entropy

MDPI

Article

An Entropy Regularization *k*-Means Algorithm with a New Measure of between-Cluster Distance in Subspace Clustering

Liyan Xiong, Cheng Wang *, Xiaohui Huang and Hui Zeng

School of Information Engineering Department, East China Jiaotong University, R.d 808, East Shuanggang Avenue, Nanchang 330013, China

* Correspondence: wangcheng.me@gmail.com

Received: 15 April 2019; Accepted: 10 July 2019; Published: 12 July 2019

Abstract: Although within-cluster information is commonly used in most clustering approaches, other important information such as between-cluster information is rarely considered in some cases. Hence, in this study, we propose a new novel measure of between-cluster distance in subspace, which is to maximize the distance between the center of a cluster and the points that do not belong to this cluster. Based on this idea, we firstly design an optimization objective function integrating the between-cluster distance and entropy regularization in this paper. Then, updating rules are given by theoretical analysis. In the following, the properties of our proposed algorithm are investigated, and the performance is evaluated experimentally using two synthetic and seven real-life datasets. Finally, the experimental studies demonstrate that the results of the proposed algorithm (ERKM) outperform most existing state-of-the-art *k*-means-type clustering algorithms in most cases.

Keywords: *k*-means; between-cluster information; entropy regularization; data mining

1. Introduction

Clustering is a process of dividing a set of points into multiple clusters. In this process, the similarity among points in a cluster is higher than that among points from different clusters. Due to high efficiency, *k*-means-type clustering algorithms have been widely used in various fields of real life [1,2], such as marketing [3] and bioinformatics [4].

In the past few decades, the research of clustering techniques has been extended to many fields [5] such as gene analysis [6] and community detection [7]. The idea of most clustering algorithms aims at making the similarity among points in a cluster higher than those from different clusters, namely to minimize the within-cluster distance [8]. However, the traditional clustering methods seem to be weak when dealing with high-dimensional data under many cases [9]. For example, for points in the same cluster, the distance between each dimension should be small, but the true results after clustering by traditional clustering algorithms are that the distance between some dimensions may be very large.

The classic *k*-means-type algorithms cannot automatically determine the importance of a dimension, i.e., which are important dimensions and which are noise dimensions, because they treat all dimensions equally in the clustering process [10]. At the same time, it is also worth noting that the valid dimensions are often a part of the dimensions rather than all in the process of high-dimensional data clustering. Fortunately, in recent years, some subspace clustering algorithms [11–16] have largely alleviated this problem. For example, the work in [16] considered that different dimensions make different contributions, by introducing entropy weighting to the identification of objects in clusters.

To date, many subspace clustering algorithms have been proposed and well applied in various fields; however, most of them overlooked the within-cluster distance in the clustering process. Namely, the between-cluster information is not fully utilized [17] for improving the clustering performance. The between-cluster information is utilized in some algorithms [5,18,19]. However, in some cases, e.g., clusters with a blurred decision boundary, the existing methods cannot conquer this effectively.

In traditional ways, many algorithms utilize the between-cluster information by introducing the global center [5,18]. The main idea of these methods is to maximize the distance between the center of each cluster and the global center. Under these circumstances, if the number of points in each cluster differs greatly, then the global center will be heavily biased toward the cluster with a large number of points (e.g., in Figure 1a, the global center z_0 is heavily biased toward Cluster 1). Subsequently, in order to maximize the distance between z_0 and z_1, the latent center of Cluster 1 will greatly deviate from its true cluster center z_1, resulting in the decrease of the performance in convergence.

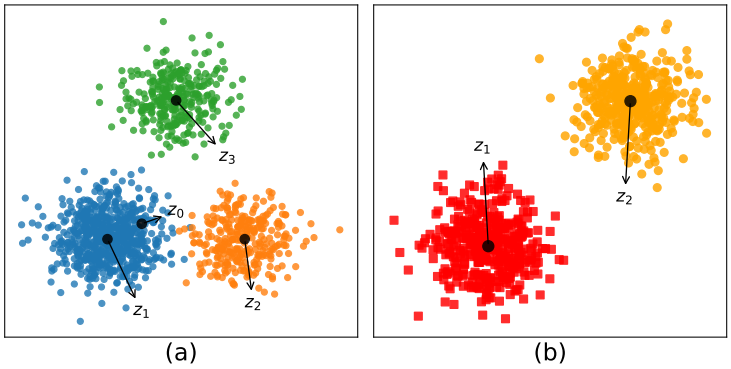

Figure 1. (**a**) The idea with the global center; (**b**) the idea without the global center.

Motivated by this, we propose a new measure of between-cluster distance based on entropy regularization in this paper, which is totally different from these existing methods (see, for instance, [5,18–22]). The key idea is to maximize the distance between the points in the subspace that do not belong to the cluster and the center point of the cluster. As shown in Figure 1b, ERKM maximizes the distance between $z_1(z_2)$ and circle-points (square-points) in subspace. In order to award more dimensions to make contributions to the identification of each cluster, avoiding few sparse dimensions, we also introduce the entropy regularization term in the objective function.

Moreover, the measure of between-cluster distance proposed in this paper can also be applied to other clustering algorithms, not only limited to *k*-means-type algorithms. The main contributions of this paper are as follows:

- The study on subspace clustering mentioned in previous papers is summarized.
- A new *k*-means clustering algorithm combining between-cluster distance and entropy regularization [16,23] is introduced.
- By optimization, the update rules of ERKM algorithm are obtained, and the convergence and robustness analysis of the algorithm are given.
- The hyperparameters are studied on synthetic and real-life datasets. Finally, through experimental comparison, the results show that the ERKM algorithm outperforms most existing state-of-the-art *k*-means-type clustering algorithms.

The rest of this paper is organized as follows: Section 2 is an overview of related works about subspace clustering algorithms. Section 3 presents a new *k*-means-type subspace clustering algorithm, and update rules are given. Experiments and results are analyzed in Section 4. In Section 5, conclusions and future work are presented.

2. Related Work

The research on subspace clustering [24] has always been an important direction in clustering algorithms. In order to achieve high performance, many methods, such as sparse clustering-based methods [25–27], weight entropy-based methods [16,28], between-cluster information-based methods [29–31], and so on, have been proposed. Furthermore, different ways in different cases will have a different impact on the results of clustering algorithms. In this section, we will give a brief summary of the ways in different fields proposed in recent years.

For a clear statement of related work, we first introduce the most common notations. Let $X = (x_{ij}) \in \mathcal{R}^{n\times m}$ denote a dataset in the matrix form with n points and m dimensions. $U = (u_{li}) \in \mathcal{R}^{k\times n}$ denotes a partition matrix, where $u_{li} = 1$ indicates that point i is assigned to cluster l; otherwise, it is not assigned to cluster l. $Z = (z_{lj}) \in \mathcal{R}^{k\times m}$ is a set of k vectors representing the centers of k clusters. W is a weighting matrix or vector depending on specific algorithms.

2.1. Sparseness-Based Methods

In order to cluster high-dimensional objects in subspace, Witten et al. [26] proposed a novel framework for sparse clustering, which clusters the points using an adaptively-chosen subset of the dimensions. The main idea it to use a lasso-type penalty to select the features. This framework can be used for sparse k-means clustering and also sparse hierarchical clustering. Taking sparse k-means as an example, its objective function can be reformulated as follows:

$$P(U, Z, W) = \underset{u_1,\ldots,u_k,W}{\text{maximize}} \left\{ \sum_{j=1}^{m} w_j \left(\frac{1}{n} \sum_{i=1}^{n} \sum_{i'=1}^{n} (x_{ij} - x_{i'j})^2 - \sum_{p=1}^{k} \frac{1}{n_p} \sum_{i,i' \in z_p} (x_{ij} - x_{i'j})^2 \right) \right\}, \quad (1)$$

subject to:

$$\|W\|^2 \leq 1, \quad \|W\|_1 \leq s, \quad w_j \geq 0 \ \forall j.$$

The weights in this algorithm will be sparse for an appropriate choice of the tuning parameter s, which should satisfy $1 \leq s \leq \sqrt{m}$. If $w_1 =, \cdots, = w_p$, the objective function (1) simply reduces to the criterion standard k-means clustering.

Combining the penalized likelihood approach with an L_1 penalty function and model-based clustering [32], Pan et al. [27] presented a penalized model-based clustering algorithm, automatically selecting features and delivering a sparse solution. This can be used for high-dimensional data. Its objective function can be expressed as follows:

$$\log L_{c,P}(W) = \sum_{p=1}^{k} \sum_{i=1}^{n} z_{pi} \left[\log \pi_p + \log f_p \left(x_i; \theta_p \right) \right] - h_\lambda(W), \quad (2)$$

where $h_\lambda(\cdot)$ is a penalty function with penalization parameter λ. The choice of $h_\lambda(\cdot)$ depends on the goal of the analysis.

2.2. Entropy-Based Methods

For the sake of stimulating dimensions in subspace, Jing et al. [16] extended the k-means algorithm to calculate a weight for each dimension in each cluster, achieved by including the weight entropy in the objective function, and used it to identify the subsets of important dimensions. The weights of all dimensions can be automatically computed only by adding an additional step to the k-means clustering process. The objective function of this algorithm (EWKM) can be written as follows:

$$P(U, Z, W) = \sum_{p=1}^{k} \left[\sum_{i=1}^{n} \sum_{j=1}^{m} u_{pi} w_{pj} (z_{pj} - x_{ij})^2 + \gamma \sum_{j=1}^{m} w_{pj} log w_{pj} \right], \quad (3)$$

subject to:

$$\begin{cases} \sum_{p=1}^{k} u_{pi} = 1, \ \ 1 \le i \le n, \ \ 1 \le p \le k, \ \ u_{pi} \in \{0,1\}, \\ \sum_{j=1}^{m} w_{pj} = 1, \ \ 1 \le p \le k, \ \ 1 \le j \le m, \ \ 0 \le w_{pj} \le 1, \end{cases}$$

where the weight w_{lj} is to measure the contribution (or importance) of the j-th dimension in cluster l. The more w_{lj}, the higher contribution of dimension j in partitioning cluster l. The positive parameter γ controls the strength of the incentive for clustering on more dimensions. The first term in (3) is the sum of the within-cluster distance and the second term the negative weight entropy, which is to stimulate more dimensions to contribute to the identification of clusters, avoiding the problem of identifying clusters by a few dimensions.

