be further simplified as

$$\begin{aligned} & \frac{\partial \theta_{ik}}{\partial u_{jr}} \Big|_{\forall i,k,u_{ik}=c^{-1}} \\ &= \frac{\delta_{ij}}{n\lambda} \left( \begin{array}{l} \lambda + 2(x_r - \bar{x})^T F_i^{-1}(x_k - \bar{x}) - \text{tr} \left( \sigma_X^{-\frac{1}{2}}(x_r - \bar{x})(x_r - \bar{x})^T \sigma_X^{-\frac{1}{2}} \right) \\ + \text{tr}(I_s) + \text{tr} \left( \sigma_X^{-\frac{1}{2}}(x_k - \bar{x})(x_k - \bar{x})^T \sigma_X^{-\frac{1}{2}} \sigma_X^{-\frac{1}{2}}(x_r - \bar{x})(x_r - \bar{x})^T \sigma_X^{-\frac{1}{2}} \right) \\ - \text{tr} \left( \sigma_X^{-\frac{1}{2}}(x_k - \bar{x})(x_k - \bar{x})^T \sigma_X^{-\frac{1}{2}} \right) \end{array} \right) \\ &= \frac{\delta_{ij}}{n\lambda} \left( \begin{array}{l} \lambda + 2(x_r - \bar{x})^T F_i^{-1}(x_k - \bar{x}) \\ + \text{tr} \left( \left( \sigma_X^{-\frac{1}{2}}(x_r - \bar{x})(x_r - \bar{x})^T \sigma_X^{-\frac{1}{2}} - I_s \right) \left( \sigma_X^{-\frac{1}{2}}(x_k - \bar{x})(x_k - \bar{x})^T \sigma_X^{-\frac{1}{2}} - I_s \right) \right) \end{array} \right) \\ &= \frac{\delta_{ij}}{n\lambda} \left( \begin{array}{l} \lambda + 2(x_r - \bar{x})^T F_i^{-1}(x_k - \bar{x}) \\ + \text{tr} \left( \left( \sigma_X^{-\frac{1}{2}}(x_r - \bar{x})(x_r - \bar{x})^T \sigma_X^{-\frac{1}{2}} - I_s \right)^T \left( \sigma_X^{-\frac{1}{2}}(x_k - \bar{x})(x_k - \bar{x})^T \sigma_X^{-\frac{1}{2}} - I_s \right) \right) \end{array} \right) \end{aligned}$$

The trace of the matrix  $A^T B$  can be interpreted as  $\text{tr}(A^T B) = \sum_{i=1}^n \sum_{j=1}^n a_{ij} b_{ij}$ , where  $a_{ij}$  and  $b_{ij}$  denote the element of row  $i$  and column  $j$  in  $A_{n \times n}$  and  $B_{n \times n}$  respectively. Moreover, we have that  $\text{vec}(A) = [a_{11}, \dots, a_{n1}, a_{12}, \dots, a_{n2}, \dots, a_{1n}, \dots, a_{nn}]^T$  and  $\text{vec}(B) = [b_{11}, \dots, b_{n1}, b_{12}, \dots, b_{n2}, \dots, b_{1n}, \dots, b_{nn}]^T$ . It implies that  $\text{tr}(A^T B) = \sum_{i=1}^n \sum_{j=1}^n a_{ij} b_{ij}$ , where  $a_{ij}$  and  $b_{ij} = \text{vec}(A)^T \times \text{vec}(B)$ . Finally, we have

$$\begin{aligned} \frac{\partial \theta_{ik}}{\partial u_{jr}} \Big|_{\forall i,k,u_{ik}=c^{-1}} &= \frac{\delta_{ij}}{n\lambda} \left( \lambda + 2(x_r - \bar{x})^T \sigma_X^{-1}(x_k - \bar{x}) \right. \\ & \left. + \text{vec} \left( \sigma_X^{-1/2}(x_k - \bar{x})(x_k - \bar{x})^T \sigma_X^{-1/2} - I_s \right)^T \times \text{vec} \left[ \left( \sigma_X^{-1/2}(x_r - \bar{x})(x_r - \bar{x})^T \sigma_X^{-1/2} - I_s \right) \right] \right) \end{aligned}$$

Let

$$H = \left[ \begin{array}{l} \lambda + 2(x_r - \bar{x})^T \sigma_X^{-1}(x_k - \bar{x}) \\ + \text{vec} \left( \sigma_X^{-1/2}(x_k - \bar{x})(x_k - \bar{x})^T \sigma_X^{-1/2} - I_s \right)^T \times \text{vec} \left[ \left( \sigma_X^{-1/2}(x_r - \bar{x})(x_r - \bar{x})^T \sigma_X^{-1/2} - I_s \right) \right] \end{array} \right]$$

We have that

$$\begin{aligned} (A^T A)_{kr} &= \left[ \lambda + 2(x_r - \bar{x})^T \sigma_X^{-1}(x_k - \bar{x}) \right. \\ & \left. + \text{vec} \left( \sigma_X^{-1/2}(x_k - \bar{x})(x_k - \bar{x})^T \sigma_X^{-1/2} - I_s \right)^T \times \text{vec} \left[ \left( \sigma_X^{-1/2}(x_r - \bar{x})(x_r - \bar{x})^T \sigma_X^{-1/2} - I_s \right) \right] \right] \\ &= (H)_{kr} \end{aligned}$$

Then the element in the Jacobian matrix  $\frac{\partial \theta(U)}{\partial U}$  at the special point  $U = [c^{-1}]_{c \times n}$  is

$$\frac{\partial \theta_{ik}}{\partial u_{jr}} \Big|_{\forall i,k,u_{ik}=c^{-1}} = \frac{\delta_{ij}}{n\lambda} (H)_{kr} = \frac{\delta_{ij}}{n\lambda} (A^T A)_{kr}$$

The proof is completed.  $\square$

We have mentioned that the spectral radius of the Jacobian matrix can reflect the theoretical behavior of the algorithm. For KLFCM algorithm, if the spectral radius of the Jacobian matrix  $\frac{\partial \theta(U)}{\partial U}$  at point  $(\bar{x}, U = [c^{-1}]_{c \times n})$  is not less than 1, then it is not a stable fixed point of the algorithm. Next, we focus on the spectral radius of the Jacobian matrix calculated by Equation (17).

