

Article

Sparse k-Means Clustering with Lasso-Based Feature Selection

Miin-Shen Yang * and Shazia Parveen

Department of Applied Mathematics, Chung Yuan Christian University, Taoyuan 32023, Taiwan

* Correspondence: msyang@cycu.edu.tw

Abstract

Clustering high-dimensional data is challenging when irrelevant or weakly informative features obscure the underlying cluster structure. To address this issue, we propose two sparse k-means clustering algorithms, S-KM1 and S-KM2, that perform cluster estimation and feature selection by sparse feature weighting simultaneously. Clustering has been applied to a wide range of disciplines, including image segmentation, social network analysis, medical imaging, market segmentation, anomaly detection, etc. In clustering algorithms, the most popular and widely used method is k-means. However, the k-means algorithm always treats feature (attribute) components in datasets equally. In practice, different features may contribute unequally to the cluster structure and should therefore not necessarily be weighed equally. This has been demonstrated by Yang and Benjamin (2023) with two types of sparse possibilistic c-means (PCM) methods called SPCM1 and SPCM2, using the Lasso concept. Motivated by SPCM1 and SPCM2, we propose two sparse k-means clustering algorithms, S-KM1 and S-KM2, in this paper. The proposed methods S-KM1 and S-KM2 consider the k-means clustering subject to different feature weight constraints based on the Lasso and L1 penalty so that they can shrink the irrelevant features towards zero and achieve sparsity in features. The proposed S-KM1 and S-KM2 are compared with some of the most popular sparsity-clustering techniques on several numerical and real-life datasets using different clustering performance measures. Comparisons and experimental findings demonstrate the validity, efficacy, and effectiveness of the proposed S-KM1 and S-KM2 clustering algorithms, and they also surpass the existing advanced algorithms in terms of efficiency and usefulness.

Keywords: clustering; k-means (KM); possibilistic c-means (PCM); feature weights; sparsity; Lasso; sparse KM

Academic Editor: George A. Tsihrintzis

Received: 9 August 2026

Revised: 24 September 2026

Accepted: 24 September 2026

Published: 26 September 2026

Copyright: © 2026 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 \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Clustering is a method of grouping objects based on how similar and dissimilar they are to one another. This strategy clusters data into various groups to boost the most similarity between the data points in the same group and to enhance the dissimilarity between distinct groups. Algorithms involving clustering have been widely used for analyzing the characteristics of items for identifying groups of comparable objects. In data sciences, grouping data is a key tool, while it has emerged as a fundamental part of multivariate analysis [1]. It is challenging to classify and differentiate relevant traits from irrelevant ones due to the vast volume of data gathered. One intriguing and emerging research area is the formulation of mathematical models that concentrate on selecting variables and features from data with a high number of dimensions [2,3]. Probability model-based and

nonparametric clustering are the two main approaches in multivariate analysis. In this paper, we focus on nonparametric clustering. There are several popular nonparametric clustering techniques in the literature, such as k-means [4,5], fuzzy c-means [6,7], possibilistic c-means [8,9], and spectral clustering [10,11]. However, these clustering techniques may be effective for low-dimensional data, but they are not suitable for high-dimensional datasets. In general, data with a high number of dimensions may include characteristics that are seen unimportant or sparse, meaning only a few essential features are chosen and utilized. As a result, it is crucial to prioritize variable and feature selection when we handle high-dimensional datasets.

Furthermore, an innovative method has been proposed to retrieve 3D human poses from shadows, enhancing conventional techniques by utilizing multiview locality-sensitive sparse coding during the retrieval phase [12]. Sparse constraints and an enhanced RELU operator were combined to develop a hierarchical deep word embedding model to handle click feature prediction from visual features [13]. In the domain of regression, the concept of variable and feature selection is broadly acknowledged. Penalizing the L_1 norm provides an appealing answer to this problem because it leads to a sparse solution, and thus, it can be used for variable and feature selection. There are several ways to obtain the sparse solution, among which Lasso, proposed by Tibshirani [14], is popular. Lasso [14] stands for “least absolute shrinkage and selection operator”, which can force the coefficients of certain variables to exactly zero and cause them to shrink towards zero. The Lasso-based k-means clustering framework was subsequently put forward by Witten and Tibshirani [15] as a sparsity approach using the Lasso constraint in the k-means clustering algorithm as a mechanism for selecting features. The sparse k-means clustering [16] based on the Lasso approach outperforms other competing techniques in terms of feature selection and identified clusters. In addition to limited k-means clustering, Castro and Pu [17] proposed a sparse clustering approach based on a pairwise dissimilarity framework, in which feature weights are incorporated into the dissimilarity measure. In contrast, S-KM2 incorporates an L_1 penalty directly into the within-cluster sum of squares objective of k-means to promote sparse feature weighting; later, Qiu et al. [18] proposed sparse fuzzy c-means (SFCM) clustering.

In 2023, Yang and Benjamin [19] expanded the concept of sparse clustering to the possibilistic c-means (PCM) algorithm. They introduced two sparse PCM algorithms, called SPCM1 and SPCM2. Motivated by SPCM1 and SPCM2, we propose the two sparse k-means (S-KM) clustering algorithms, called S-KM1 and S-KM2. S-KM1 is similar to the sparse k-means of Witten and Tibshirani [15], in which the objective function is subject to the Lasso constraint. However, S-KM2 considers the within-cluster sum of squares objective function of k-means with an L_1 penalty of feature weights. Both S-KM1 and S-KM2 can shrink the irrelevant features towards zero to achieve a sparse solution. Multiple artificial and actual datasets are applied to evaluate the effectiveness and achievement of the S-KM1 and S-KM2 algorithms in comparison with a few present-day clustering algorithms. Comparisons and results from experiments verify the usefulness and efficiency of the proposed S-KM1 and S-KM2. The rest of the paper is organized as follows. In Section 2, some related works in the literature are reviewed. The two proposed sparse k-means algorithms, S-KM1 and S-KM2, and some theoretical properties are illustrated in Section 3. In Section 4, we present the construction of a convergence theorem for the S-KM2 algorithm. Section 5 presents a comparative analysis of the proposed S-KM1 and S-KM2 models and existing methods based on artificial datasets as well as actual datasets. Finally, we present our conclusions in Section 6.

2. Related Works

Clustering with k-means is widely used due to its simplicity, efficiency, and ease of interpretation. Hartigan and Wong [20] presented an efficient iterative k-means algorithm that partitions observations into a predefined number of clusters by minimizing the within-cluster sum of squares, seeking a locally optimal partition. However, conventional k-means assigns equal importance to all features. This assumption can be restrictive for high-dimensional data, where irrelevant or weakly informative features may obscure the underlying cluster structure and adversely affect clustering performance. To address the unequal contribution of features, several weighted k-means approaches have been developed, such as those of Huang et al. [21] and Jing et al. [22]. An entropy-weighted k-means approach in which feature weights are determined through an entropy-based regularization term was developed by Jing et al. [22]; their method determines feature importance by entropy-based weighting, rather than through an L_1 sparsity penalty.