To achieve the same goal, Zhou et al. [28] proposed a novel fuzzy c-means algorithm, by using a weighted dissimilarity measure and adding a weight entropy regularization term to the objective function. It only adds a fuzzy index to the algorithm EWKM, and its objective function can be re-written as follows:

$$P(U,Z,W) = \sum_{p=1}^{k} \left[\sum_{i=1}^{n} \sum_{j=1}^{m} u_{pi}^{\alpha} w_{pj} (z_{pj} - x_{ij})^2 + \gamma \sum_{j=1}^{m} w_{pj} log w_{pj} \right], \tag{4}$$

where the exponent $\alpha \ge 1$ is to control the extent of membership sharing between the fuzzy clusters. When $\alpha = 1$, then the objective function (4) simply reduces to the EWKM algorithm.

2.3. Between-Cluster Measure-Based Methods

2.3.1. Liang Bai's Methods

The k-modes algorithms and its modified versions for categorical data have always been a hot topic in clustering algorithms. Like the clustering algorithm for real number data, the use of between-cluster information is also particularly vital for the categorical data. A good between-cluster measure approach can often achieve better results.

Based on the fuzzy k-modes algorithm [1], Bai et al. [21] presented a new algorithm by adding the between-cluster information term so that one can simultaneously minimize the within-cluster dispersion and enhance the between-cluster separation. The objective function can be written as follows:

$$F_n(U,Z,\gamma) = \sum_{p=1}^{k} \sum_{i=1}^{n} u_{pi}^{\alpha} \sum_{j=1}^{m} \delta(z_{pj}, x_{ij}) + \gamma \sum_{p=1}^{k} \sum_{i=1}^{n} u_{pi}^{\alpha} \frac{1}{n} \sum_{i=1}^{n} s(z_p, x_i), \tag{5}$$

where the second term in (5) is a definition of the between-cluster information, which is a similarity measure between cluster z_l and point x_i and defined as:

$$s(z_p, x_q) = \sum_{j=1}^{m} \phi(z_{pj}, x_{qj}), \ \ \phi(z_{pj}, x_{qj}) = \begin{cases} 1, & z_{pj} = x_{qj} \\ 0, & z_{pj} \ne x_{qj} \end{cases},$$

where the parameter γ is used to maintain a balance between the effect of the within-cluster information and the between-cluster information on the minimization process of the objective function (5).

Similar to (5), Bai et al. [22] defined another between-cluster similarity term to evaluate the between-cluster separation. The between-cluster similarity term is defined as:

$$B_g(U,Z) = \sum_{p=1}^{k} \sum_{i=1}^{n} u_{pi} \frac{1}{n} \sum_{p=1}^{n} s_g(z_p, x_q), \tag{6}$$

where $s_g(z_p, x_i)$ is a similarity measure between z_p and x_i. At the same time, this term with different forms can be added in different k-means-types algorithm in different cases, such as in (7).

$$F_0(W, Z, \gamma) = \sum_{p=1}^{k} \sum_{i=1}^{n} u_{pi} d_0(z_p, x_i) + \gamma B_0(U, Z), \tag{7}$$

where $\gamma \geq 0$ is to maintain a balance between the effect of the within-cluster information and the between-cluster information on the minimization process. B_0 is defined as:

$$B_0(U, Z) = \sum_{p=1}^{k} \sum_{i=1}^{n} u_{pi} \frac{1}{n} \sum_{p=1}^{n} (m - d_0(z_p, x_q)),$$

where $d_0(z_p, x_q)$ is the distance between cluster z_p and point x_q.

2.3.2. Huang's Methods

k-means-type clustering aims at partitioning a dataset into clusters such that the objects in a cluster are compact and in different clusters are well separated. By integrating within-cluster compactness and between-cluster separation, Huang et al. [18] also designed a new method to utilize the between-cluster information. Based on this basic idea, many traditional algorithms without considering between-cluster separation can be modified, such as basic *k*-means and wk-means [10] algorithms.

An example of a modified objective function of *k*-means algorithm can be re-written as follows:

$$P(U, Z) = \sum_{p=1}^{k} \sum_{i=1}^{n} u_{pi} \sum_{j=1}^{m} \frac{(x_{ij} - z_{pj})^2}{(z_{pj} - z_{0j})^2}, \tag{8}$$

where z_{0j} is the j-th feature of the global center z_0 of a dataset. Its main idea is to minimize the distances between objects and the center of the cluster that the objects belong to, while maximizing the distances between centers of clusters and the global center. From (8), we can find that it considers the distance between cluster center z_p and global center z_0, indicating that the greater the distance, the higher the probability of cluster center z_p being confirmed.

Based on a similar motivation, Huang et al. [19] proposed another new discriminative subspace *k*-means-type clustering algorithm, which integrates the within-cluster compactness and the between-cluster separation simultaneously. Its main idea is to use a three-order tensor weighting method to discriminate the weights of features when comparing every pair of clusters. The objective function can be represented as:

$$\begin{aligned} P(U, W, Z) &= \sum_{p=1}^{k} \sum_{\substack{q=1 \\ q \neq p}}^{k} \sum_{j=1}^{m} w_{pqj} D_{pqj} + \gamma \sum_{p=1}^{k} \sum_{\substack{q=1 \\ q \neq p}}^{k} \sum_{j=1}^{m} w_{pqj} \log(w_{pqj}), \\ D_{p,q,j} &= \textstyle\sum_{i=1}^{n} u_{ip} [(x_{ij} - z_{pj})^2 - \eta (z_{pj} - z_{qj})^2], \end{aligned} \tag{9}$$

where the weight w is a three-order tensor and each value $w_{p,q,j}$ in denotes the importance of the feature j in cluster p when comparing cluster p to cluster q, where $p \neq q$. γ is a parameter that controls the distribution of the weight, and parameter η is used for balancing the effect of within-cluster compactness and between-cluster separation. In the objective function (9), the first term includes two parts: one is the sum of within-cluster compactness; the other is the sum of the distances between centers of different clusters, which involves maximizing the inter-cluster separation.

2.4. Others

While between-cluster information and weight entropy are commonly used in most subspace clustering algorithms, their performance can be further enhanced. A major weakness of entropy-based clustering algorithms is that they do not consider the between-cluster information. Motivated by this, Deng et al. [5] proposed an Enhanced Soft Subspace Clustering (ESSC) algorithm, combining the

EWKM and a new measure between-cluster distance, which is to maximize the distance between cluster centers and global center in subspace.

An intuitive explanation of the between-cluster separation is given below. As illustrated in Figure 2, the main idea of ESSC is to both minimize the with-cluster distance and maximize between-cluster distance (e.g., $||z_0z_1||, ||z_0z_2||, ||z_0z_3||$) simultaneously in subspace, which can make the three clusters' centers v_1, v_2, v_3 as far apart as possible from each other.

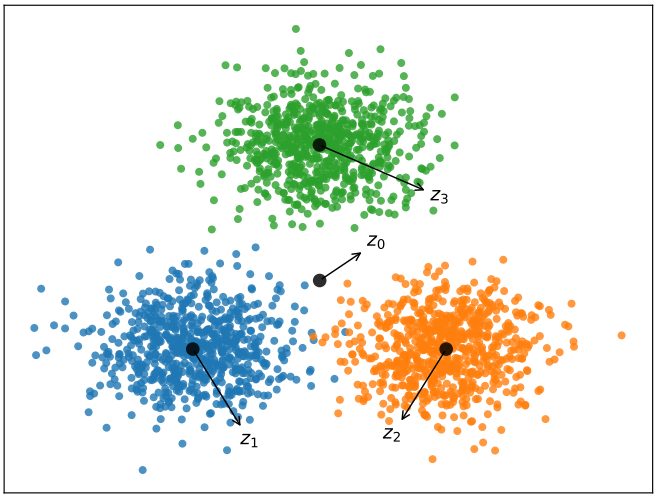

Figure 2. Main idea of ESSC (z_1, z_2, z_3 are the cluster centers, and z_0 is the global center). ESSC: Enhanced Soft Subspace Clustering.

The objective function of ESSC can be expressed as:

$$\begin{aligned} P(U,Z,W) = &\sum_{p=1}^{k}\sum_{i=1}^{n} u_{pi}^{\beta} \sum_{j=1}^{m} w_{pj}(x_{ij}-z_{pj})^2 + \\ &\gamma \sum_{p=1}^{k}\sum_{j=1}^{m} w_{pj}\log w_{pj} - \alpha \sum_{p=1}^{k}\left(\sum_{i=1}^{n} u_{pi}^{\beta}\right)\sum_{j=1}^{m} w_{pj}(z_{pj}-z_{0j})^2 \end{aligned} \tag{10}$$

where β and $W = [W_1, \cdots, W_k]$ are the fuzzy index and weighting matrix, respectively, and $W_q = [w_{q1}, \cdots, w_{qm}]$, w_{qj} is the weight of the j-th feature in the p-th cluster. $\gamma > 0, \alpha > 0$ are two parameters. z_0 is the global center of all points. The total weighted distance in the subspace between the global center and each cluster center is calculated by the third term in (10), which is used to maximize the between-cluster distance as much as possible in clustering.