**Theorem 3.** Let  $r^*$  denote the spectral radius of Jacobian matrix  $\frac{\partial \theta}{\partial U} |_{\forall i,k,u_{ik}=c^{-1}}$ , then we have that  $r^* \geq 1$ .

**Proof.** Because of the eigenvalues of matrix  $A^T A$  are equal to the eigenvalues of matrix  $A A^T$ , the spectral radius of matrix  $A A^T$  is the same as that of  $A^T A$ . That is, the spectral radius of Jacobian matrix  $\frac{\partial \theta}{\partial U} |_{\forall i,k,u_{ik}=c^{-1}}$  is equal to the spectral radius of symmetric matrix  $\frac{1}{n\lambda}(A \times A^T)$  computed by Equation (19).

$$\frac{1}{n\lambda}(A \times A^T) = \frac{1}{n\lambda} \begin{pmatrix} L_{11} & L_{12} & L_{13} \\ L_{21} & L_{22} & L_{23} \\ L_{31} & L_{32} & L_{33} \end{pmatrix} = \frac{1}{n\lambda} \begin{pmatrix} \lambda n & 0 & 0 \\ 0 & 2n \times I_s & L_{23} \\ 0 & L_{32} & L_{33} \end{pmatrix} \tag{19}$$

where

$$\begin{aligned} L_{11} &= \lambda n, L_{12} = \sqrt{2\lambda} \sum_{k=1}^n (x_k - \bar{x})^T \sigma_X^{-1/2} = 0 \\ L_{13} &= \sqrt{\lambda} \sum_{k=1}^n \text{vec} \left[ \sigma_X^{-1/2} (x_k - \bar{x})(x_k - \bar{x})^T \sigma_X^{-1/2} - I_s \right]^T = 0 \\ L_{12} &= L_{21}^T, L_{13} = L_{31}^T, L_{22} = 2 \sum_{k=1}^n \sigma_X^{-1/2} (x_k - \bar{x})(x_k - \bar{x})^T \sigma_X^{-1/2} = 2n \times I_s \\ L_{23} &= \sqrt{2} \sum_{k=1}^n \sigma_X^{-1/2} (x_k - \bar{x}) \times \text{vec} \left[ \sigma_X^{-1/2} (x_k - \bar{x})(x_k - \bar{x})^T \sigma_X^{-1/2} - I_s \right]^T \\ L_{23} &= L_{32}^T \\ L_{33} &= \sum_{k=1}^n \text{vec} \left[ \sigma_X^{-1/2} (x_k - \bar{x})(x_k - \bar{x})^T \sigma_X^{-1/2} - I_s \right] \times \text{vec} \left[ \sigma_X^{-1/2} (x_k - \bar{x})(x_k - \bar{x})^T \sigma_X^{-1/2} - I_s \right]^T \end{aligned}$$

It is true that for the symmetric matrix  $\frac{1}{n\lambda}(A \times A^T)$ , the following statement holds.

$$k_{\max} \left( \frac{1}{n\lambda}(A \times A^T) \right) = \max_{x \neq 0} \frac{x^T \left( \frac{1}{n\lambda}(A \times A^T) \right) x}{x^T x}$$

where the symbol  $k_{\max}$  represents the maximum eigenvalue of the matrix. Let  $e^{(i)} = \{0, \dots, 1, \dots, 0\}$  is a vector in which the  $i$ th element is one and other elements are zero. Obviously, we have the following inequality holds:

$$\max_{x \neq 0} \frac{x^T \left( \frac{1}{n\lambda}(A \times A^T) \right) x}{x^T x} \geq \frac{[e^{(i)}]^T \left( \frac{1}{n\lambda}(A \times A^T) \right) e^{(i)}}{[e^{(i)}]^T e^{(i)}} = \left( \frac{1}{n\lambda}(A \times A^T) \right)_{ii}$$

Based on the above analysis, we can conclude that  $r^* \geq 1$ .  $\square$

The study reveals that the coincident clustering results  $(\bar{x}, U = [c^{-1}]_{c \times n})$  are not stable fixed points of the KLFCM algorithm. As an example, when analyzing the Iris dataset, inappropriate selection of the fuzziness parameter may lead to all data points being assigned to a single cluster in the clustering result, as shown in Figure 1. Despite this clustering outcome being incorrect, the KLFCM algorithm avoids outputting the coincident clustering result, where all cluster centers are equal to the sample mean.