Let $X = \{x_i \in R^d, i = 1, \dots, n\}$ be the data set with data points x_i in a d -dimensional Euclidean space R^d , $U = [u_{ik}]_{n \times c}, i = 1, \dots, n, k = 1, \dots, c$ be the $n \times c$ membership values, where u_{ik} is a binary variable (*i.e.* $u_{ik} \in \{0, 1\}$) stating whether or not the data point x_i is a member of the k^{th} cluster, $k = 1, 2, \dots, c$ and the cluster center is denoted by v_k . The notation u_{ik} represents the membership of the data point x_i in the k^{th} cluster, where $i = 1, \dots, n$ and $k = 1, \dots, c$. The following is the k-means objective function [20]:

$$J(U, V; X) = \sum_{i=1}^n \sum_{k=1}^c u_{ik} \|x_i - v_k\|^2 \quad (1)$$

where, $X = \{x_1, \dots, x_n\}$ is the dataset, $U = (u_{ik})$ is the membership or assignment matrix, $V = \{v_1, \dots, v_c\}$ is the set of cluster centers and $\|x_i - v_k\|$ is the Euclidean distance between the data point x_i and the cluster center v_k . The objective function $J(U, V; X)$ denotes the within-cluster sum-of-squared distances, which is minimized with respect to U and V for the given dataset X . The optimization variables are U and V , whereas X is the fixed observational data. The membership functions and cluster centers updating equations are as follows, respectively:

$$v_{kj} = \frac{\sum_{i=1}^n u_{ik} x_{ij}}{\sum_{i=1}^n u_{ik}}, \quad j = 1, \dots, d \quad (2)$$

$$u_{ik} = \begin{cases} 1, & \text{if } \|x_i - v_k\|^2 = \min_l \|x_i - v_l\|^2, \\ 0, & \text{otherwise} \end{cases}, \quad i = 1, \dots, n, \quad k = 1, \dots, c \quad (3)$$

In Equation (2), for the fixed cluster assignments, the cluster center v_{kj} is updated as the membership-weighted mean of the j^{th} feature among the observations assigned to cluster k . The denominator $\sum_{i=1}^n u_{ik}$ in Equation (2) normalizes the weighted sum by the total membership of cluster k , rather than imposing an additional normalization on the cluster center. For the k-means, u_{ik} either 0 or 1; therefore, $\sum_{i=1}^n u_{ik}$ simply gives the number of observations assigned to cluster k . Consequently, v_{kj} indicates the membership-weighted

mean of the j^{th} feature within cluster k . In Equation 3, $\|x_i - v_k\|$ is the Euclidean distance between the data point x_i and the cluster center v_k , where $i = 1, \dots, n$ and $k = 1, \dots, c$.

For high-dimensional datasets, traditional k-means may be less effective because it assigns equal importance to all features, including irrelevant or redundant variables. Consequently, this limitation highlights the need for clustering methods that can assign different importance to features while retaining the simplicity and computational efficiency of k-means. This motivates the development of our sparse k-means approaches considered in subsequent studies. In particular, incorporating feature selection into the clustering process can help identify informative variables and reduce the influence of uninformative features. In general, conventional clustering methods may suffer from the presence of irrelevant and noisy features for high-dimensional data, which can obscure the true clustering structure. Therefore, sparse clustering addresses this issue by simultaneously identifying meaningful cluster structure and selecting the most informative features, and therefore it can improve clustering performance and interpretability. In sparse clustering, Witten and Tibshirani [15] extended the k-means clustering based on Lasso. Their method is based on the idea that maximizing the between-cluster sum of squares (BCSS) and minimizing the within-cluster sum of squares (WCSS) are equivalent. The feature-specific BCSS for variable j is specifically defined as

$$BCSS_j(C) = \frac{1}{n} \sum_{i=1}^n \sum_{i'=1}^n d_{ii'j} - \sum_{k=1}^c \frac{1}{n_k} \sum_{i,i' \in C_k} d_{ii'j} \quad (4)$$

where the squared distance between observations i and i' for the variable j is defined as $d_{ii'j} = (x_{ij} - x_{i'j})^2$, n is number of observations, the variable $j = 1, \dots, d$, and the cluster $k = 1, \dots, c$, $i, i' = 1, \dots, n$ are the indices of observations, C_k is the set of observations assigned to cluster k , C is the complete partition of observations into c clusters, and $n_k = |C_k|$ is the number of observations in cluster k . The Lasso-based k -means [15] considered the BCSS subject the Lasso constraints of feature weights with the following objective function:

$$\max_{C_1, \dots, C_c, w} \left\{ \sum_{j=1}^d w_j \left(\frac{1}{n} \sum_{i=1}^n \sum_{i'=1}^n d_{ii'j} - \sum_{k=1}^c \frac{1}{n_k} \sum_{i,i' \in C_k} d_{ii'j} \right) \right\} \quad (5)$$

subject to $\|w\|_2^2 \leq 1$, $\|w\|_1 \leq s$, $w_j \geq 0 \forall j$, where w_j is a feature weight corresponding to every feature j , and s is a tuning parameter.

However, this Lasso-based k -means [15] remains sensitive to initialization and the choice of the sparsity parameter, motivating the development of more flexible sparse clustering methods in which Qiu et al. [18] suggested sparse (Lasso-based) fuzzy c-means (S-FCM) to enhance FCM for handling high-dimensional datasets. The sparse S-FCM follows a similar framework to that of Witten and Tibshirani [15]. The ultimate difference is that FCM is used in place of k-means. The S-FCM objective function [18] is as follows:

$$\max_{U, V, w} \sum_{j=1}^d w_j \left(D_j - \sum_{i=1}^n \sum_{k=1}^c u_{ik}^m d_{ik,j} \right) \quad (6)$$

subject to $\|w\|_2^2 \leq 1$, $\|w\|_1 \leq s$, $w_j \geq 0 \forall j$ where $w = (w_1, \dots, w_p) \in R^d$ are the feature weights, $V = \{v_1, \dots, v_k\}$ are the cluster centers, $U = [u_{ik}]_{n \times K}$ is the matrix of fuzzy membership degrees, and $D_j = \sum_{i=1}^n (x_{ij} - \bar{x}_j)^2$ with $\bar{x}_j = \sum_{i=1}^n x_{ij} / n$ and $d_{ik,j} = (x_{ij} - v_{kj})^2$. The tuning

parameter s is used to control the number of attributes to pick from. In Qiu et al. [18], they used gap statistic or binary search as a choice of s . In Equations (5) and (6), the maximization is performed with respect to the corresponding optimization variables within a single objective function, while the remaining variables are fixed. Thus, these equations represent a single optimization problem rather than two separate maximization procedures.

Afterward, Yang and Benjamin [19] extended the Lasso-based k-means and S-FCM clustering to possibilistic c-means (PCM) clustering and then proposed two sparse PCM algorithms, namely SPCM1 and SPCM2. SPCM1 in Yang and Benjamin [19] uses the Lasso constraint on feature weights, similar to Lasso-based k-means and S-FCM. However, SPCM2 uses the feature weight L_1 regularization as a penalty term to the PCM objective function. The SPCM1 objective function [19] is as follows:

$$\max_{U,V,w} \left\{ \sum_{j=1}^d w_j \left(\frac{1}{n} \sum_{i=1}^n \sum_{t'=1}^n (x_{ij} - x_{t'j})^2 - \left[\sum_{i=1}^n \sum_{k=1}^c u_{ik}^m (x_{ij} - v_{kj})^2 + \lambda \sum_{i=1}^n \sum_{k=1}^c (1 - u_{ik})^m \right] \right) \right\} \quad (7)$$

subject to $\|w\|_2^2 \leq 1, \|w\|_1 \leq s, w_j \geq 0 \forall j$, where w_j is the feature weight, and s is a tuning parameter. The SPCM2 objective function [19] is given below:

$$J_{S-PCM2}(U, V, W) = \sum_{i=1}^n \sum_{k=1}^c \sum_{j=1}^d w_j u_{ik}^m (x_{ij} - v_{kj})^2 + \lambda \sum_{i=1}^n \sum_{k=1}^c (1 - u_{ik})^m + \alpha \|w\|_1 \quad (8)$$

where $\lambda = \beta / m^2 c$, w_j is feature weights, and $\|w\|_1 = \sum_{j=1}^d |w_j|, w_j \geq 0 \forall j$ is a L_1 lasso penalty. v_{kj} and u_{ik} are the cluster centers and membership functions, respectively.