Enlightened by the regularization, Chang et al. [33] proposed a novel Fuzzy c-Means (FCM) model with sparse regularization, by reformulating the FCM objective function into the weighted between-cluster sum of squares form and imposing the spare regularization on the weights. Its objective function can be rewritten as follows:

$$P(U,Z,W) = \text{maximize}\left\{\sum_{j=1}^{m} w_j\left(\frac{1}{n}\sum_{i=1}^{n}\sum_{i'=1}^{n}(x_{ij}-x_{i'j})^2 - \sum_{p=1}^{k}\frac{1}{n_p}\sum_{i=1}^{n}\sum_{i'=1}^{n} u_{ip}^{\alpha}u_{i'p}^{\alpha}(x_{ij}-x_{i'j})^2\right)\right\}, \tag{11}$$

subject to:

$$\begin{cases} \sum_{p=1}^{k} u_{ip} = 1, 0 \le u_{ip} \le 1, \\ \|W\|_2 \le 1, \|W\|_q^q \le s, \\ \\ w_j \ge 0, j = 1, \ldots, m, \end{cases}$$

where $0 < q \leq 1$, $\|W\|_q^q = \sum_{j=1}^{m} |w_j|^q$, and $\|W\|_q^q$ is the sparse regularization constraint conditions, which is to make the weight of some dimensions near zero so that the relevant features can be found.

3. Entropy Regularization Clustering

The ERKM algorithm proposed in this paper mainly extends the EWKM algorithm [16]. On this basis, a new method of measuring between-cluster distance is introduced. The idea of the new method is to maximize the between-cluster distance by maximizing the distance between the center of a cluster and the points that do not belong to the cluster in subspace. The new algorithm uses vector-weighting to find the best subspace and adjusts the weight of each dimension through the entropy regularization term. Based on this idea, we firstly develop an objective function for the algorithm. Then, the update rules of each variable are obtained by minimizing the objective function, and the convergence is proven.

3.1. ERKM Algorithm

Let $W = \{w_1, w_2, \cdots, w_m\}$ be the weights of features that represent the contribution of each dimension in the clustering process.

The new objective function is written as follows:

$$\begin{aligned} P(W, U, Z) &= \sum_{p=1}^{k} \sum_{i=1}^{n} u_{pi} \sum_{j=1}^{m} w_j (x_{ij} - z_{pj})^2 \\ &+ \gamma \sum_{j=1}^{m} w_j \log w_j - \eta \sum_{p=1}^{k} \sum_{i=1}^{n} (1 - u_{pi}) \sum_{j=1}^{m} w_j (x_{ij} - z_{pj})^2, \end{aligned} \tag{12}$$

subject to:

$$\begin{cases} \sum_{j=1}^{m} w_j = 1, \ \ 0 < w_j < 1, \\ \sum_{p=1}^{k} u_{pi} = 1, \ \ u_{pi} \in \{0, 1\}. \end{cases}$$

There are three terms in the objective function: the weighted within-cluster distance term, the entropy regularization term, and the weighted between-cluster distance term. The first and second terms are directly extended from EWKM. The first term is to make the within-cluster distance as small as possible in subspace, and the second term is to allow more dimensions to participate in the clustering process. The parameter $\gamma > 0$ controls the distribution of w in each dimension. The last term is a new between-cluster distance measure method proposed in this paper, which is to maximize the between-cluster distance. $\eta > 0$ is a hyperparameter used to control the influence of between-cluster distance on the objective function, degenerating into the EWKM algorithm when $\eta = 0$.

In order to optimize this function conveniently, it can modify Equation (12) as follows:

$$\begin{aligned} P(W, U, Z) &= (1 + \eta) \sum_{p=1}^{k} \sum_{i=1}^{n} u_{pi} \sum_{j=1}^{m} w_j (x_{ij} - z_{pj})^2 \\ &+ \gamma \sum_{j=1}^{m} w_j \log w_j - \eta \sum_{p=1}^{k} \sum_{i=1}^{n} \sum_{j=1}^{m} w_j (x_{ij} - z_{pj})^2, \end{aligned} \tag{13}$$

subject to

$$\begin{cases} \sum_{j=1}^{m} w_j = 1, \ \ 0 < w_j < 1, \\ \sum_{p=1}^{k} u_{pi} = 1, \ \ u_{pi} \in \{0, 1\}. \end{cases}$$

It is established based on the three basic theorems below.

Theorem 1. *Given U and Z are fixed, P is minimized only if:*

$$w_j = \exp\left(\frac{-D_j}{\gamma}\right) / \sum_{t=1}^{m} \exp\left(\frac{-D_t}{\gamma}\right), \tag{14}$$

where:

$$D_j = (1+\eta) \sum_{p=1}^{k} \sum_{i=1}^{n} u_{pi}(x_{ij} - z_{pj})^2 - \eta \sum_{p=1}^{k} \sum_{i=1}^{n} (x_{ij} - z_{pj})^2. \tag{15}$$

Proof. We use the Lagrangian multiplier technique to obtain the following unconstrained minimization problem:

$$\begin{aligned} \min P(\{w_j\}, \{\alpha\}) = (1+\eta) \sum_{p=1}^{k} \sum_{i=1}^{n} u_{pi} \sum_{j=1}^{m} w_j (x_{ij} - z_{pj})^2 \\ -\eta \sum_{p=1}^{k} \sum_{i=1}^{n} \sum_{j=1}^{m} w_j (x_{ij} - z_{pj})^2 \\ +\gamma \sum_{j=1}^{m} w_j \log w_j + \alpha \left(\sum_{j=1}^{m} w_j - 1 \right) \end{aligned}. \tag{16}$$

where α is the Lagrange multiplier. By setting the gradient of the function Equation (16) with respect to w_j and α to zero, we obtain the equations:

$$\begin{aligned} &\tfrac{\partial P}{\partial w_j} = D_j + \gamma(\log w_j + 1) + \alpha = 0, \\ &D_j = (1+\eta) \sum_{p=1}^{k} \sum_{i=1}^{n} u_{pi}(x_{ij} - z_{pj})^2 - \eta \sum_{p=1}^{k} \sum_{i=1}^{n} (x_{ij} - z_{pj})^2, \end{aligned} \tag{17}$$

where D_j includes the information of the within-cluster distance and the between-cluster distance of all the points on the dimension.

$$\frac{\partial P}{\partial \alpha} = \sum_{j=1}^{m} w_j - 1 = 0 \tag{18}$$

From (17), we obtain:

$$w_j = \exp\left(\frac{-D_j}{\gamma}\right) \exp\left(\frac{-\alpha - \gamma}{\gamma}\right). \tag{19}$$

Substituting (19) into (18), we have:

$$\sum_{j=1}^{m} w_j = \exp\left(\frac{-\alpha - \gamma}{\gamma}\right) \sum_{j=1}^{m} \exp\left(\frac{-D_j}{\gamma}\right) = 1. \tag{20}$$

From (20), we obtain:

$$\exp\left(\frac{-\alpha - \gamma}{\gamma}\right) = 1 / \sum_{j=1}^{m} \exp\left(\frac{-D_j}{\gamma}\right). \tag{21}$$

Substituting (21) into (19), we have:

$$w_j = \exp\left(\frac{-D_j}{\gamma}\right) / \sum_{t=1}^{m} \exp\left(\frac{-D_t}{\gamma}\right). \tag{22}$$

□

Theorem 2. *Given U and W are fixed, P is minimized only if:*

$$z_{pt} = \frac{(1+\eta)\sum_{i=1}^{n} u_{pi}x_{it} - \eta\sum_{i=1}^{n} x_{it}}{(1+\eta)\sum_{i=1}^{n} u_{pi} - \eta n} \tag{23}$$

Proof. When we have fixed U and W, from (13), we have:

$$Q(W,U,Z) = (1+\eta)\sum_{p=1}^{k}\sum_{i=1}^{n} u_{pi}\sum_{j=1}^{m} w_j(x_{ij}-z_{pj})^2 - \eta\sum_{p=1}^{k}\sum_{i=1}^{n}\sum_{j=1}^{m} w_j(x_{ij}-z_{pj})^2 \ . \tag{24}$$

By setting the gradient of the function (24) with respect to z_{pj} to zero, we obtain the equations:

$$\frac{\partial Q(W,U,Z)}{\partial z_{pj}} = 2(1+\eta)\sum_{i=1}^{n} u_{pi}w_j(z_{pj}-x_{ij}) - 2\eta\sum_{i=1}^{n} w_j(z_{pj}-x_{ij}) = 0. \tag{25}$$

From (25), we have:

$$(1+\eta)\sum_{i=1}^{n} u_{pi}(z_{pj}-x_{ij}) = \eta\sum_{i=1}^{n}(z_{pj}-x_{ij}). \tag{26}$$

From (26), we derive:

$$\sum_{i=1}^{n} u_{ip}z_{pj} + \eta\sum_{i=1}^{n} u_{pi}z_{pj} - \eta\sum_{i=1}^{n} z_{pj} = \sum_{i=1}^{n} u_{pi}x_{ij} + \eta\sum_{i=1}^{n} u_{pi}x_{ij} - \eta\sum_{i=1}^{n} x_{ij}. \tag{27}$$

From (27), we obtain:

$$z_{pj}\left[(1+\eta)\sum_{i=1}^{n} u_{pi} - \eta n\right] = (1+\eta)\sum_{i=1}^{n} u_{pi}x_{ij} - \eta\sum_{i=1}^{n} x_{ij}. \tag{28}$$

then, from (28) we have (23). □

Theorem 3. *Similarly to the k-means algorithm, given Z and W are fixed, u is updated as:*

$$u_{pi} = \begin{cases} 1, \ if \ \sum_{j=1}^{m} w_j(x_{ij}-z_{pj})^2 \geq \sum_{j=1}^{m} w_j(x_{ij}-z_{p'j})^2 \\ 0, \ otherwise \end{cases} . \tag{29}$$

The detailed proof process about Theorem 3 can be found in [34,35].