### 4. Experimental Results

In this section, we validate our theoretical results through experimental examples. We use both artificial and real datasets to demonstrate that the KLFCM clustering algorithm may exhibit self-annealing properties when selecting a suitable fuzzy parameter  $\lambda$ . We calculate the spectral radius of the Jacobian matrix at the coincident clustering result for the KLFCM algorithm under different lambda parameter conditions. A spectral radius greater than 1 indicates that the coincident clustering result is not a stable fixed point of the clustering algorithm. In addition, we found that the spectral radius of the Jacobian matrix can be applied in analyzing the convergence rate of the KLFCM algorithm. In all examples, the results of the K-means algorithm are used as the initialization for the KLFCM clustering algorithm. Let  $r^*$  denote the spectral radius of Jacobian matrix  $\frac{\partial \theta}{\partial U}$  at point  $(\bar{x}, U = [c^{-1}]_{c \times n})$ . Subsequently, we conducted this experiment through a MATLAB model running on Windows 11 with the version of MATLAB being R2022a.

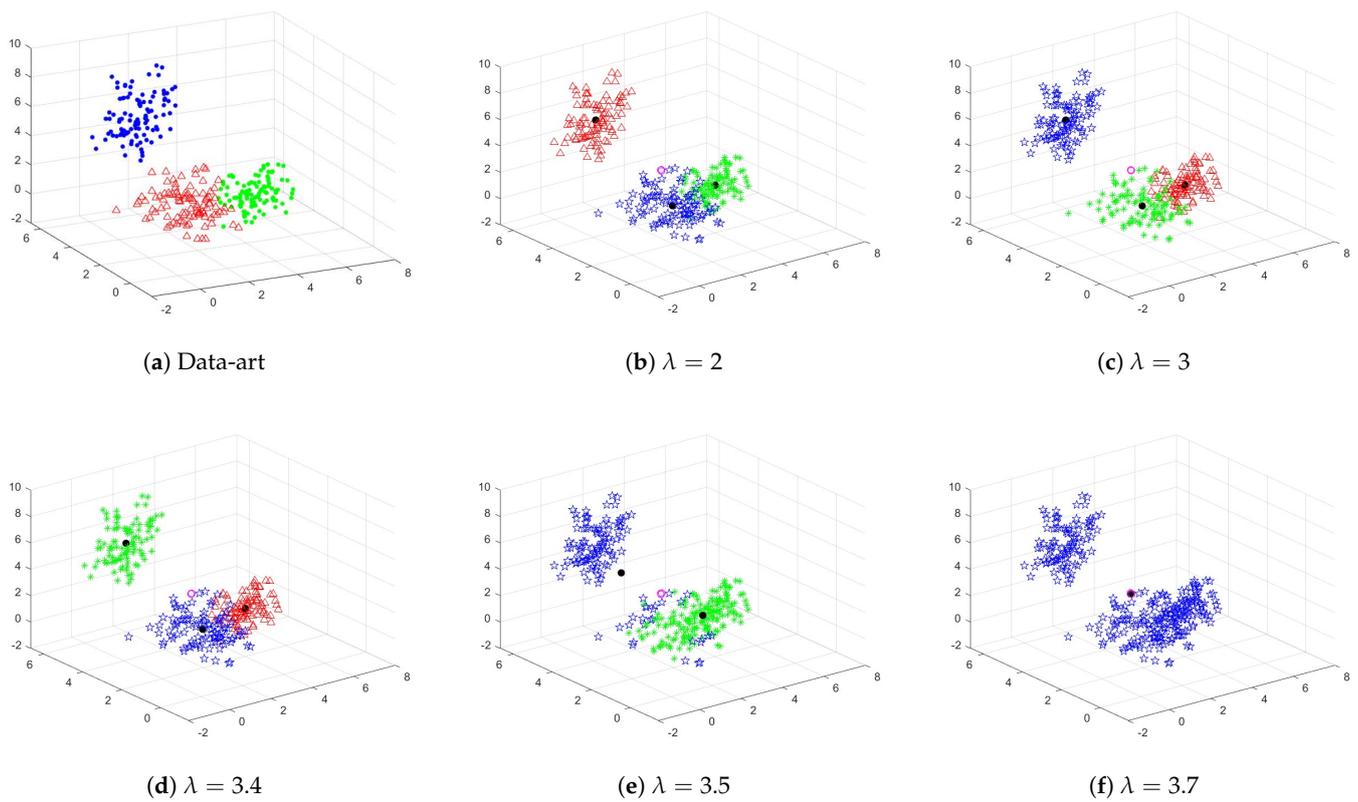
**Example 1.** First, we synthesized GMM data with three clusters. The mixing proportions, mean values, and variances are listed in Table 1. The total number of data points is 300. The artificial dataset named Data-art is shown in Figure 3a. Data points generated from different models are denoted by different shapes, such as \* (green),  $\Delta$  (red) and  $\star$  (blue).

**Table 1.** Mixing Proportions, Means Values and Variances of Gaussian Mixture Models to Generate Data-art.

Mixing Proportions	Mean Values	Variances
$\alpha_1 = 1/3$	$m_1 = (1, 1, 2)$	$\Sigma_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
$\alpha_2 = 1/3$	$m_2 = (5, 3, 0.5)$	$\Sigma_1 = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
$\alpha_3 = 1/3$	$m_3 = (2, 6, 5)$	$\Sigma_1 = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 2 \end{bmatrix}$

After initializing the KLFCM clustering algorithm with manually given cluster centers, we choose different  $\lambda$  values to observe their influence on the clustering result. The clustering results corresponding to different  $\lambda$  values are listed in Figure 3b–f.