It is to be noted that the k -means algorithm is simple and faster to execute because it updates cluster centers and assigns data points. On the other hand, the PCM algorithm demands complex computations for membership degree modifications, which increases its computational time as well. The objective function of k -means has fewer parameters, with $m=1$, whereas the PCM has a larger number of parameters with fuzzifier $m > 1$. The SPCM1 algorithm requires more execution time due to incorporating gap statistics, which helps to obtain s and thus controls the sparsity on feature weights. Similar complexity can be observed in the objective function of SPCM2 due to the presence of additional parameters. Notably, more recent studies have demonstrated advanced sparse clustering and feature selection for high-dimensional data. In Ma et al. [23], the authors proposed a correlation-based clustering-assisted sparse learning model based on Lasso for solving a battery electric vehicle market demand forecasting problem. Tutsoy and Sumbul [24] considered a supervised learning approach to enhance multiclass cardiovascular disease detection with Kernel-PSO-based feature selection by combining kernel learning for bio-inspired optimization to identify informative features. In Dong et al. [25], the authors proposed an unsupervised discriminative feature selection method that can simultaneously perform to learn feature-wise sparse projection for discriminative feature selection by combining fuzzy membership learning and $\ell_{2,0}$ -norm constrained projection learning. Motivated by these mentioned points, we demonstrate the S-KM1 and S-KM2 algorithms by incorporating the L_1 penalty term with a smaller number of parameters compared to the SPCM1 and SPCM2 algorithms in the next section.

3. The Proposed Sparse k-Means Clustering Algorithms

The proposed two approaches for sparse k-means (S-KM), called S-KM1 and S-KM2, are discussed in this section. The first approach of S-KM is based on the k-means objective

function with respect to Lasso feature weight constraints, called S-KM1. The second approach within S-KM is to incorporate the k-means objective function with an L_1 regularization term of feature weights, called S-KM2. Both S-KM1 and S-KM2 can be penalized by irrelevant attributes being pushed towards zero by using the Lasso feature weight constraints or incorporating L_1 regularization of feature weights. First, we propose the following S-KM1 objective function:

$$\max_{U,V,w} \left\{ \sum_{j=1}^d w_j \left(\frac{1}{n} \sum_{i=1}^n \sum_{i'=1}^n (x_{ij} - x_{i'j})^2 - \left[\sum_{i=1}^n \sum_{k=1}^c u_{ik} (x_{ij} - v_{kj})^2 \right] \right) \right\} \quad (9)$$

subject to $\|w\|_1 \leq s, \|w\|_2^2 \leq 1, w_j \geq 0 \forall j$, for each feature j , in which w_j is the feature weight with restriction $\|w\|_2^2 \leq 1$ and a tuning parameter s with restriction as $1 \leq s \leq \sqrt{d}$, d is dimension or characteristics of the dataset. In Equation (9), $u_{ik} \in \{0, 1\}$ is a binary variable representing whether or not the data point x_i is a member of the k^{th} cluster, and the k^{th} cluster center is denoted by v_k .

To maximize Equation (9) with respect to $U = [u_{ik}]_{n \times K}$ and $V = \{v_1, \dots, v_k\}$, while keeping $w = [w_1, w_2, \dots, w_d]$ fixed, is equivalent to minimize the following equation:

$$J(U, V) = \sum_{j=1}^d w_j \sum_{i=1}^n \sum_{k=1}^c u_{ik} (x_{ij} - v_{kj})^2 \quad (10)$$

Similar to the k-means, the equations for minimizing Equation (10) can be obtained as follows:

$$u_{ik} = \begin{cases} 1, & \text{if } \sum_{j=1}^d w_j (x_{ij} - v_{kj})^2 = \min_{1 \leq l \leq c} \sum_{j=1}^d w_j (x_{ij} - v_{lj})^2, 1 \leq k \leq c, 1 \leq i \leq n \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

$$v_{kj} = \frac{\sum_{i=1}^n u_{ik} x_{ij}}{\sum_{i=1}^n u_{ik}} \quad (12)$$

We next let a_j be

$$a_j = \frac{1}{n} \sum_{i=1}^n (x_{ij} - \bar{x}_j)^2 - \left[\sum_{i=1}^n \sum_{k=1}^c u_{ik} (x_{ij} - v_{kj})^2 \right] \quad (13)$$

where $\bar{x}_j = \sum_{i=1}^n x_{ij} / n$. To maximize Equation (9) with respect to $w = [w_1, w_2, \dots, w_d]$ while keeping $U = [u_{ik}]_{n \times K}$ and $V = \{v_1, \dots, v_k\}$ fixed, is equivalent to the following equation:

$$\max_w \{w^T a\} \quad (14)$$

subject to $\|w\|_1 \leq s, \|w\|_2^2 \leq 1, w_j \geq 0 \forall j$, where a_j is denoted by Equation (13) and $w = [w_1, w_2, \dots, w_d] \in R^d$ are feature weights, s is L_1 bound to weights w , and $a = [a_1, a_2, \dots, a_d]$.

Equation (9) is a single scalar constrained objective function, in which the membership variables, cluster centers, and feature weights are three blocks of decision variables, rather than three distinct objectives. Specifically, this optimization procedure updates these variable blocks sequentially while optimizing the same objective function, while promoting sparse feature weights due to the Lasso constraint. Therefore, no scalarization or multi-objective trade-off scheme is required. The optimization is performed using an alternating optimization procedure: for fixed feature weights, the membership variables and cluster centers are optimized, leading to Equations (10)–(12); subsequently, for fixed memberships and cluster centers, the feature weights are optimized subject to the Lasso-type constraint, leading to Equation (14). Thus, all updates optimize the same scalar objective function, while the Lasso constraint promotes sparse feature weights and automatic feature selection.

For comparing the proposed S-KM1 and Witten and Tibshirani's method [15], we provide the following descriptions. In Witten and Tibshirani [15], the conventional k-means criterion is first reformulated in terms of the BCSS. Their Lasso-based k-means objective function, given in Equation (5), maximizes the weighted BCSS subject to Lasso-type constraints on the feature weights. In the formulation, the contribution of each cluster is represented at the cluster level and explicitly involves the cluster-size quantity n_k . In contrast, our proposed S-KM1 is formulated directly from the SPCM1 framework with Lasso-constrained feature weights. As given in Equation (9), S-KM1 is expressed in terms of the observation-level binary membership variables, cluster centers, and feature weights. In particular, the cluster-size term n_k that appears explicitly in the BCSS-based objective of Witten and Tibshirani [15] is not explicitly present in our Equation (9). Our subsequent derivation is therefore performed directly from Equation (9), leading to the equivalent minimization formulation of Equation (10) and corresponding to the updated procedures of Equations (11)–(14). Thus, although both approaches of the S-KM1 and Witten and Tibshirani's method employ sparse feature weighting and Lasso-type constraints, the objective functions are formulated differently. Witten and Tibshirani's method [15] are explicitly based on a weighted BCSS formulation, whereas our S-KM1 is derived directly from the observation-level k-means objective with Lasso constraints on the feature weights. We acknowledge that the formulations of both the proposed S-KM1 and the k-means method are mathematically related. Therefore, we do not claim that the absence of an explicit n_k term alone makes S-KM1 completely unrelated to Witten and Tibshirani's method. Instead, we highlight that S-KM1 has its own optimization process and a distinct objective function formulation and derivation.