The ERKM algorithm that minimizes Equation (12), using (14), (23), and (29), is summarized as follows (Algorithm 1):

Algorithm 1 ERKM.

Input: The number of clusters k and parameters γ, η;
Randomly choose k cluster centers, and set all initial weights with a normalized uniform distribution;

repeat
Fixed Z, W, update the partition matrix U by (29)
Fixed U, W, update the cluster centers Z by (23)
Fixed U, Z, update the dimension weights W by (14)
until Convergence
return U, Z, W

The hyperparameter η is used to balance the within-cluster distance and the between-cluster distance. It has the following features in the control of the clustering process:

- When $0 < \eta < \frac{\sum_{p=1}^{k}\sum_{i=1}^{n} u_{pi}\left(x_{ij}-z_{pj}\right)^2}{\sum_{p=1}^{k}\sum_{i=1}^{n}\left(1-u_{pi}\right)\left(x_{ij}-z_{pj}\right)^2}$, according to (14) and (15), w_j is inversely proportional to D_j. The more important the j-th dimension, the larger w_j, and the smaller D_j.
- For others, according to (14) and (15), w_j is proportional to D_j. The larger D_j, the larger w_j. This violates the basic idea that the more important the corresponding dimension, the smaller the sum of the distance on this dimension. Under this circumstance, it will cause the value of $D_j \leq 0$, so that the objective function diverges.

Proof. According to (14), if we want to satisfy this basic idea, namely the smaller D_j, the larger w_j, D_j should be great than zero. Hence, from (15), we have:

$$D_j = (1+\eta)\sum_{n=1}^{k}\sum_{i=1}^{n} u_{pi}\left(x_{ij}-z_{pj}\right)^2 - \eta\sum_{n=1}^{k}\sum_{i=1}^{n}\left(x_{ij}-z_{pj}\right)^2 > 0. \tag{30}$$

From (30), we obtain:

$$\sum_{p=1}^{k}\sum_{i=1}^{n} u_{pi}(x_{ij}-z_{pj})^2 - \eta\sum_{p=1}^{k}\sum_{i=1}^{n}(1-u_{pi})(x_{ij}-z_{pj})^2 > 0. \tag{31}$$

Moreover, according to (31), we have:

$$\sum_{p=1}^{k}\sum_{i=1}^{n} u_{pi}(x_{ij}-z_{pj})^2 > \eta\sum_{p=1}^{k}\sum_{i=1}^{n}(1-u_{pi})(x_{ij}-z_{pj})^2. \tag{32}$$

Finally, from (32), we derive:

$$\eta < \frac{\sum_{p=1}^{k}\sum_{i=1}^{n} u_{pi}(x_{ij}-z_{pj})^2}{\sum_{p=1}^{k}\sum_{i=1}^{n}(1-u_{pi})(x_{ij}-z_{pj})^2}. \tag{33}$$

□

3.2. Convergency and Complexity Analysis

For the ERKM algorithm, when the parameter η satisfies this condition (33), global or local optimal values will be obtained after a finite number of iterations. Obviously, there are only a finite number of possible partitions U, because the number of points is not infinite, and each of the possible partitions will appear only once in the clustering process. Similar to [16], assume that we have $U^{t_1} = U^{t_2}$, where $t_1 \neq t_2$ and t_i represents the number of iterations. Then, based on U^{t_i}, we can obtain Z^{t_i} by minimizing $Q(W, U^{t_i}, Z)$ according to (24). Subsequently, Z^{t_1} and Z^{t_2} are obtained respectively, and furthermore, $Z^{t_1} = Z^{t_2}$ because $U^{t_1} = U^{t_2}$. Finally, according to (14), we can compute the minimizer W^{t_1} and W^{t_2} by using U^{t_1} and Z^{t_1}, and U^{t_2} and Z^{t_2} respectively. Naturally, $W^{t_1} = W^{t_2}$ again. Therefore, we obtain $P(U^{t_1}, Z^{t_1}, W^{t_1}) = P(U^{t_2}, Z^{t_2}, W^{t_2})$. However, the sequence $P(*,*,*)$ is strictly decreasing, which is broken by the analysis result, that is to say, the ERKM algorithm converges in a finite number of iterations.

Similar to the basic k-means algorithm, the proposed one is also iterative. The computational complexity of the basic k-means is $O(tmnk)$, where t is the iterative times; m, n, and k are the number of dimensions, points, and clusters, respectively. As shown in Section 3.1, ERKM has three computational steps including updating the weights, updating the cluster centers, and updating the partition matrix [16]. The complexity of updating the weights is $O(kmn)$. The complexities of updating the cluster centers and partition matrix are $O(kmn + mn + 1)$ and $O(kmn)$, respectively. Hence, the overall computational complexity of ERKM is also $O(tkmn)$. Compared with the basic k-means algorithm, its only needs extra $O(kmn)$ computational time to calculate the weights and $O(mn + 1)$ to calculate the distance of each point i in dimension j. Fortunately, it does not change the total computational complexity of ERKM.

4. Experiments and Discussion

4.1. Experimental Setup

In the experiments, the performance of the proposed algorithm was extensively evaluated on two synthetic and seven real-life datasets tabulated in Table 1, which can be downloaded at the UCI website. We compared the clustering results produced by ERKM with the benchmark clustering algorithms including basic *k*-means (KMEA), WKME, EWKM, ESSC, and the last three years' clustering algorithms, AFKM [36] Sampling-Clustering (SC) [37], and SSC-MP [38].

Table 1. Information of the datasets.

Datasets	Points	Dimensions	Clusters
Synthetic1	500	4	3
Synthetic2	250	1000	3
Knowledge	403	5	4
Iris	150	4	3
Chess	3196	36	2
Tictactoe	958	9	2
Messidor	1151	19	2
Car	1728	6	4
Wine	178	13	3

As we all know, most of *k*-means-type clustering algorithms produce local optimal solution, and the final results depend on the initial cluster centers. For the weighting *k*-means algorithms: WKME and EWKM (also including ours), the initial weights also affect the final results. For the sake of fairly comparing the clustering results, all the cluster centers and weights of each algorithm were randomly initialized. Finally, we compared the average value and standard deviation of each metric produced by the algorithms after 100 runs. in order to speed up the convergence time of the algorithms, all datasets were normalized.

4.2. Evaluation Method

In this paper, four metrics, Accuracy (Acc), Adjusted Rand Index (ARI), F-score (Fsc), and Normal Mutual Information (NMI), were used for evaluating the proposed ERKM algorithm. Acc was used to measure the accuracy of clustering results, and its value range was [0, 1]. Fsc is a weighted harmonic average of precision and recall, and its value range was also [0, 1]. Both ARI and NMI were used to measure the degree of agreement between the two data distributions, which ranged from $[-1, 1]$ and [0, 1], respectively. For the above four evaluation metrics, the larger the value, the better the clustering results. More detail can be found in [19].

4.3. Parameter Setting

Form the objective function (12), the proposed algorithm had only two hyperparameters γ and η. Hence, we will first choose proper parameters for γ and η produced by the results on the datasets.

- Parameter γ:

  Hyperparameter γ appears in the EWKM, ESSC, and ERKM algorithms. ESSC and ERKM algorithms were directly extended from the EWKM algorithm, and only a between-cluster distance constraint was added to the objective function of the EWKM algorithm. At the same time, since the EWKM only contained one parameter γ, the value of γ can be studied by the performance of EWKM on two synthetic and seven real-life datasets, respectively. We fixed the range of γ in [1, 50] and set the step to 1.

- Parameter η:

  In the algorithm ERKM, η has a similar effect as α in the ESSC algorithm. However, since the two algorithms adopted different feature weighting methods and between-cluster distance measures, a reasonable η value will be selected by the results of the ERKM algorithm on two synthetic and seven real-life datasets, respectively. Since the within-cluster distance and the between-cluster distance have different contributions to the whole clustering process, the value of η should satisfy the condition (33), so as to avoid the divergence of the objective function. In the section below, we fixed the range of η in [0.002, 0.2] and set the step to 0.002 in [0, 0.05] and 0.02 in (0.05, 0.2] to search for a proper value.

In summary, we will first choose parameter γ by the results of EWKM and parameter η by the results of ERKM both on two synthetic and seven real-life datasets. For the rest of the comparative algorithms, the default configuration was done (as tabulated in Table 2).

Table 2. Default parameters of comparative algorithms. $*$ We will study this parameter of the corresponding algorithms.

Algorithms	Parameters
KMEA	-
WKME [10]	$\beta = 7$
EWKM [16]	$\gamma = *$
ESSC [5]	$\gamma = *, \alpha = 0.01$
AFKM [36]	-
SC [37]	r = 0.2, size = 16
SSC-MP [38]	$s_{max} = 3, p_{max} = none, \tau = 100$

4.4. Experiment on Synthetic Data

In this section, we first use two synthetic datasets as described in [27] to verify the performance of the proposed algorithm ERKM. In each synthetic dataset, there were three clusters.

For the first synthetic dataset: there was a total of 4 variables with 2 effective, while the other 2 were noise variables. There were 500 points scattered in three clusters with 200 in the first cluster, 100 in the second cluster, and 200 in the last one. As plotted in Figure 3, each dimension was independent and identically distributed (iid) from a normal distribution with the 2 effective features iid from $N(5,1)$ and $N(1,1)$, respectively, for Cluster 1, $N(2.5,1)$ and $N(4.0,1)$, respectively, for Cluster 2, and $N(8,1)$ for Cluster 3; the remaining 2 features were all iid from $N(0,1)$ for all clusters.