To illustrate the clustering outcome, we use different colors and shapes to signify data points that belong to different clusters. Furthermore, we represent the initial cluster centers with  $\circ$  (magenta circles). It can be seen from Figure 3 that, although the three clustering centers are very close during initialization, as long as we choose appropriate fuzzy parameters, the KLFCM algorithm can still produce relatively good clustering results through iteration. This is demonstrated in Figure 3b–d. We observed that as the value of the fuzzy parameter  $\lambda$  increases, the KLFCM algorithm may produce clustering results where all samples belong to the same cluster. For example, when  $\lambda = 3.7$ . We further calculated that when  $\lambda = 3.7$ , the spectral radius of the Jacobian matrix at coincident clustering result  $(\bar{x}, U = [c^{-1}]_{c \times n})$  is equal to 1,  $r^* = 1$ . In other words, since the KLFCM algorithm inherently exhibits self-annealing properties, choosing appropriate parameters that satisfy  $r^* > 1$  will ensure that the algorithm produces interpretable and acceptable clustering results with any initial cluster centers, except for the case where all initial cluster centers are set equal to the sample mean. This finding is particularly intriguing. Next, we present more experimental results on real datasets.



**Figure 3.** Clustering Results of the Data-art Dataset with Different  $\lambda$ .

**Example 2.** We conduct experiments on six datasets from UCI Machine Learning Repository. The datasets we used in our experiments are described in Table 2.

**Table 2.** Experiments on Real Datasets from UCI Databases.

Datasets	Sample No. n	Feature No. s	Cluster No. c	$r^*(\lambda = 2)$
Iris	150	4	3	2.08364
Cloud	1024	10	3	50.07438
Wine	178	13	3	9.94855
Haberman’s Survival	306	3	2	1.49047
seeds	210	7	3	4.76313
Sonar	208	60	2	9.07394
Balance Scale	625	4	3	1.00000

We have theoretically proved the spectral radius of Jacobian matrix at the special point  $(\bar{x}, U = [c^{-1}]_{c \times n})$  is not less than 1,  $r^* \geq 1$ . That is,  $(\bar{x}, U = [c^{-1}]_{c \times n})$  is not a stable fixed point of the KLFCM clustering algorithm. Our previous analysis reveals that for the KLFCM algorithm, the spectral radius of its Jacobian matrix at  $(\bar{x}, U = [c^{-1}]_{c \times n})$  is solely dependent on the fuzziness parameter  $\lambda$  and the data, while it remains unaffected by the initial clustering center of the algorithm. If the lambda value we choose ensures  $r^* > 1$ , then the clustering algorithm is likely to output good clustering results despite poorly choosing initial clustering centers, thanks to its self-annealing property.

By employing Equation (19), we have computed the spectral radius  $r^*$  for different  $\lambda$  values, and the corresponding results are showcased in Table 3.

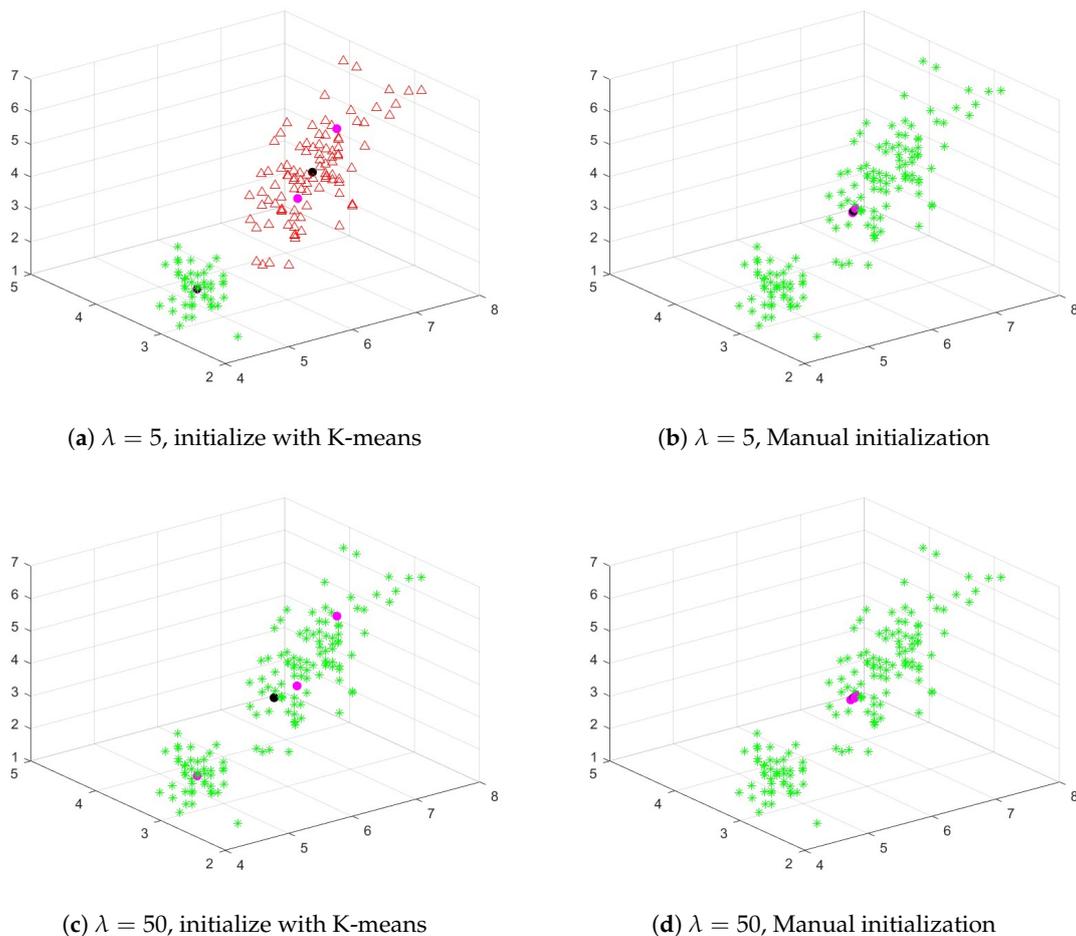
**Table 3.** Spectral Radius  $r^*$  of the Jacobian Matrix Corresponding to Different  $\lambda$ .