Theorem 1. *The convex problem $\max_w \{w^T a\}$ subject to $\|w\|_1 \leq s$, $\|w\|_2^2 \leq 1 \forall j, w_j \geq 0$ is solved by $w = S(a, \Delta) / \|S(a, \Delta)\|_2$, where $w = [w_1, \dots, w_d]$ and $a = [a_1, \dots, a_d]$. The soft-thresholding operator S is defined by $S(a, \Delta) = [S(a_1, \Delta), \dots, S(a_d, \Delta)]$ with $S(a_j, \Delta) = \text{sign}(a_j) \max(|a_j| - \Delta, 0)$, $j = 1, \dots, d$. We have that $\Delta = 0$ if $\|w\|_1 \leq s$; otherwise, we appropriate $\Delta > 0$ so that $\|w\|_1 = s$. The tuning parameter s exists such that $1 \leq s \leq \sqrt{d}$, in which the number of dimensions is denoted by d .*

Proof. The Karush-Kuhn-Tucker (KKT) criteria are useful for proving the proposition. Throughout the optimization problem (14), we require w that minimizes the expression $\{-w^T a\}$ subject to $\|w\|_1 \leq s, \|w\|_2^2 \leq 1 \forall j, w_j \geq 0$. The objective function of the convex problem (14) with corresponding constraints is

$$\text{Minf}(W) = \text{Min}(-w^T a) \quad (*_1)$$

Subject to the constraints of $\|w\|_1 - s \leq 0$, $\|w\|_2^2 \leq 1 \forall j$ for $w_j \geq 0$. The feature-weight optimization problem has the constraint $g(w) = \|w\|_2^2 - 1 \leq 0$ or $\|w\|_2^2 \leq 1$. In order to minimize $(*_1)$, the first KKT condition related to the constraint is $\|w\|_2^2 - 1 \leq 0$. Let δ be the Lagrange multiplier associated with this constraint $\delta \geq 0$; then, $\delta(\|w\|_2^2 - 1) = 0$ by complementary slackness implies that the two cases are $\delta = 0$ or $\|w\|_2^2 = 1$, while δ is the KKT multiplier. Thus, to minimize $(*_1)$, we apply KKT conditions on constraints as follows:

$$\text{Inequality Condition/Constraint-1: } \|w\|_1 - s \leq 0 \Rightarrow \Delta(\|w\|_1 - s) = 0 \quad (*_2)$$

$$\text{Equality Condition/Constraint-2: } \|w\|_2^2 - 1 \leq 0, \delta(\|w\|_2^2 - 1) = 0 \quad (*_3)$$

Hence, $\delta = 0$ or $\|w\|_2^2 - 1 = 0$. Next, we introduce the Lagrange multiplier (LM) with Δ and δ as follows as follows:

$$\begin{aligned} L(W, \Delta, \delta) &= -w^T a + \Delta \text{constraint 1} + \delta \text{constraint 2} \\ L(W, \Delta, \delta) &= -w^T a + \Delta(\|w\|_1 - s) + \delta(\|w\|_2^2 - 1) \end{aligned} \quad (*_4)$$

Stationary Condition/Condition-3: We take the partial derivative of the Lagrangian with respect to w , and then set it equal to zero as follows:

$$\begin{aligned} \frac{\partial L}{\partial w} &= \left\{ -w^T a + \Delta(\|w\|_1 - s) + \delta(\|w\|_2^2 - 1) \right\} = 0, \\ \frac{\partial L}{\partial w} &= -a \frac{\partial L}{\partial w}(w^T) + \Delta \frac{\partial L}{\partial w}(\|w\|_1 - s) + \delta \frac{\partial L}{\partial w}(\|w\|_2^2 - 1), \quad \frac{\partial L}{\partial w} - a + 2\delta w + \Delta \text{sign}(w) = 0 \text{ (stationary)}, \\ \Delta \partial(\|w\|_1 - s) &= \Delta \text{sign}(w) \Rightarrow w = \frac{a - \Delta \text{sign}(w)}{2\delta}. \end{aligned}$$

Consider w to minimize $(*_4)$, we implement soft-thresholding approximation. The Lasso penalty term can be deal separately, and we will locate the possible result for $\Delta \partial(\|w\|_1 - s) = \Delta \text{sign}(w)$. There are two possibilities for optimal solution of w_j .

Case 1: Let us consider $w_j \neq 0$ if $w_j > 0$, $a_j > \Delta$ and if $w_j < 0$, $a_j < -\Delta$, then

$$|a_j| > \Delta, \text{ and so } \text{sign}(w_j) = \text{sign}(a_j).$$

Case 2: Let us consider $w_j = 0$; the sub-differential of the LASSO L_1 term

$$(\therefore \Delta \partial(\|w\|_1 - s) = \Delta \text{sign}(w)) \text{ is the interval } [-1, 1], \text{ and for this, we have the optimal}$$

condition as follows: $0 \in a_j + \Delta[-1, 1] \Leftrightarrow |a_j| \in \Delta[-1, 1] \Rightarrow |a_j| \leq \Delta$. Thus, by considering

the above-mentioned Cases 1 and 2, we have the optimal result for this w_j as follows:

$$w_j = \begin{cases} 0 & ; \text{ when } |a_j| \leq \Delta \\ \frac{a_j - \Delta \text{sign}(a_j)}{2\delta} & ; \text{ when } |a_j| > \Delta \end{cases} . \text{ Let } \forall j, S(a_j, \Delta) = \text{sign}(a_j) \max(|a_j| - \Delta, 0) \text{ be}$$

the soft-thresholding function. We know that $\forall j$,

$$a_j - \Delta \text{sign}(a_j) = \text{sign}(a_j) \max(|a_j| - \Delta, 0) = S(a_j, \Delta) . \quad \text{Thus, we have}$$

$$w_j = \frac{a_j - \Delta \text{sign}(a_j)}{2\delta} = \frac{S(a_j, \Delta)}{2\delta} \quad \forall j, \delta \geq 0 . \quad \text{That is, } w = \frac{S(a, \Delta)}{2\delta}, \delta \geq 0, \quad \text{where}$$

$$S(a, \Delta) = [S(a_1, \Delta), \dots, S(a_d, \Delta)], \quad w = [w_1, \dots, w_d] \quad \text{and} \quad a = [a_1, \dots, a_d] . \text{ We differentiate}$$

$(*_3)$, with respect to δ by imposing KKT condition 2. We have $\|w\|_2^2 = 1$ and then

$$\|w\|_2 = 1 . \quad \text{Thus, we obtain } \left\| \frac{S(a, \Delta)}{2\delta} \right\|_2 = \|w\|_2 = 1, \delta \geq 0, \quad \text{and then}$$

$$\|S(a, \Delta)\| \delta = |2\delta| = 2\delta . \text{ Since we have } w = S(a, \Delta)/2\delta, \text{ we obtain the required solution}$$

$$\text{as follows: } w = S(a, \Delta) / \|S(a, \Delta)\|_2 . \quad \square$$

Furthermore, in contrast to traditional k-means, where all features contribute equally to the distance computation, S-KM1 assigns a feature-specific weight to each variable and estimates these weights jointly with the cluster structure. The feature weights represent the relative importance of individual features in determining the clustering structure; the cluster centers represent the corresponding cluster prototypes; and the membership variables determine the allocation of observations to clusters. The S-KM1 formulation incorporates these feature weights into the traditional k-means objective and imposes a Lasso-type constraint on the weight vector. This constraint promotes sparsity by encouraging some feature weights to become zero, thereby reducing the contribution of irrelevant or less informative features to the clustering process. Thus, Equation (9) simultaneously determines the cluster assignments, cluster centers, and feature weights, while the Lasso constraint enables automatic selection of the most relevant features. In Equation (9), the objective function is defined in terms of the binary membership variables, cluster centers, and feature weights. The membership variables determine the assignment of each observation to a cluster, while the cluster centers represent the corresponding cluster prototypes.

The feature weights determine the contribution of individual features to the clustering objective. The Lasso constraint restricts the feature weight vector and promotes sparsity by encouraging the weights of irrelevant or weakly informative features to become zero. This feature selection property is particularly useful for high-dimensional data containing irrelevant or weakly informative variables. To optimize the objective function, we use an alternating optimization strategy. First, when the feature weights are fixed, the optimization with respect to the memberships and cluster centers reduces to the corresponding k-means optimization problem. This leads to the minimization formulation in Equation (10) and the membership and cluster center updating equations given in Equations (11) and (12), respectively. Thus, observations are assigned according to their

weighted distances from the cluster centers, while the centers are updated according to the observations assigned to each cluster. Next, when the memberships and cluster centers are fixed, the objective function is optimized with respect to the feature weights. To simplify this optimization, the feature-dependent part of the objective is represented by the quantity introduced in Equation (13). The resulting feature weight optimization problem is given in Equation (14), subject to the Lasso constraint.