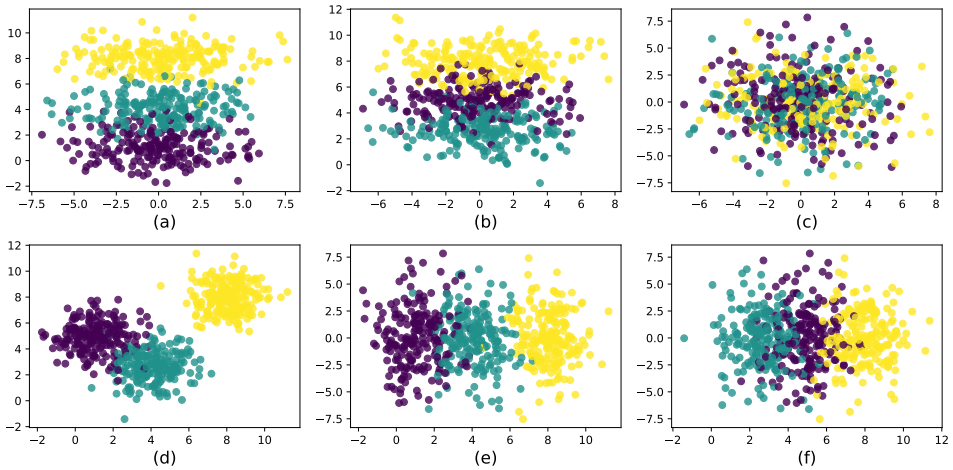

Figure 3. Synthetic dataset with three clusters in the two-dimensional subspace of x_2, x_3 and two noise dimensions x_1, x_4. (**a**) Subspace of x_1, x_2. (**b**) Subspace of x_1, x_3. (**c**) Subspace of x_1, x_4. (**d**) Subspace of x_2, x_3. (**e**) Subspace of x_2, x_4. (**f**) Subspace of x_3, x_4.

For the second synthetic dataset: there was a total of 1000 variables with 150 effective, while the other 850 were noise. Specifically, there were 250 points with 100 in Cluster 1, 50 in Cluster 2, and 100 in the other. The 150 effective features were all iid from $N(0,1)$ for the first cluster, $N(1.5,1)$ for the second cluster, and $N(2,1)$ for the third cluster; the remaining 850 features were all iid from $N(0,1)$ for the three clusters.

Now, we use the EWKM algorithm on two synthetic datasets to select a reasonable value of parameter γ using the four metrics selected in Section 4.2. Figure 4 shows the change of Acc, Fsc, ARI, and NMI for EWKM on synthetic datasets.

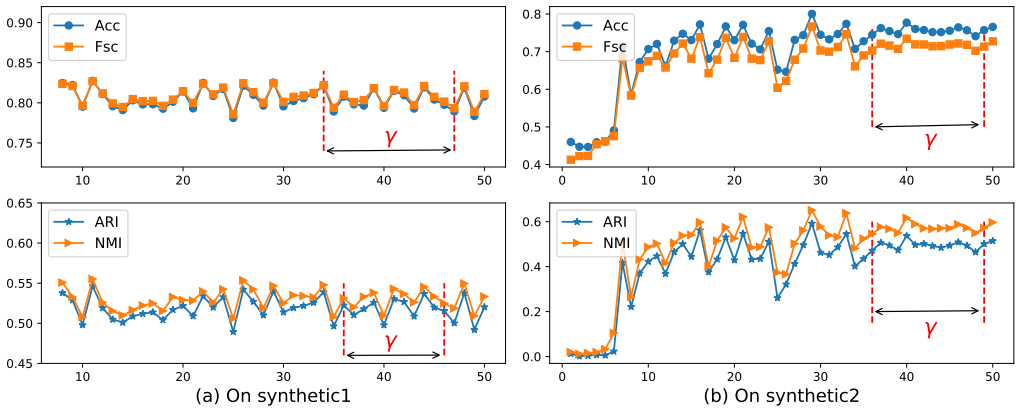

Figure 4. Four metrics' change with EWKM for different γ on two synthetic datasets. Acc: accuracy; Fsc: F-score; ARI: adjusted rand index; NMI: normal mutual information.

4.4.1. Parameter Study

As can be seen from Figure 4, when the value of γ was located in the range of 36–49 on Synthetic 1 and 36–45 on Synthetic 2, the algorithm EWKM was generally stable, where the changes in the four metrics were within 2%. However, in the other range, the performance of EWKM was greatly affected by γ, either greater than about 4% or no convergence. from Figure 5, we can see that when the range of η was about 0.04–0.05 on Synthetic 1 and 0.034–0.044 on Synthetic 2, the results of ERKM with a higher performance did not change much. Therefore, on these two synthetic datasets, we chose the two hyperparameters $\gamma = 40.0, \eta = 0.04$.

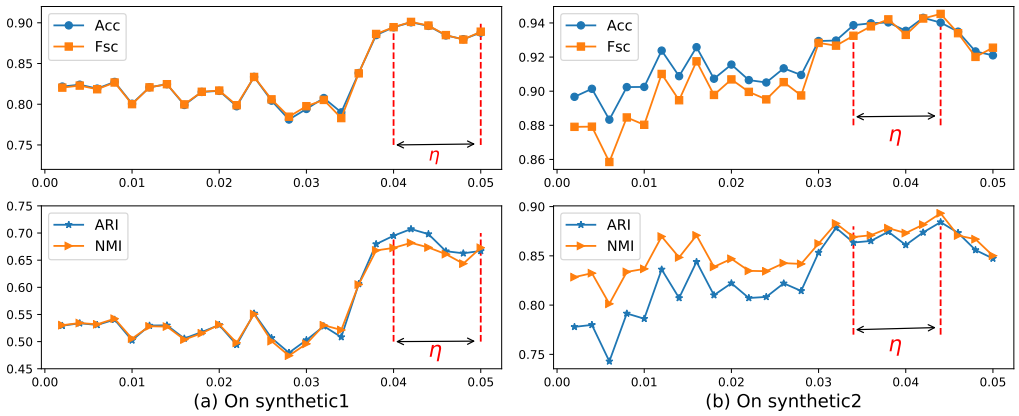

Figure 5. Four metrics' change with ERKM for different η on two synthetic datasets.

4.4.2. Results and Analysis

The experiment results shown in Figures 6 and 7 on two synthetic datasets indicated that: (1) the Acc, Fsc, ARI, and NMI of the ERKM algorithm were higher than v other algorithms; ERKM achieved the highest score on the four metrics on both datasets. In Acc and Fsc, the results of ERKM were at

least 6% higher than that of the second-best algorithm on Synthetic 1 and at least 13% on Synthetic 2. In ARI and NMI, it also was 2% higher than that of the second on Synthetic 1. Compared with the other algorithms, ERKM obtained significant achievement on Synthetic 2, where each of metric was 13% higher than that of the second-best, especially in ARI and NMI by more than 17%. (2) From the perspective of stability, ERKM, achieved the best one of all eight algorithms. From the standard deviation, ERKM achieved the minimum value among the four metrics on Synthetic 1 and good results on Synthetic 2.

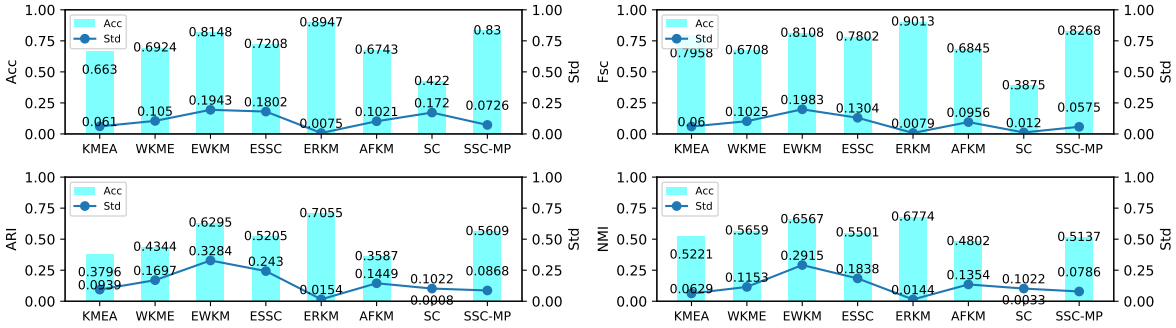

Figure 6. Results of the eight algorithms on the Synthetic 1 dataset.

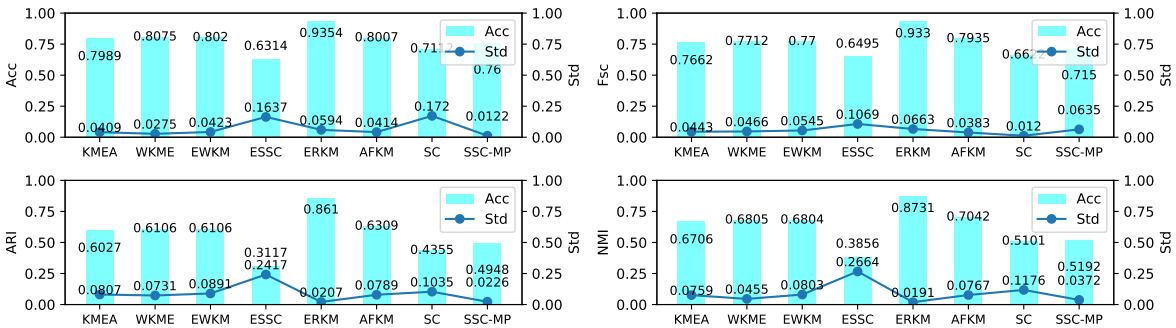

Figure 7. Results of the eight algorithms on the Synthetic 2 dataset.