$\lambda$	2	3	3.5	4	4.5	5	6	7	8	9	10
Iris	2.08364	1.38909	1.19065	1.04182	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Cloud	50.07438	33.38292	28.61393	25.03719	22.25528	20.02975	16.69146	14.30697	12.51860	11.12764	10.01488
Wine *	9.94855	6.63236	5.68488	4.97427	4.42158	3.97942	–	–	–	–	–
Haberman’s											
Survival	7.45235	4.96823	4.25848	3.72617	3.31215	2.98094	2.48412	2.12924	1.86309	1.65608	1.49047
seeds	4.76313	3.17542	2.72179	2.38157	2.11695	1.90525	1.58771	1.36089	1.19078	1.05847	1.0000
Sonar	45.36970	30.24647	25.92554	22.68485	20.16431	18.14788	15.12323	12.96277	11.34243	10.08216	9.07394
Balance Scale	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

\* It should be mentioned that the FLFCM clustering algorithm may fail to cluster data for the Wine dataset when  $\lambda \geq 6$ .

It can be seen from Table 3 that the spectral radius of Jacobian matrix at the special point  $(\bar{x}, U = [c^{-1}]_{c \times n})$  is not smaller than 1 for any fuzziness index value, which is consistent with the result of our theoretical analysis. We select parameter values that satisfy  $r^* = 1$ , such as  $\lambda = 5$  or  $\lambda = 50$ , and employ the K-means clustering algorithm and manual initialization to set the initial cluster centers of the KLFCM algorithm. Next, we apply the KLFCM clustering algorithm with different initialization methods to cluster the Iris dataset.

The clustering results are depicted in Figure 4. The ● (magenta circles) and ● (black circles), respectively, represent the initial cluster centers and the cluster centers obtained after clustering.



**Figure 4.** The Clustering Results of KLFCM Algorithm with Different Initialization Methods on the Iris Dataset.

We have observed that when the spectral radius is equal to 1, several clusters are merged into a new cluster. However, different results can be obtained using different initialization methods. For instance, when  $\lambda = 5$  and we use the K-means algorithm to initialize KLFCM, two clusters are obtained in the clustering results, which also preserve some structural information of the original data. On the other hand, if we initialize the KLFCM algorithm manually, all the samples in the clustering results will belong to the same cluster. If the value of the lambda parameter is large enough, for example,  $\lambda = 50$ , regardless of the initialization method used, the KLFCM algorithm may output clustering results where all samples belong to the same cluster. Interestingly enough, the KLFCM algorithm is not suitable for the Balance Scale clustering issue because  $r^* = 1$  under any parameter condition.

Therefore, for the case where the spectral radius is equal to 1, as shown in Table 3, does the clustering algorithm avoid outputting overlapping clustering results? Next, we borrow the non-fuzzy index (NFI) to make a more advanced analysis. The NFI index is proposed by Roubens [24].

$$NFI(c, U, \lambda) = (c/(n \times (c - 1))) \sum_{i=1}^c \sum_{k=1}^n u_{ik}^2 - 1/(c - 1)$$

These NFIs can be used to compare the performances of the clustering results. It is obvious that if the membership values are close to 0 or 1, then the NFI index will be close to 1. Otherwise, if  $U = [c^{-1}]_{c \times n}$ , the NFI index will then be close to 0. In other words,  $NFI(c, U, \lambda) = 0$  indicates that the algorithm outputs the coincident clustering result  $[\bar{x}, U = [c^{-1}]_{c \times n}]$ . That is,

$$NFI(c, U, \lambda) = \begin{cases} 0, & \text{if } U = [c^{-1}]_{c \times n} \text{ and } V = \bar{x} \\ 1, & \text{if hard clustering results} \end{cases}$$

We initialized the KLFCM clustering algorithm with the K-means clustering algorithm and calculated the NFI value of the resulting clustering. The results are shown in Table 4.

**Table 4.** NFI Values Corresponding to Different  $\lambda$  for Different Real Datasets.

$\lambda$	2	3	3.5	4	4.5	5	6	7	8	9	10
Iris	0.9749	0.9811	0.6978	0.5871	0.5079	0.5049	0.4956	0.3656	0.2515	0.2512	0.2511
Cloud	0.8846	0.8170	0.7769	0.7440	0.7110	0.6771	0.7064	0.6571	0.5952	0.7368	0.5581
Wine *	0.9618	0.8998	0.9256	0.9587	0.9600	0.9806	–	–	–	–	–
Haberman’s Survival	0.8861	0.7323	0.6548	0.5827	0.5179	0.3752	0.2784	0.2372	0.2256	0.1973	0.4192
seeds	0.9956	0.9673	0.9832	0.9773	0.9691	0.9458	0.9192	0.9417	0.7818	0.9546	0.8540
Sonar	0.9999	0.9990	0.9998	0.9996	1.0000	0.9996	0.9969	0.9999	0.9949	0.9932	0.9946
Balance Scale	0.7794	0.0895	0.1249	0.0791	0.0709	0.1258	0.0197	0.0053	0.0057	0.0458	0.0026

\* It should be mentioned that the FLFCM clustering algorithm may fail to cluster data for the Wine dataset when  $\lambda \geq 6$ .