In practical terms, u_{ik} indicates the membership of observation i in cluster k . Since it is a binary variable, an observation is assigned to a cluster when its corresponding membership value is one. The quantity v_k , represents the center, or representative profile, of cluster k . The distance between an observation and a cluster center therefore determines how well that observation fits the corresponding cluster. The feature weight w_j represents the relative importance of feature j in forming the clusters. A larger w_j means that feature j has a greater influence on the weighted distance and consequently contributes more strongly to the clustering structure. Conversely, a small weight indicates that the feature contributes less to distinguishing the clusters. As Equation (14) is a constrained convex optimization problem, the Karush–Kuhn–Tucker (KKT) conditions are employed to obtain its solution. The resulting solution involves a soft-thresholding operation, which is particularly important for achieving sparsity in the feature weights. Features whose contributions are insufficient to satisfy the optimization criterion can therefore obtain zero weights, whereas informative features retain non-zero weights. Therefore, the derivation of S-KM1 consists of three alternating optimization steps: updating the cluster memberships, updating the cluster centers, and updating the feature weights. Until the clustering solution meets the convergence criterion, these procedures are repeated. The S-KM1 algorithm is a summary of the resulting process. The preceding derivation establishes the updating rules required to minimize the S-KM1 objective function. Specifically, the optimization with respect to the cluster assignments, cluster centers, and feature weights leads to the corresponding updating equations. These equations provide the computational basis of the proposed algorithm: given the current feature weights and cluster centers, the observations are reassigned to the nearest cluster; the cluster centers are then updated using the resulting assignments; and the feature weights are updated subject to the Lasso constraint. This alternating optimization procedure is repeated until the clustering solution satisfies the specified convergence criterion. The resulting implementation of S-KM1 is summarized in Algorithm 1.

Consequently, the proposed S-KM1 algorithm can be described as follows:

Algorithm 1. The S-KM1 Algorithm

S1: Initialize $W^{(0)}$ as $w_1 = \dots = w_d = 1/\sqrt{d}$. Fix $\varepsilon \geq 0, 2 \leq c \leq n$. Pick $V^{(0)}$ at random, and let $t = 1$.

S2: Compute $U^{(t)}$ with $W^{(t-1)}$ and $V^{(t-1)}$ by using Equation (11).

S3: Compute $V^{(t)}$ with $U^{(t)}$ by using Equation (12).

S4: Based on Proposition 1, to optimize Equation (9) with respect to w while keeping U

and V fixed, use $w = \frac{S(a, \Delta)}{\|S(a, \Delta)\|_2}$, where $w = [w_1, \dots, w_d]$, $S(a, \Delta) = [S(a_1, \Delta), \dots, S(a_d, \Delta)]$

with $S(a_j, \Delta) = \text{sign}(a_j) \max(|a_j| - \Delta, 0)$ and $a_j = \frac{1}{n} \sum_{i=1}^n (x_{ij} - \bar{x}_j)^2 - \left[\sum_{i=1}^n \sum_{k=1}^c u_{ik} (x_{ij} - v_{kj})^2 \right]$,

in which $\Delta = 0$ if $\|w\|_1 \leq s$; Otherwise, select a suitable $\Delta > 0$ so that $\|w\|_1 = s$.

S5: Data point x_i should be allocated to the cluster k having the maximum value of membership, i.e., $u_{ik} \geq u_{ik'}, \forall k'$ or a resulting membership matrix in the form of U , where u_{ik} is the updated membership matrix.

S6: IF $\sum_{j=1}^d |w_j^{(t)} - w_j^{(t-1)}| / \sum_{j=1}^d |w_j^{(t-1)}| < \varepsilon$, where $w_j^{(t)}$ is t^{th} updated step of w_j , then STOP;

OTHERWISE, set $t = t + 1$ and return to S2.

In Equation (9), s is a tuning parameter used to control how many features are allocated. In S-KM1, more attributes must be picked, as s is significant. We apply the technique used in Tibshirani et al. [26] to determine the appropriate setting of the tuning parameter s via the gap statistic to our S-KM1 algorithm. To determine the magnitude of the tuning parameter s for S-KM1, the gap statistic is applied as follows:

- To create the permuted dataset $\{X_1, X_2, \dots, X_Z\}$, we independently permute the data points in every feature.
- The target function $T(s)$ of s at each potential value of s is first generated using the S-KM1 on the original dataset, where $T(s)$ is as follows:

$$T(s) = \sum_{j=1}^d w_j \left(\frac{1}{n} \sum_{i=1}^n \sum_{i'=1}^n (x_{ij} - x_{i'j})^2 - \left[\sum_{i=1}^n \sum_{k=1}^c u_{ik} (x_{ij} - v_{kj})^2 \right] \right)$$

- Leveraging the permuted dataset $\{X_1, X_2, \dots, X_Z\}$ with S-KM1 enables us to determine the objective function $T(s)$ on every iteration of s . We designate the consequence $T_z(s)$, $z=1, 2, \dots, Z$.
- We compute the gap statistic as follows:

$$\text{gap}(s) = \log(T(s)) - [1/Z] \sum_{z=1}^Z \log(T_z(s)).$$
- We obtain $s^* = \max(\text{gap}(s))$.

The gap statistic is the difference between actual dispersion and the reference dispersion (generated by uniform distribution) under the null hypothesis. We generate Z reference datasets using uniform distribution and cluster each one by giving within-dispersion $T_z(s)$. The degree of the tuning parameter s is determined by the maximum $\text{Gap}(s)$ to be $s^* = \max(\text{gap}(s))$. The procedure for the maximum gap is lengthy to execute ([15,26]). All of this generally incurs a great computational burden. To improve it, we next propose simpler sparse k-means clustering.

As already mentioned, the estimation of the tuning parameter s in S-KM1 is complicated and time-consuming. Thus, we propose another simpler strategy, called S-KM2. The proposed S-KM2 objective function is as follows:

$$J_{S-KM2}(U, V, W) = \sum_{i=1}^n \sum_{k=1}^c \sum_{j=1}^d u_{ik} w_j (x_{ij} - v_{kj})^2 + \lambda \|w\|_1 \quad (15)$$

Theorem 2. The necessary conditions for minimizing $J_{S-KM2}(U, V, W)$ are as follows:

$$u_{ik} = \begin{cases} 1, & \text{if } \sum_{j=1}^d w_j (x_{ij} - v_{kj})^2 = \min_{1 \leq l \leq c} \sum_{j=1}^d w_j (x_{ij} - v_{lj})^2 \\ 0, & \text{otherwise} \end{cases} \quad (16)$$

$$v_{kj} = \sum_{i=1}^n u_{ik} x_{ij} / \sum_{i=1}^n u_{ik} \quad (17)$$

$$w_j = \begin{cases} \frac{-\sum_{i=1}^n \sum_{k=1}^c u_{ik} (x_{ij} - v_{kj})^2 - \lambda}{\left\| -\sum_{i=1}^n \sum_{k=1}^c u_{ik} (x_{ij} - v_{kj})^2 - \lambda \right\|_2}, & \text{when } \theta < -\lambda \\ 0, & \text{when } |\theta| \leq \lambda \text{ where } \theta = \sum_{i=1}^n \sum_{k=1}^c \mu_{ik} (x_{ij} - v_{kj})^2 \\ \frac{-\sum_{i=1}^n \sum_{k=1}^c u_{ik} (x_{ij} - v_{kj})^2 + \lambda}{\left\| -\sum_{i=1}^n \sum_{k=1}^c u_{ik} (x_{ij} - v_{kj})^2 + \lambda \right\|_2}, & \text{when } \theta > \lambda \end{cases} \quad (18)$$