4.5. Experiment on Real-Life Data

4.5.1. Parameter Study

In this section, we will also use the EWKM algorithm on seven real-life datasets to select a reasonable parameter γ with the four metrics selected in Section 4.2. Figure 8 shows the change of four metrics of EWKM on seven real-life datasets.

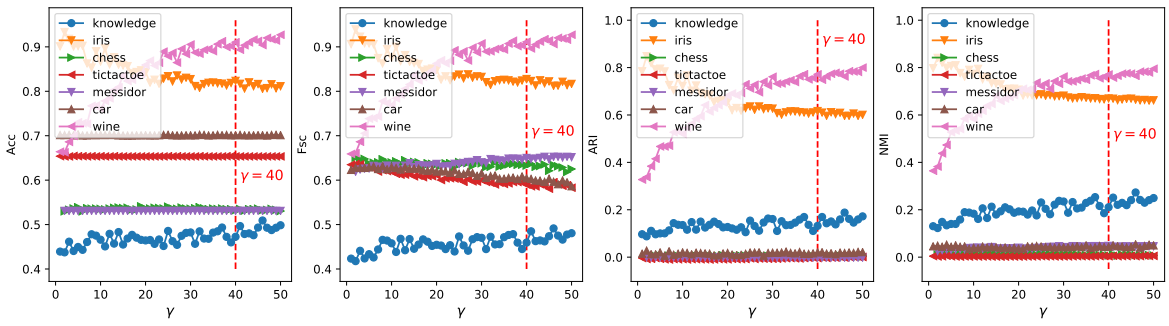

Figure 8. Four metrics' change with EWKM for different γ on real-life datasets.

It can be seen from Figure 8 that among the four metrics, EWKM had a large fluctuation mainly concentrating on the two datasets of iris and wine, and also a little fluctuation on knowledge. However, there was almost no significant change on the remaining four datasets. Therefore, this paper only determines the value of γ from the results of EWKM on the three datasets of iris, wine, and knowledge.

In Figure 8, the results of EWKM on the dataset wine changed by more than 30% for all four metrics and more than 40% for ARI and NMI. At the same time, with the increase of γ, the results of EWKM on the dataset wine was increasing and generally tended to be stable near $\gamma = 40$, with a change of less than 2%. On the iris dataset, the results of EWKM on all four metrics showed a downward trend with a change of more than 10% and generally stabilized after $\gamma > 30$. Finally, for the dataset knowledge, the EWKM algorithm had a small change for the four metrics, mainly within 10%, and with the increase of γ, the results of EWKM slightly increased, while when γ was around 40, it tended to be stable overall. In summary, through the above comparison and analysis, we found that for the two algorithms EWKM and ESSC, when γ was equal to 40, it was a reasonable value.

In order to obtain a robust range of eta, which should enable ERKM to have reasonable results for most unknown datasets, we can identify it by analyzing the results of ERKM on seven known real-life datasets. Figure 9 is a plot of the results of ERKM on seven real-life datasets. Then, we will determine a reasonable η value by analyzing the results in Figure 9. From the three metrics of Acc, ARI, and NMI, with the increase of η, the results of the ERKM algorithm showed a significant change only for the three datasets wine, iris, and knowledge. For the dataset wine, with the increase of η, the results of ERKM started to decrease slightly around $\eta = 0.01$ and tended to be stable in the vicinity of $\eta = 0.03$. For the dataset iris, the results of ERKM, from $\eta = 0.01$, started to rise and were stable at around $\eta = 0.03$. For the dataset knowledge, when $0.01 < \eta \leq 0.03$, the results of ERKM were generally in a stable state.

For Fsc, when $\eta > 0.03$, the results of ERKM on the datasets wine, iris, knowledge, and chess had a declining trend and on the dataset car had an increasing trend, while they were almost unchanged on tictactoe and messidor. When $\eta < 0.01$, the results had large fluctuations only on knowledge; and when $0.01 < \eta \leq 0.03$, the results on the seven datasets were generally stable.

In summary, through the above analysis of the ERKM algorithm on the four metrics, it had a reasonable value for $\eta = 0.03$.

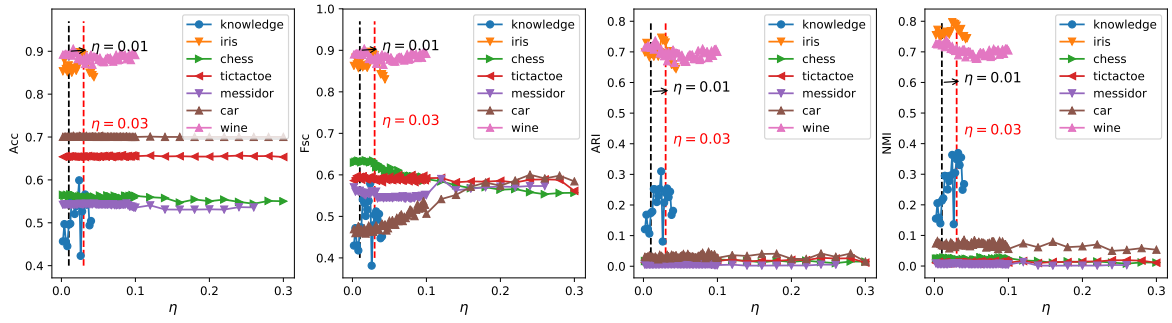

Figure 9. Four metrics' change with ERKM for different η on real-life datasets.

4.5.2. Results and Analysis

In this section, we will compare ERKM (with $\gamma = 40.0$ and $\eta = 0.03$ obtained in Section 4.5.1) with all the algorithms on seven real-life datasets and analyze the results. The final test results are tabulated in Table 3.

Table 3. Results on real-life datasets (standard deviation in brackets).

Metric	Model	Knowledge	Iris	Tictactoe	Chess	Messidor	Car	Wine
Acc	KMEA	0.4600(0.02)	0.8054(0.06)	0.6535(0.01)	0.5359(0.02)	0.5452(0.01)	0.7057(0.02)	0.9443(0.07)
	WKME	0.4897(0.06)	0.7847(0.08)	0.6534(0.00)	0.5432(0.03)	0.5382(0.01)	0.7063(0.016)	0.9471(0.07)
	EWKM	0.4789(0.09)	0.8209(0.06)	0.6534(0.00)	0.5334(0.02)	0.5329(0.00)	0.7011(0.01)	0.9024(0.08)
	ESSC	0.5244(0.02)	0.8466(0.00)	0.6539(0.00)	0.5288(0.00)	0.5308(0.00)	0.7004(0.00)	0.9506(0.00)
	AFKM	0.4780(0.04)	0.8127(0.06)	0.6534(0.00)	0.5354(0.00)	0.5416(0.01)	0.7002(0.00)	0.9399(0.08)
	SC	0.4987(0.00)	0.8066(0.00)	0.6535(0.00)	0.5222(0.00)	0.5308(0.00)	0.7002(0.01	0.8707(0.00)
	SSC-MP	0.5012(0.00)	0.7120(0.01)	0.6513(0.00)	0.5222(0.00)	0.5311(0.00)	0.7002(0.00)	0.5865(0.00)
	ERKM	0.5848(0.08)	0.9036(0.01)	0.6580(0.01)	0.5714(0.03)	0.5456(0.01)	0.7051(0.02)	0.9016(0.04)
Fsc	KMEA	0.1945(0.02)	0.8115(0.04)	0.5672(0.03)	0.5270(0.02)	0.5533(0.03)	0.4613(0.06)	0.9469(0.06)
	WKME	0.4769(0.05)	0.7979(0.07)	0.5643(0.03)	0.5350(0.03)	0.6029(0.06)	0.4551(0.05)	0.9482(0.06)
	EWKM	0.4652(0.09)	0.8264(0.05)	0.6051(0.03)	0.6210(0.05)	0.6601(0.01)	0.5206(0.06)	0.9046(0.07)
	ESSC	0.5151(0.03)	0.8477(0.00)	0.6004(0.04)	0.6642(0.00)	0.6604(0.01)	0.5156(0.05)	0.9503(0.00)
	AFKM	0.4689(0.03)	0.8158(0.05)	0.5699(0.03)	0.5895(0.00)	0.5814(0.05)	0.3864(0.04)	0.9431(0.06)
	SC	0.4355(0.21)	0.4719(0.00)	0.5709(0.01)	0.543(0.00)	0.5349(0.00)	0.3818(0.04)	0.8694(0.00)
	SSC-MP	0.5189(0.01)	0.7672(0.00)	0.5207(0.00)	0.6662(0.00)	0.5629(0.00)	0.5176(0.00)	0.5842(0.00)
	ERKM	0.5295(0.08)	0.9015(0.01)	0.6065(0.04)	0.6297(0.02)	0.5503(0.02)	0.5606(0.06)	0.8997(0.04)
ARI	KMEA	0.1318(0.02)	0.5890(0.06)	0.0140(0.02)	0.0068(0.01)	0.0070(0.00)	0.0526(0.05)	0.8580(0.11)
	WKME	0.1588(0.06)	0.5719(0.10)	0.0128(0.02)	0.0113(0.01)	0.0024(0.00)	0.0474(0.04)	0.8632(0.11)
	EWKM	0.1425(0.14)	0.6144(0.07)	−0.0028(0.02)	0.0047(0.01)	0.0000(0.00)	0.0152(0.05)	0.7545(0.12)
	ESSC	0.1965(0.03)	0.6210(0.00)	0.0023(0.02)	0.0015(0.00)	−0.0008(0.00)	0.0259(0.05)	0.8520(0.00)
	AFKM	0.1412(0.04)	0.5987(0.07)	0.0171(0.02)	0.0053(0.00)	0.0046(0.00)	0.0162(0.02)	0.8496(0.12)
	SC	0.1062(0.00)	0.1851(0.00)	0.0123(0.00)	0.0004(0.00)	0.0013(0.00)	−0.0070(0.02)	0.6458(0.00)
	SSC-MP	0.1453(0.00)	0.5520(0.00)	0.0055(0.00)	0.0013(0.00)	0.0020(0.00)	0.0266(0.00)	0.2135(0.00)
	ERKM	0.277(0.10)	0.7535(0.01)	0.0263(0.05)	0.0234(0.02)	0.0072(0.00)	0.0247(0.01)	0.8632(0.11)
NMI	KMEA	0.1945(0.02)	0.6472(0.02)	0.0100(0.01)	0.0054(0.01)	0.0088(0.01)	0.1187(0.06)	0.8474(0.08)
	WKME	0.2323(0.08)	0.6511(0.04)	0.0084(0.01)	0.0091(0.01)	0.0184(0.01)	0.0962(0.06)	0.8503(0.08)
	EWKM	0.2197(0.15)	0.6691(0.02)	0.0038(0.00)	0.0074(0.01)	0.0261(0.01)	0.0396(0.03)	0.7620(0.09)
	ESSC	0.2662(0.03)	0.6321(0.00)	0.0060(0.011)	0.0153(0.01)	0.0089(0.01)	0.0458(0.04)	0.8197(0.00)
	AFKM	0.2102(0.05)	0.6560(0.03)	0.0127(0.01)	0.0125(0.00)	0.0164(0.01)	0.0431(0.03)	0.8414(0.09)
	SC	0.1819(0.00)	0.4667(0.00)	0.0045(0.01)	0.0003(0.00)	0.0011(0.00)	0.0159(0.04)	0.649(0.00)
	SSC-MP	0.1684(0.00)	0.6930(0.01)	0.0032(0.00)	0.0006(0.00)	0.0087(0.00)	0.0671(0.00)	0.2289(0.00)
	ERKM	0.4003(0.14)	0.8026(0.01)	0.0193(0.03)	0.0291(0.03)	0.0079(0.00)	0.0526(0.06)	0.7333(0.05)