Table 4 shows that under the parameter conditions listed, the NFI value of the KLFCM clustering results is almost always greater than 0. We further find that as the values of  $\lambda$  increase, the KLFCM clustering result has the NFI values with a decreasing trend. For the Iris dataset, if  $\lambda = 8$ , then the spectral radius  $r^*$  will be equal to 1 (see Table 3). We found that all data points are assigned to one cluster in this situation (see Figure 1). However, the NFI value is greater than 0, which means that the KLFCM clustering algorithm did not output the coincident clustering results in  $(\bar{x}, U = [c^{-1}]_{c \times n})$ . In fact, the KLFCM algorithm may output two cluster centers.

The above results indicate that even if the fuzzy parameters we choose are not optimal enough, the KLFCM algorithm has self-annealing property, which enables the algorithm to avoid producing coincident clustering centers at convergence. Even in some cases where the spectral radius  $r^*$  is equal to 1, the KLFCM clustering results can still capture aspects of the underlying data structure because K-means initialization allows initial clustering

centers to be distributed across different regions of the data. However, when the value of parameter  $\lambda$  used in the KLFCM clustering algorithm is too large, for example,  $\lambda = 50$ , when clustering the Iris dataset, the algorithm may fail to distinguish between overlapping clusters and produce inaccurate clustering results.

**Example 3.** In this example, we further discuss the impact of parameter  $\lambda$  on the convergence rate of the KLFCM clustering algorithm. We have previously regarded the KK algorithm as a mapping  $U^{(t)} = \theta(U^{(t-1)})$ . If we assume that  $U^{(t)}$  converges to  $U^*$  and that the mapping is differentiable in a neighborhood of  $U^*$ , then we can use a simple Taylor expansion to derive an expression.

$$U^{(t+1)} - U^* = (U^{(t)} - U^*) \frac{\partial \theta(U)}{\partial U} \Big|_{U=U^*} + O\left(\|U^{(t)} - U^*\|^2\right) \tag{20}$$

where  $\|\cdot\|$  is the usual Euclidean norm. Within a certain neighborhood of  $U^*$ , the behavior of the KLFCM algorithm can be well approximated by a linear iteration using the Jacobian matrix  $\frac{\partial \theta(U)}{\partial U} \Big|_{U=U^*}$ .

Our focus now is on investigating the convergence rate of linear iterations in the KLFCM algorithm. Specifically, we define the global rate of convergence for this algorithm as  $cr = \lim_{t \rightarrow \infty} \frac{\|U^{(t+1)} - U^*\|}{\|U^{(t)} - U^*\|}$ . Furthermore, a higher value of  $cr$  corresponds to a slower rate of convergence. To estimate the convergence rate of KLFCM, we calculate the spectral radius of  $\frac{\partial \theta(U)}{\partial U} \Big|_{U=U^*}$  at the point of convergence,  $(V^*, U^*)$ . This is because the convergence rate of KLFCM is determined by the spectral radius of  $\frac{\partial \theta(U)}{\partial U} \Big|_{U=U^*}$ , as indicated by Equation (20). By evaluating the spectral radius, we can approximate how quickly the algorithm will converge to its solution.

We varied the parameter  $\lambda$  in the KLFCM clustering algorithm, and for each value, we computed the spectral radius of the Jacobian matrix at the convergence point. This was done to explore how different parameter values influence the convergence rate of the algorithm. At the point of convergence, we can use Equation (14) to calculate the spectral radius of the Jacobian matrix for KLFCM and denote it as  $r_{SR}$ . The results are shown in Table 5.

**Table 5.** Spectral radius  $r_{SR}$  of the Jacobian matrix corresponding to different values of  $\lambda$ .

Data	$\lambda_{SR}$ Corresponding to Different $\lambda$						
Iris	$\lambda$	2	2.5	3	3.5	4	4.5
	$r_{SR}$	0.5655	0.8622	0.9221	0.9484	0.9857	1.0000
Cloud	$\lambda$	2	3	4	4.5	5	6
	$r_{SR}$	0.8505	0.8713	0.8945	0.9164	0.9554	0.9701
Wine	$\lambda$	2	2.5	3	3.5	4	4.5
	$r_{SR}$	0.6995	0.7983	0.7198	0.9559	0.7526	0.8223
Haberman’s Survival	$\lambda$	2	3	4	5	6	7
	$r_{SR}$	0.3786	0.6212	0.7940	0.9469	0.9067	0.9339
seeds	$\lambda$	2	3	4	5	6	7
	$r_{SR}$	0.4051	0.8357	0.8255	0.8093	0.8632	0.8777
Sonar	$\lambda$	2	3	4	5	6	7
	$r_{SR}$	0.1066	0.5920	0.1118	0.2586	0.4763	0.8928
Balance Scale	$\lambda$	2	3	4	5	6	7
	$r_{SR}$	0.8963	1.0000	1.0000	1.0000	1.0000	1.0000

We know that the larger the value of  $r_{SR}$ , the slower the convergence rate. Table 5 demonstrates that as the values of  $\lambda$  increase, the KLFCM algorithm exhibits a decreasing trend in convergence rates due to an increasing trend in the spectral radius. In the case of the Iris dataset, setting  $\lambda = 2$  results in  $r_{SR} = 0.5655$ . However, if we increase the value of  $\lambda$  to 3, we get a spectral radius of 0.9221. Clearly, the convergence rate is observed to

decrease in response to higher parameter values. This trend aligns with most experimental findings of the KLFCM algorithm. Specifically, setting a large value for the parameter  $\lambda$  in the KLFCM clustering algorithm can result in a fuzzier output, potentially causing slower convergence. Our demonstration shows that the Jacobian matrix can also be utilized to estimate the convergence rate.