Proof. We first minimize $J_{S-KM2}(U, V, W) = \sum_{i=1}^n \sum_{k=1}^c \sum_{j=1}^d u_{ik} w_j (x_{ij} - v_{kj})^2 + \lambda \|w\|_1$ with respect to u_{ik} by considering v_{kj} and w_j fixed. It is equivalent to minimizing

$$J(U) = \sum_{i=1}^n \sum_{k=1}^c \sum_{j=1}^d u_{ik} w_j (x_{ij} - v_{kj})^2 \text{ and will get the same equation as Equation (11). Thus, we}$$

obtain Equation (16). We next consider the minimizer of $J_{S-KM2}(U, V, W)$ with respect to v_{kj} by considering u_{ik} and w_j fixed. Applying a partial derivative of $J_{S-KM2}(U, V, W)$ with respect to v_{kj} and setting it equal to zero, we have

$$\frac{\partial J_{S-KM2}}{\partial v_{kj}} = \frac{\partial}{\partial v_{kj}} \left[\sum_{i=1}^n \sum_{k=1}^c \sum_{j=1}^d u_{ik} w_j (x_{ij} - v_{kj})^2 \right] = 0, \text{ and have } -2 \sum_{i=1}^n w_j u_{ik} (x_{ij} - v_{kj}) = 0. \text{ Thus,}$$

we obtain $v_{kj} = \sum_{i=1}^n u_{ik} x_{ij} / \sum_{i=1}^n u_{ik}$, which is Equation (17). We next compute the updating

weight equation of w_j by considering u_{ik} and v_{kj} be fixed. We know that

$$\frac{\partial J_{S-KM2}}{\partial w} = \frac{\partial}{\partial w} \left[\sum_{i=1}^n \sum_{k=1}^c \sum_{j=1}^d u_{ik} w_j (x_{ij} - v_{kj})^2 \right] + \frac{\partial}{\partial w} [\lambda \|w\|_1]. \text{ Since the term } \lambda \|w\|_1 \text{ in}$$

$J_{S-KM2}(U, V, W)$ cannot be directly used by the derivative, we separately consider

$$J_{S-KM2}(U, V, W) \text{ into two term of } J_1(W) = \sum_{i=1}^n \sum_{k=1}^c \sum_{j=1}^d u_{ik} w_j (x_{ij} - v_{kj})^2 \text{ and}$$

$J_2(W) = \lambda \|w\|_1$, respectively. For $J_1(W)$, we use derivative but cannot directly obtain a

solution in it. Thus, we add a reasonable term in $J_1(W)$ with

$$J_1^*(W) = \sum_{i=1}^n \sum_{k=1}^c \sum_{j=1}^d u_{ik} w_j (x_{ij} - v_{kj})^2 + \phi(\|w\|_2^2 - 1) \text{ and have an extra updating equation}$$

in the algorithm. Using the partial derivative of $J_1^*(W)$ with respect to w_j and setting it equal to zero, we have

$$\frac{\partial J_1^*}{\partial w_j} = \frac{\partial}{\partial w_j} \left[\sum_{i=1}^n \sum_{k=1}^c \sum_{j=1}^d u_{ik} w_j (x_{ij} - v_{kj})^2 + \phi(\|w\|_2^2 - 1) \right] = \sum_{i=1}^n \sum_{k=1}^c u_{ik} (x_{ij} - v_{kj})^2 + 2\phi w_j = 0. \text{ Thus, we}$$

$$\text{have } w_j = - \sum_{i=1}^n \sum_{k=1}^c u_{ik} (x_{ij} - v_{kj})^2 / 2\phi. \text{ Similarly, taking } \frac{\partial J_1^*}{\partial \phi} \text{ and setting it equal to zero,}$$

$$\text{we obtain } \frac{\partial J_1^*}{\partial \phi} = \|w\|_2^2 - 1 = 0, \text{ and then } \|w\|_2^2 = 1. \text{ The values of } \phi \text{ must be selected such}$$

$$\text{that } \|w\|_2^2 = 1. \text{ Thus, we obtain the solution as } \left\| - \sum_{i=1}^n \sum_{k=1}^c u_{ik} (x_{ij} - v_{kj})^2 / 2\phi \right\|_2 = 1, \text{ then}$$

$$\left\| - \sum_{i=1}^n \sum_{k=1}^c u_{ik} (x_{ij} - v_{kj})^2 \right\|_2 = \|2\phi\|_2^2. \text{ However, } \|2\phi\|_2^2 = 2\phi. \text{ By substituting the value of } 2\phi, \text{ we}$$

$$\text{obtain the updating equation for weight } w_j \text{ with } w_j = \frac{- \sum_{i=1}^n \sum_{k=1}^c u_{ik} (x_{ij} - v_{kj})^2}{\sum_{i=1}^n \sum_{k=1}^c \sum_{j=1}^d \left[-u_{ik} (x_{ij} - v_{kj})^2 \right]}. \text{ We}$$

can find that the updating equation of w_j has $\sum_{j=1}^d w_j = 1$. We now come to use sub-differential to solve the second term $\lambda \|w\|_1$ in $J_{S-KM2}(U, V, W)$. Let us consider a function

$f(w) = \lambda |w|, \lambda > 0$; then, we have the sub differential function $f(w)$ w.r.t w , which

can be expressed as $\frac{\partial}{\partial w_j} \lambda \|w\|_1 = \lambda \frac{\partial}{\partial w_j} \left(\sum_{j=1}^d |w_j| \right) = \lambda \text{sign}(w_j)$. By using the defini-

$$\text{tion, we can write the above term } \lambda \text{sign}(w_j) \text{ as follows: } \lambda \text{sign}(w_j) = \begin{cases} -\lambda; & \text{when } w_j < 0 \\ [-\lambda, \lambda]; & \text{when } w_j = 0. \\ \lambda; & \text{when } w_j > 0 \end{cases}$$

$$\text{Thus, we have } \frac{\partial J_{S-KM2}(U, V, W, \phi)}{\partial w} = T^* + \begin{cases} -\lambda; & \text{when } w_j < 0 \\ [-\lambda, \lambda]; & \text{when } w_j = 0, \\ \lambda; & \text{when } w_j > 0 \end{cases} \text{ where}$$

$$T = \frac{\partial T}{\partial w_j} = \sum_{i=1}^n \sum_{k=1}^c u_{ik} (x_{ij} - v_{kj})^2 + 2\phi w_j, \text{ we already have } T^* = \frac{\partial T}{\partial w} = \sum_{i=1}^n \sum_{k=1}^c u_{ik} (x_{ij} - v_{kj})^2 + 2\phi w_j = 0,$$

and there are three possibilities to solve this expression, detailed as follows:

Case 1: For $w_j < 0$, let $T^* - \lambda < 0$. Then, $T^* - \lambda = \sum_{i=1}^n \sum_{k=1}^c u_{ik} (x_{ij} - v_{kj})^2 + 2\phi w_j - \lambda$. Let

$$\theta = \sum_{i=1}^n \sum_{k=1}^c u_{ik} (x_{ij} - v_{kj})^2 \Rightarrow \theta + 2\phi w_j - \lambda < 0 \Rightarrow w_j = (-\theta + \lambda) / (2\phi) \text{ as } w_j < 0, \theta > \lambda.$$

Since $w_j < 0$, $(-\theta + \lambda) / (2\phi) < 0$, i.e., $-\theta + \lambda < 0$, $\lambda < \theta$ or $\theta > \lambda$.