As can be seen from Table 3, for Acc, the results of ERKM for the five datasets were higher than the other seven algorithms. Among them, the results on knowledge and iris were the best, which was 6% higher than the second-best algorithm ESSC. On datasets tictactoe, chess, and messidor, it was also higher than the second-best algorithm by 0.41%, 2.8%, and 0.4%, respectively. For Fsc, ERKM performed better than the other seven algorithms on the four datasets: 1.44% higher than the second-best algorithm for the dataset knowledge, 5.38% higher for iris, 0.14% higher for tictactoe, and 4% higher for car. For ARI, the results of the ERKM algorithm on the six datasets were higher than the other seven algorithms: for the datasets knowledge, iris, tictactoe, chess, messidor, and wine, it was higher than the second-best algorithm by 8.05%, 13.2%, 1.23%, 1.21%, 0.2%, and 1.12%, respectively. For NMI, the results of the ERKM algorithm were higher than the other seven algorithms on the four datasets: 13.3% higher than the second-best algorithm for the dataset iris, 13.4% higher for knowledge, 0.93% higher for tictactoe, and 0.14% higher for chess.

From Figure 10 left, it is clear that ERKM obtained in general better results than the other second-best cluster algorithms on most datasets. Just as clearly, for the metrics of Acc and Fsc, the results of ERKM were higher than the second-best algorithms on the datasets knowledge, iris, and tictactoe; and nearest to the best algorithms on the datasets chess, messidor, and car. On the right, it shows that the results of ERKM were much better than the others on the datasets knowledge and iris. On the rest of the datasets, the results of ERKM were slightly better than others.

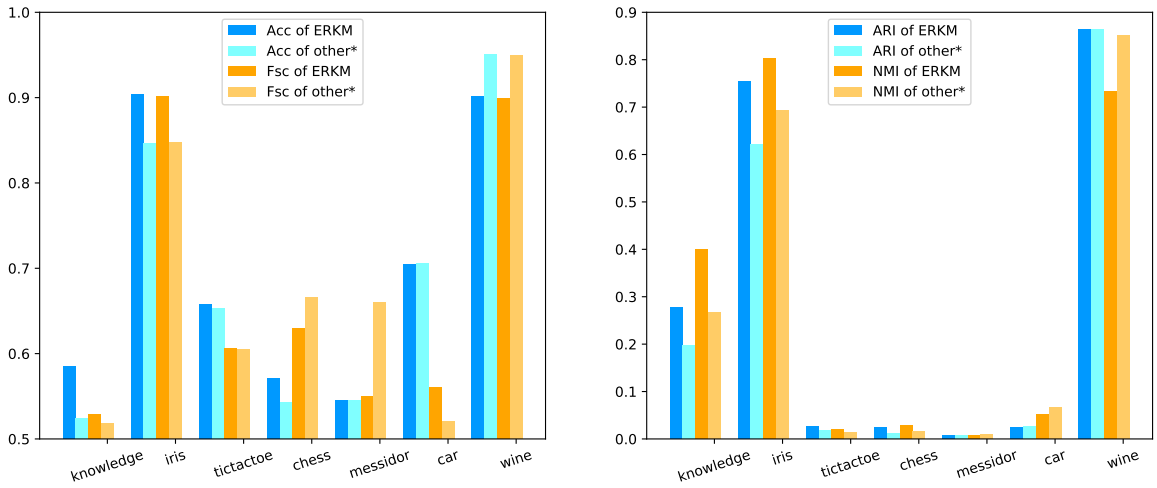

Figure 10. The results of ERKM and others* on real-life datasets. other* means the second-best algorithm, if ours is the best; or the best, if ours is not the best.

Based on the two-part analysis above, the ERKM algorithm could be better than the other seven clustering algorithms in most cases. That is to say, through the experimental results, the new between-cluster distance measure can effectively improve the clustering results.

4.6. Convergence Speed

Figure 11 shows the convergence time of the eight algorithms on the two synthetic and seven real-life datasets. The method of recording convergence time was to record the total time spent by each algorithm run on each dataset 100 times with randomly initializing the cluster center and weights. It can be seen from Figure 11 that in addition to AFKM, on the six datasets of knowledge, iris, chess, car, wine, and Synthetic 2, the convergence speed of the ERKM algorithm was significantly faster than that of the other five algorithms. It was also relatively moderate on the other three datasets, tictactoe, messidor, and Synthetic 1. Compared with the algorithm KMEA, WKME, EWKM, ESSC, SC, and SSC-MP, the convergence speed of the algorithm ERKM was the best one on the datasets knowledge, iris, car, and wine. The reason may be that the KMEA, WKKM, and EWKM algorithms did not utilize the between-cluster distance, and WKME and EWKM used complex matrix feature weighting. Although the ESSC algorithm incorporated the between-cluster information, its measure was improper. for the algorithm SC, before clustering, it needed to generate a KNN-graph. In general, ERKM had a good and competitive convergence speed.

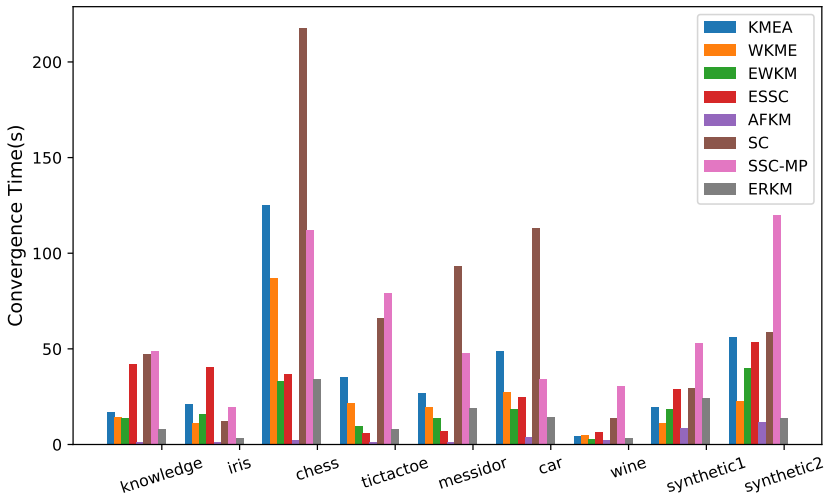

Figure 11. Convergence time.

4.7. Robustness Analysis

The two hyperparameters γ and η of the algorithm ERKM will affect the performance of the algorithm. At the same time, the ERKM is directly extended from the EWKM where γ is used to control the effect of weights on the objective function. From Figures 4 and 8, it is clear that when γ was around 40, the results of the algorithm ERKM were relatively stable on both synthetic data and real-life data. Then, we fixed $\gamma = 40.0$ to analyze the effect of η on ERKM. In ERKM, η plays the role of adjusting the influence between within-cluster distance and between-cluster distance. Therefore, the robustness of the algorithm ERKM can be analyzed by its sensitivity to hyperparameters η. From Figures 5 and 9, when η was around 0.03, the algorithm ERKM was relatively stable on both synthetic and real-life datasets.