## 5. Discussion

### 5.1. Main Results

Our experimental results indicate the following facts:

(1) The fuzzy parameter  $\lambda$  is a critical factor in the KLFCM algorithm. If the value is excessively large, it could lead to clustering failure.

(2) The KLFCM algorithm will exhibit self-annealing properties. Let  $r^*$  denote the spectral radius of Jacobian matrix  $\frac{\partial \theta}{\partial U} |_{\forall i,k, u_{ik}=c^{-1}}$ . If we choose a parameter that satisfies the condition of  $r^* > 1$ , even with suboptimal initial clustering center selection, the algorithm can still produce satisfactory clustering results through its self-regulating nature.

(3) This study suggests that selecting parameter values leading to  $r^*$  greater than 1 is optimal due to the possibility of some clusters merging or overlapping when  $r^* = 1$ .

(4) The convergence rate of the KLFCM clustering algorithm can be estimated using the spectral radius of the Jacobian matrix at the convergence point.

### 5.2. Discussion of $\lambda$

As with other clustering algorithms, the KLFCM algorithm's outcomes can be influenced by parameter values. When the  $\lambda$  value is too large, clustering results may exhibit partial cluster overlap. In cases where the parameter values are exceedingly large, the algorithm may produce outputs where all samples belong to one cluster due to its inability to differentiate significant overlapping clusters. This issue arises regardless of the initial class center selection of the KLFCM algorithm. Hence, discussing the algorithm's self-annealing behavior becomes irrelevant when the parameter values are excessively large, (see Figures 1 and 2).

### 5.3. Discussion of Self-Annealing Property

Based on the experimental results, it was observed that selecting parameters leading to  $r^*$  greater than 1 allows the KLFCM algorithm to produce satisfactory clustering outcomes, even with suboptimal initial center selection or cases where the cluster centers are very close to each other. For instance, in Example 1, when  $\lambda$  is less than 3.4, all values satisfy the condition of  $r^* > 1$ . As a result, in these scenarios, the data were perfectly partitioned into different clusters. The experimental results suggest that the KLFCM clustering algorithm can exhibit self-annealing properties when certain conditions are met in parameter selection, (see Figures 2 and 3).

### 5.4. Discussion of Parameter Selection

It is evident that when the chosen parameter values meet the condition of  $r^* > 1$ , the clustering algorithm's output results exhibit strong interpretability and acceptability. When the selected parameter leads to  $r^* = 1$ , the initial cluster center selection can impact the algorithm's clustering results. For example, in the case of the Iris dataset, we employed two different cluster center initialization methods under the same parameter conditions (e.g.,  $\lambda = 5$ ). Initializing the clustering centers using the K-means algorithm ensured that they were uniformly distributed across various data regions, resulting in more reliable clustering outcomes. Therefore, when investigating parameter selection issues, we used the K-means algorithm output as initialization for the KLFCM algorithm. From the findings presented in Table 4, we discovered that even in some cases where  $r^* = 1$ , we could still identify partial underlying data features through the clustering results, (see Figure 4, Tables 3 and 4).

It should be emphasized that for datasets from UCI databases in Table 2, the majority of them meet the condition of  $r^* > 1$  when  $\lambda = 2$ . In previous literature, it has been suggested that when  $\lambda = 2$ , the KLFCM algorithm is equivalent to the EM algorithm for Gaussian mixtures. Our prior work has illustrated the self-annealing property of the EM algorithm, which aligns with the research outcomes presented in this paper.

### 5.5. Discussion of Convergence Rate

The degree of fuzziness in KLFCM algorithm clusters can be managed by adjusting the value of lambda. When  $\lambda$  increases, the fuzziness of the clusters also increases, leading to less distinct and more blurred cluster results that are harder to interpret. Furthermore, as lambda increases, the algorithm takes longer to converge due to the greater complexity of the optimization problem. Our experimental findings align with this phenomenon, (see Table 5).

## 6. Conclusions

Since its inception by Ichihashi and Honda, the KLFCM algorithm has become one of the most widely used clustering models. In this article, we perform convergence and parameter analysis of the KLFCM clustering algorithm using the Jacobian matrix. Our findings suggest that the coincident clustering result is not a stable fixed point of KLFCM clustering. Additionally, we discovered an interesting pattern in most datasets where  $\lambda = 2$  results in  $r^* > 1$ . While our research includes theoretical analysis, it also has practical implications for the application of the KLFCM algorithm. Not only did we provide a theoretical basis for the self-annealing behavior of the KLFCM algorithm, but we also proposed an initialization selection strategy for algorithm parameters. The ideal parameter value should avoid  $r^* > 1$ .

The primary advantage of our proposed analysis method is that we can use Jacobian matrix-based analysis to evaluate any clustering algorithm with an iterative update formula. However, this method has its limitations. For example, we only considered consistent clustering results, but when the parameters are not appropriately selected, there may be some clusters merging in the output. Clearly, while the KLFCM algorithm has self-annealing properties, it may not always produce good results, especially when the parameters are not initialized properly. Therefore, future research must focus on developing better parameter selection strategies to ensure that clustering results accurately reflect the underlying structure of the data.