Case 2: For $w_j > 0$, let $T^* + \lambda > 0$. $T^* + \lambda = \sum_{i=1}^n \sum_{k=1}^c u_{ik} (x_{ij} - v_{kj})^2 + 2\phi w_j + \lambda > 0$. Let

$$\theta = \sum_{i=1}^n \sum_{k=1}^c u_{ik} (x_{ij} - v_{kj})^2 \Rightarrow \theta + 2\phi w_j + \lambda = 0 \Rightarrow w_j = (-\theta - \lambda)/(2\phi) \text{ as } w_j > 0, \theta < -\lambda.$$

Since $w_j > 0$, $(-\theta - \lambda)/(2\phi) > 0$, i.e., $-\theta - \lambda > 0$, and $\theta < -\lambda$.

Case 3: For $w_j = 0$, the interval $[T^* - \lambda, T^* + \lambda]$ must contain 0 so that $w_j = 0$ is to be an

optimum. For $T^* - \lambda \geq 0$, $T^* - \lambda = \theta + 2\phi w_j - \lambda \geq 0$, $w_j \geq 0$, then $w_j = (-\theta + \lambda)/(2\phi)$; $w_j \geq 0$,

$\theta \leq \lambda$, and therefore, $\frac{-\theta + \lambda}{2\phi} \geq 0 \Rightarrow -\theta + \lambda \geq 0$. Thus, we have $\theta \leq \lambda$. For $T^* - \lambda \leq 0$,

$T^* - \lambda = \theta + 2\phi w_j - \lambda \leq 0$, $w_j \leq 0$, then $w_j = \frac{-\theta - \lambda}{2\phi}$; $w_j \leq 0$, $-\lambda \leq \theta$, and $\frac{-\theta + \lambda}{2\phi} \leq 0 \Rightarrow -\theta + \lambda \leq 0$.

Thus, we have $\theta \geq -\lambda$. By combining both equations, we have $w_j = 0$, when $\theta \in [-\lambda, \lambda]$.

Thus, the updated equation for the weights w_j^* can be written as

$$w_j = \begin{cases} \frac{-\theta - \lambda}{2\phi} & \text{for } \theta < -\lambda \\ 0 & \text{for } |\theta| \leq \lambda \\ \frac{-\theta + \lambda}{2\phi} & \text{for } \theta > \lambda \end{cases}, \text{ since } \theta = \sum_{i=1}^n \sum_{k=1}^c u_{ik} (x_{ij} - v_{kj})^2 \text{ into the}$$

above equation of w_j , we obtain the updated Equation (18) of the weight w_j . $\square$

The second proposed method, S-KM2, is formulated differently from S-KM1 by incorporating an L_1 -regularization term directly into the k-means objective rather than imposing sparsity through a Lasso constraint. The purpose of this formulation is to simultaneously minimize the within-cluster dispersion and control the magnitude of the feature weights. Consequently, S-KM2 provides a penalized optimization framework in which feature selection is achieved directly through regularization. In the S-KM2 objective function, the optimization is performed with respect to the membership variables, cluster centers, and feature weights. For fixed feature weights and cluster centers, minimizing the objective with respect to the membership variables leads to the membership updating Equation (16). Next, with fixed memberships and feature weights, the objective is minimized with respect to the cluster centers. Taking the partial derivative with respect to the cluster center and setting it equal to zero gives the updating Equation (17). The feature weights are optimized after the cluster centers and memberships have been updated. This step is the central component that distinguishes S-KM2 from ordinary k-means and from the constrained formulation of S-KM1. In S-KM2, the membership and center updates establish the current cluster structure, while the regularized feature weight update evaluates the contribution of each feature. The informative features can retain non-zero weights,

whereas features whose contribution is insufficient relative to the regularization penalty are shrunk to zero. Consequently, subsequent cluster assignments are based primarily on the informative variables, reducing the influence of irrelevant or noisy dimensions.

Overall, Equations (16)–(18) should therefore be viewed as complementary components of S-KM2: Equation (16) updates the cluster memberships, Equation (17) updates the cluster centers, and Equation (18) updates the feature weights and performs feature selection. Repeating these updates produces a sparse clustering solution in which the final cluster structure is determined primarily by the relevant features. The role of the regularization parameter is also important. It determines the strength of the penalty applied to the feature weights. Increasing the regularization strength increases the amount of shrinkage and makes it more difficult for a feature to retain a non-zero weight. Conversely, weaker regularization allows more features to retain non-zero weights. Thus, the parameter controls the balance between clustering fit and sparsity. The parameter selection procedure described after the S-KM2 algorithm uses binary search to determine an appropriate value of the regularization parameter. The binary search procedure then searches for a suitable regularization value, and Equation (18) is used to update the feature weights. This process is repeated until the specified convergence condition is satisfied.

The derivation above provides the updating rules for the S-KM2 objective function with an explicit L_1 penalty on the feature weights. These update equations directly determine the iterative procedure used by S-KM2. At each iteration, the cluster assignments and cluster centers are updated according to the current feature weights, followed by an update of the feature weights that accounts for the L_1 penalty. In this way, clustering and feature selection are performed jointly within the same optimization procedure. The iterative process continues until the objective function or the cluster assignments satisfy the convergence criterion. The complete computational procedure is given in Algorithm 2. The preceding derivation also explains how sparsity is achieved in the proposed algorithms. For fixed cluster assignments and cluster centers, the feature weight optimization is derived directly from the proposed objective function. The Lasso term penalizes the feature weights and encourages unnecessary variables to receive smaller weights. The optimality conditions further show that a feature whose contribution to the clustering structure is insufficient can attain an optimal weight of exactly zero.

Consequently, irrelevant or weakly informative features can be eliminated directly during the optimization rather than through a separate feature selection procedure. More specifically, the L_1 regularization induces sparsity through the optimality condition of the feature weight subproblem. For a given feature, if its contribution to reducing the within-cluster dispersion is insufficient to compensate for the corresponding Lasso penalty, the optimality condition at the boundary ($w_j = 0$) is satisfied. Consequently, zero becomes an optimal value for the corresponding feature weight. This thresholding effect of the L_1 penalty allows features that provide little or no contribution to the cluster structure to be assigned exactly zero weights, thereby excluding them from the subsequent clustering criterion. Hence, feature selection is achieved intrinsically within the optimization process rather than through a separate preprocessing or post-processing step. The sparse feature weight update is coupled with the updates of the cluster assignments and cluster centers through an alternating optimization scheme. At each iteration, one set of variables is optimized, while the remaining variables are fixed. Consequently, each update does not increase the objective function, yielding a non-increasing sequence of objective values. Since the objective function is bounded below, this sequence converges. Therefore, the introduction of Lasso-based feature selection does not constitute a separate procedure that alters the resulting clusters; instead, sparsity is achieved simultaneously with

cluster optimization. Hence, the Lasso mechanism provides a principled basis for assigning zero weights to irrelevant or weakly informative features, while the alternating optimization preserves the convergence properties established in the theoretical analysis.

Thus, the proposed S-KM2 algorithm can be summarized as follows, with its flowchart shown in Figure 1.