5. Conclusions

In this paper, a new soft subspace clustering algorithm based on between-cluster distance and entropy regularization was proposed. Different from the traditional algorithms that utilize the between-cluster information by maximizing the distance between each cluster center and the global center (e.g., ESSC), ERKM effectively uses the between-cluster information by maximizing the distance between the points in the subspace that do not belong to the cluster and the center point of the cluster. Based on this assumption, this paper first designed an objective function for the algorithm and then derived the update formula by the Lagrange multiplier method. Finally, we compared several traditional subspace clustering algorithms on seven real datasets and concluded that the ERKM algorithm can achieve better clustering results in most cases.

In real-world applications, many high-dimensional data have various cluster structure features. In future research work, we plan to expand the application scope of the algorithm by modifying the objective function to adapt the case of including various complex cluster structures.

Author Contributions: L.X. and X.H. contributed equally to this work; Conceptualization, L.X. and X.H.; formal analysis, C.W. and X.H.; software, C.W. and H.Z.; validation, C.W. and H.Z.; writing, original draft preparation, L.X. and C.W.; writing, review and editing, C.W. and X.H.; visualization, H.Z.

Funding: This research was funded by the NSFC under Grant No. 61562027, the Natural Science Foundation of Jiangxi Province under Grant No. 20181BAB202024, and the Education Department of Jiangxi Province under Grant Nos. GJJ170413, GJJ180321, and GJJ170379.

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

Abbreviations

The following abbreviations are used in this manuscript:

ERKM	Entropy Regularization *k*-means
KMEA	*k*-Means
WKME	Weighted *k*-Means
EWKM	Entropy Weighted *k*-Means
ESSC	Enhanced Soft-Subspace Clustering
AFKM	Assumption-Free *k*-Means
SC	Sampling-Clustering
SSC-MP	Subspace Sparse Clustering by the greedy orthogonal Matching Pursuit
Acc	Accuracy
ARI	Adjusted Rand Index
Fsc	F-score
NMI	Normal Mutual Information

References

1. Huang, Z. Extensions to the k-means algorithm for clustering large datasets with categorical values. *Data Min. Knowl. Discov.* **1998**, *2*, 283–304. [CrossRef]
2. MacQueen, J. Some methods for classification and analysis of multivariate observations. In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Oakland, CA, USA, 21 June–18 July 1967; Volume 1, pp. 281–297.
3. Green, P.E.; Kim, J.; Carmone, F.J. A preliminary study of optimal variable weighting in *k*-means clustering. *J. Classif.* **1990**, *7*, 271–285. [CrossRef]
4. ElSherbiny, A.; Moreno-Hagelsieb, G.; Walsh, S.; Wang, Z. Phylogenomic clustering for selecting non-redundant genomes for comparative genomics. *Bioinformatics* **2013**, *29*, 947–949.
5. Deng, Z.; Choi, K.S.; Chung, F.L.; Wang, S. Enhanced soft subspace clustering integrating within-cluster and between-cluster information. *Pattern Recognit.* **2010**, *43*, 767–781. [CrossRef]
6. Sardana, M.; Agrawal, R. A comparative study of clustering methods for relevant gene selection in microarray data. In *Advances in Computer Science, Engineering & Applications*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 789–797.
7. Tang, L.; Liu, H.; Zhang, J. Identifying evolving groups in dynamic multimode networks. *IEEE Trans. Knowl. Data Eng.* **2012**, *24*, 72–85. [CrossRef]
8. Jain, A.K.; Murty, M.N.; Flynn, P.J. Data clustering: A review. *ACM Comput. Surv.* **1999**, *31*, 264–323. [CrossRef]
9. Cao, Y.; Wu, J. Projective ART for clustering datasets in high dimensional spaces. *Neural Netw.* **2002**, *15*, 105–120. [CrossRef]
10. Huang, J.Z.; Ng, M.K.; Rong, H.; Li, Z. Automated variable weighting in *k*-means type clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 657–668. [CrossRef]
11. DeSarbo, W.S.; Carroll, J.D.; Clark, L.A.; Green, P.E. Synthesized clustering: A method for amalgamating alternative clustering bases with differential weighting of variables. *Psychometrika* **1984**, *49*, 57–78. [CrossRef]
12. De Soete, G. Optimal variable weighting for ultrametric and additive tree clustering. *Qual. Quant.* **1986**, *20*, 169–180. [CrossRef]
13. De Soete, G. OVWTRE: A program for optimal variable weighting for ultrametric and additive tree fitting. *J. Classif.* **1988**, *5*, 101–104. [CrossRef]
14. Makarenkov, V.; Legendre, P. Optimal variable weighting for ultrametric and additive trees and *k*-means partitioning: Methods and software. *J. Classif.* **2001**, *18*, 245–271.
15. Wang, Y.X.; Xu, H. Noisy sparse subspace clustering. *J. Mach. Learn. Res.* **2016**, *17*, 320–360.
16. Jing, L.; Ng, M.K.; Huang, J.Z. An entropy weighting *k*-means algorithm for subspace clustering of high-dimensional sparse data. *IEEE Trans. Knowl. Data Eng.* **2007**, *19*, 1026–1041. [CrossRef]
17. Wu, K.L.; Yu, J.; Yang, M.S. A novel fuzzy clustering algorithm based on a fuzzy scatter matrix with optimality tests. *Pattern Recognit. Lett.* **2005**, *26*, 639–652. [CrossRef]
18. Huang, X.; Ye, Y.; Zhang, H. Extensions of kmeans-type algorithms: A new clustering framework by integrating intracluster compactness and intercluster separation. *IEEE Trans. Neural Netw. Learn. Syst.* **2014**, *25*, 1433–1446. [CrossRef] [PubMed]
19. Huang, X.; Ye, Y.; Guo, H.; Cai, Y.; Zhang, H.; Li, Y. DSKmeans: A new kmeans-type approach to discriminative subspace clustering. *Knowl.-Based Syst.* **2014**, *70*, 293–300. [CrossRef]
20. Han, K.J.; Narayanan, S.S. Novel inter-cluster distance measure combining GLR and ICR for improved agglomerative hierarchical speaker clustering. In Proceedings of the 2008 IEEE International Conference on Acoustics, Speech and Signal Processing, Las Vegas, NV, USA, 31 March–4 April 2008; pp. 4373–4376.
21. Bai, L.; Liang, J.; Dang, C.; Cao, F. A novel fuzzy clustering algorithm with between-cluster information for categorical data. *Fuzzy Sets Syst.* **2013**, *215*, 55–73. [CrossRef]
22. Bai, L.; Liang, J. The k-modes type clustering plus between-cluster information for categorical data. *Neurocomputing* **2014**, *133*, 111–121. [CrossRef]
23. Zhou, J.; Chen, L.; Chen, C.P.; Zhang, Y.; Li, H.X. Fuzzy clustering with the entropy of attribute weights. *Neurocomputing* **2016**, *198*, 125–134. [CrossRef]
24. Deng, Z.; Choi, K.S.; Jiang, Y.; Wang, J.; Wang, S. A survey on soft subspace clustering. *Inf. Sci.* **2016**, *348*, 84–106. [CrossRef]

25. Chang, X.; Wang, Y.; Li, R.; Xu, Z. Sparse k-means with ℓ_∞/ℓ_0 penalty for high-dimensional data clustering. *Stat. Sin.* **2018**, *28*, 1265–1284.
26. Witten, D.M.; Tibshirani, R. A framework for feature selection in clustering. *J. Am. Stat. Assoc.* **2010**, *105*, 713–726. [CrossRef] [PubMed]
27. Pan, W.; Shen, X. Penalized model-based clustering with application to variable selection. *J. Mach. Learn. Res.* **2007**, *8*, 1145–1164.
28. Zhou, J.; Chen, C.P. Attribute weight entropy regularization in fuzzy c-means algorithm for feature selection. In Proceedings of the 2011 International Conference on System Science and Engineering, Macao, China, 8–10 June 2011; pp. 59–64.
29. Sri Lalitha, Y.; Govardhan, A. Improved Text Clustering with Neighbours. *Int. J. Data Min. Knowl. Manag. Process* **2015**, *5*, 23–37.
30. Forghani, Y. Comment on "Enhanced soft subspace clustering integrating within-cluster and between-cluster information" by Z. Deng et al. (Pattern Recognition, vol. 43, pp. 767–781, 2010). *Pattern Recognit.* **2018**, *77*, 456–457. [CrossRef]
31. Das, S.; Abraham, A.; Konar, A. Automatic clustering using an improved differential evolution algorithm. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2008**, *38*, 218–237. [CrossRef]
32. McLachlan, G.J.; Peel, D.; Bean, R. Modelling high-dimensional data by mixtures of factor analyzers. *Comput. Stat. Data Anal.* **2003**, *41*, 379–388. [CrossRef]
33. Chang, X.; Wang, Q.; Liu, Y.; Wang, Y. Sparse Regularization in Fuzzy c-Means for High-Dimensional Data Clustering. *IEEE Trans. Cybern.* **2017**, *47*, 2616–2627. [CrossRef]
34. Bezdek, J.C. A convergence theorem for the fuzzy ISODATA clustering algorithms. *IEEE Trans. Pattern Anal. Mach. Intell.* **1980**, *PAMI-2*, 1–8. [CrossRef]
35. Selim, S.Z.; Ismail, M.A. K-means-type algorithms: A generalized convergence theorem and characterization of local optimality. *IEEE Trans. Pattern Anal. Mach. Intell.* **1984**, *PAMI-6*, 81–87. [CrossRef]
36. Bachem, O.; Lucic, M.; Hassani, H.; Krause, A. Fast and provably good seedings for k-means. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 55–63.
37. Tarn, C.; Zhang, Y.; Feng, Y. Sampling Clustering. *arXiv* **2018**, arXiv:1806.08245.
38. Tschannen, M.; Bölcskei, H. Noisy subspace clustering via matching pursuits. *IEEE Trans. Inf. Theory* **2018**, *64*, 4081–4104. [CrossRef]