**Funding:** This work was supported by the National Natural Science Foundation of China (Grant No. 62006257).

**Data Availability Statement:** The dataset used for this work is sourced from the UCI database. It can be found at <https://archive.ics.uci.edu/ml/index.php> (accessed on 12 January 2023).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Sathiyasek, K.; Karthick, S.; Kanniya, P.R. The survey on various clustering technique for image segmentation. *Int. J. Adv. Res. Electron. Comm. Eng.* **2014**, *3*, 1.
2. Ghosal, A.; Nandy, A.; Das, A.K.; Goswami, S.; Panday, M. A short review on different clustering techniques and their applications. *Emerg. Technol. Model. Graph.* **2018**, *2020*, 69–83.
3. Ghosh, S.; Dubey, S.K. Comparative analysis of K-means and fuzzy C-means algorithms. *Int. J. Adv. Comput. Sci. Appl.* **2013**, *4*, 35–39. [[CrossRef](#)]
4. Everitt, B.S. *Cluster Analysis*; Halstead Press: London, UK, 1974.
5. Zadeh, L.A. Fuzzy sets. *Inf. Control* **1954**, *8*, 338–356. [[CrossRef](#)]
6. Dunn, J.C. A fuzzy relative of the ISODATA process and its use in detecting compact well separated clusters. *J. Cybern.* **1974**, *3*, 32–57. [[CrossRef](#)]
7. Ichihashi, H.; Miyagishi, K.; Honda, K. Fuzzy c-means clustering with regularization by KL information. In Proceedings of the 10th IEEE International Conference on Fuzzy Systems (Cat. No. 01CH37297), Melbourne, VIC, Australia, 2–5 December 2001; Volume 2, pp. 924–927.

8. Karlekar, A.; Seal, A.; Krejcar, O.; Gonzalo-Martin, C. Fuzzy k-means using non-linear s-distance. *IEEE Access* **2019**, *7*, 55121–55131. [[CrossRef](#)]
9. Zhao, X.; Nie, F.; Wang, R.; Li, X. Improving projected fuzzy K-means clustering via robust learning. *Neurocomputing* **2022**, *491*, 34–43. [[CrossRef](#)]
10. Yang, X.; Zhu, M.; Sun, B.; Wang, Z.; Nie, F. Fuzzy C-Multiple-Means Clustering for Hyperspectral Image. *IEEE Geosci. Remote Sens. Lett.* **2023**, *20*, 5503205. [[CrossRef](#)]
11. Seal, A.; Karlekar, A.; Krejcar, O.; Gonzalo-Martin, C. Fuzzy c-means clustering using Jeffreys-divergence based similarity measure. *Appl. Soft Comput.* **2020**, *88*, 106016. [[CrossRef](#)]
12. Honda, K.; Ichihashi, H. Regularized linear fuzzy clustering and probabilistic PCA mixture models. *IEEE Trans. Fuzzy Syst.* **2005**, *13*, 508–516. [[CrossRef](#)]
13. Gharieb, R.R.; Gendy, G. Fuzzy C-means with a local membership KL distance for medical image segmentation. In Proceedings of the Cairo International Biomedical Engineering Conference (CIBEC), Giza, Egypt, 11–13 December 2014; pp. 47–50.
14. Zhang, H.; Wu, Q.M.J.; Nguyen, T.M. A robust fuzzy algorithm based on student's t-distribution and mean template for image segmentation application. *IEEE Signal Process. Lett.* **2012**, *20*, 117–120. [[CrossRef](#)]
15. Li, X.L.; Chen, J.S. Region-Based Fuzzy Clustering Image Segmentation Algorithm with Kullback-Leibler Distance. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2020**, *4*, 27–31. [[CrossRef](#)]
16. Amira, O.; Zhang, J.S.; Liu, J. Fuzzy c-means clustering with conditional probability based K–L information regularization. *J. Stat. Comput. Simul.* **2021**, *91*, 2699–2716. [[CrossRef](#)]
17. Yun, S.; Zanetti, R. Clustering Methods for Particle Filters with Gaussian Mixture Models. *IEEE Trans. Aerosp. Electron. Syst.* **2021**, *58*, 1109–1118. [[CrossRef](#)]
18. Yu, J.; Chaomurilige, C.; Yang, M.S. On convergence and parameter selection of the EM and DA-EM algorithms for Gaussian mixtures. *Pattern Recognit.* **2018**, *77*, 188–203. [[CrossRef](#)]
19. Chaomurilige; Yu, J.; Yang, M.S. Analysis of parameter selection for Gustafson–Kessel fuzzy clustering using Jacobian matrix. *IEEE Trans. Fuzzy Syst.* **2015**, *23*, 2329–2342. [[CrossRef](#)]
20. Chaomurilige; Yu, J.; Yang, M.S. Deterministic annealing Gustafson-Kessel fuzzy clustering algorithm. *Inf. Sci.* **2017**, *417*, 435–453. [[CrossRef](#)]
21. Chaomurilige, C.; Yu, J.; Zhu, J. Analysis of Convergence Properties for Gath-Geva Clustering Using Jacobian Matrix. In Proceedings of the Chinese Conference on Pattern Recognition, Chengdu, China, 5–7 November 2016.
22. Sadaaki, M.; Masao, M. Fuzzy c-means as a regularization and maximum entropy approach. In Proceedings of the 7th International Fuzzy Systems Association World Congress (IFSA'97), Prague, Czech Republic, 25–29 June 1997; Volume 2, pp. 86–92.
23. Olver, P.J. Lecture Notes on Numerical Analysis. Available online: <http://www.math.umn.edu/~olver/num.html> (accessed on 1 March 2023).
24. Roubens, M. Pattern classification problems and fuzzy sets. *Fuzzy Sets Syst.* **1978**, *1*, 239–253. [[CrossRef](#)]

**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.