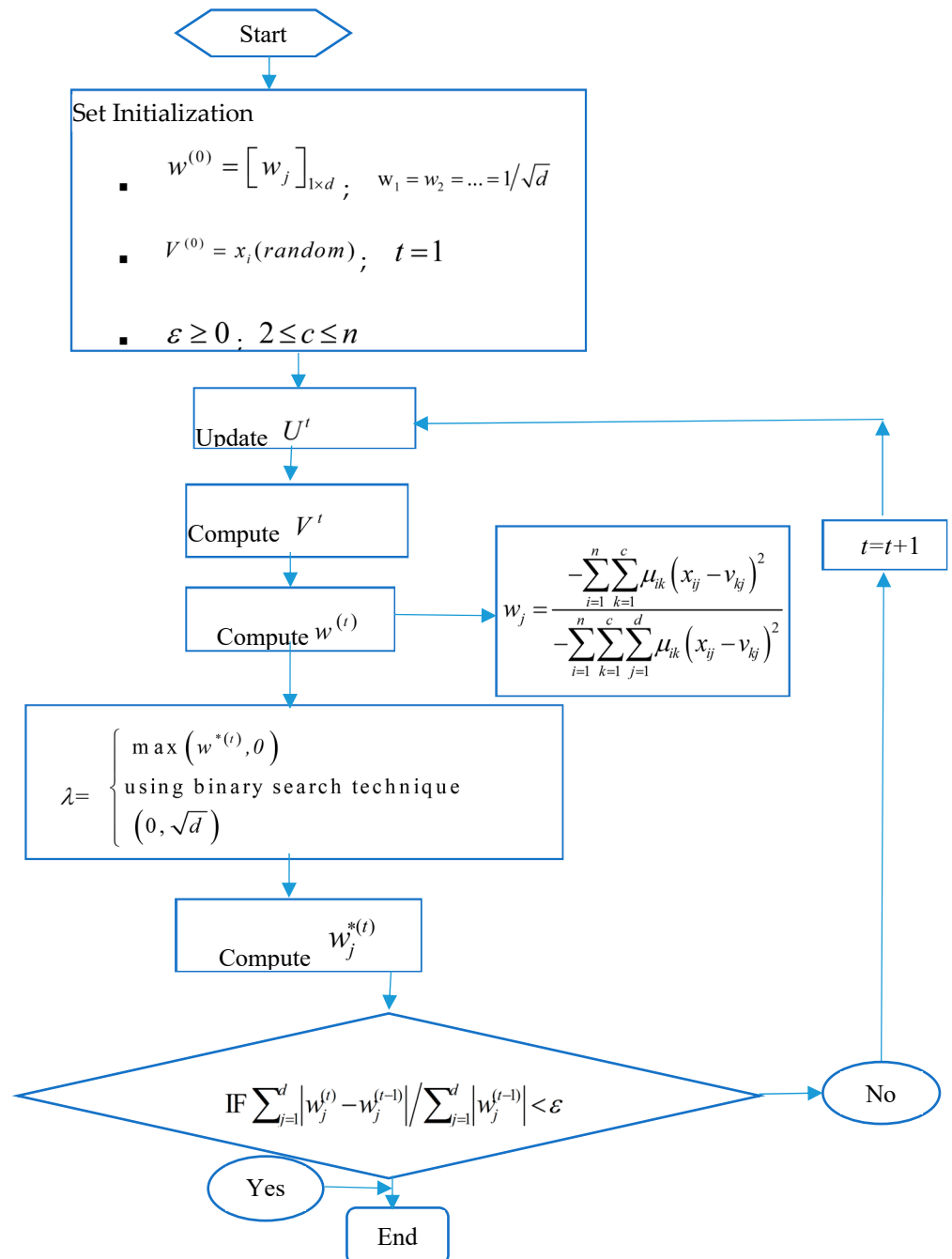


Figure 1. The flowchart of the S-KM2 algorithm.

Algorithm 2. The S-KM2 Algorithm

- S1:** Initialize feature weights as $w^{(0)} = [w_j]_{1 \times d}$ with $w_1 = w_2 = w_3 = \dots = 1/\sqrt{d}$. Fix $\varepsilon \geq 0$, and $2 \leq c \leq n$ in which c is the total count of clusters. Pick $V^{(0)}$ at random, and let the value of t equal 1.
- S2:** Update $U^{(t)}$ with $W^{(t-1)}$ and $V^{(t-1)}$ using Equation (16).
- S3:** Compute $V^{(t)}$ with $U^{(t)}$ using Equation (17).
- S4:** Compute $w^{(t)}$ with $U^{(t)}$ and $V^{(t)}$ by the equation $w_j = \frac{-\sum_{i=1}^n \sum_{k=1}^c \mu_{ik} (x_{ij} - v_{kj})^2}{-\sum_{i=1}^n \sum_{k=1}^c \sum_{j=1}^d \mu_{ik} (x_{ij} - v_{kj})^2}$.
- S5:** Choose λ via the binary search method among $\max(w^{*(t)}, 0)$ and the interval $(0, \sqrt{d})$ that is the weighted bound.
- S6:** Compute $w_j^{*(t)}$ with λ by using Equation (18).
- S7:** IF $\sum_{j=1}^d |w_j^{(t)} - w_j^{(t-1)}| / \sum_{j=1}^d |w_j^{(t-1)}| < \varepsilon$, where $w_j^{(t)}$ is the t^{th} step updating w_j . Then give off; OTHERWISE, set $t = t + 1$, and return to S2.

Next, we will explore the process of selecting the parameter λ . The cluster centers and feature weights are initialized by the S-KM2 algorithm, and then we find the membership values. We can modify weights and cluster centers by utilizing them. To find the parameter λ , the technique of binary searching is applied, which might generate sparsity. To obtain the value λ , we might use the updated weighted equation of weights in Step-4. Next, we determine $w^* = \max(w_j^*, 0)$. By applying the technique of binary searching [27], we select the most favorable value of λ from $[w^*, (a, \sqrt{d})]$, in which a is positive. We can obtain the optimal value for λ by this strategy, and then we can update weights by using the updated cluster centers. Once the condition specified in Step-7 is achieved in accordance with a threshold value ε , generally set at 1×10^{-5} , the program terminates. The weight quantity calculated in the step S4 is an intermediate quantity used for determining the regularization parameter λ ; it should not be interpreted as the final feature weight update. After λ is selected, the final feature weights are obtained from the penalized update in Equation (18). The L_1 -regularization and the cluster updates are not separate procedures; rather, they are coupled within the same iterative optimization framework. As the feature weights are updated at each iteration, the resulting sparse feature representation directly influences the subsequent cluster assignments and cluster centers. Conversely, the updated cluster structure determines the feature contributions used in the next feature weight update. Feature selection and cluster formation can potentially be performed simultaneously thanks to this iterative interaction, which also makes it possible to omit unnecessary informative features from the clustering criterion by using zero feature weights. In S-KM2, the L_1 -penalty is directly involved in determining the feature weights during the iterative clustering process. Consequently, features that provide insufficient contribution to the clustering structure can attain zero weights, whereas informative features retain non-zero weights and continue to contribute to cluster formation. The resulting feature weights are then incorporated into the subsequent cluster assignment and cluster center updates. Thus, feature selection and cluster estimation are jointly performed within the same iterative optimization procedure.

The contribution of S-KM2 lies in its specific integration of the L_1 -penalized formulation with the k-means clustering framework, the derivation of the associated feature weight updating procedure, and the theoretical analysis of the resulting iterative optimization scheme. The proposed formulation provides a unified mechanism in which feature weighting, feature selection, and cluster estimation are performed jointly. Importantly, we do not claim that the use of L_1 /Lasso regularization itself is novel. Rather, following the L_1 penalization idea of SPCM2, our contribution is its formulation and development within the k-means framework, together with the derivation of the associated feature weight updating procedure and the theoretical analysis of the resulting iterative optimization scheme. In S-KM1, sparsity originates from the Lasso-type constraint on the feature weight vector in Equation (9). With memberships and cluster centers fixed, the feature weight subproblem is given by Equation (14), whose KKT-based solution yields a soft-thresholding rule. Consequently, insufficiently informative features receive zero weights, while relevant features retain non-zero weights. In S-KM2, sparsity is introduced directly through the feature weight regularization term in Equation (15). At each iteration, the memberships and cluster centers are first updated using Equations (16) and (17), respectively, followed by the selection of the regularization parameter and the feature weight update using Equation (18). This regularized update shrinks less informative feature weights toward zero. Thus, in both methods, sparsity is enforced directly within the alternating optimization loop through the feature weight update, rather than being applied as a separate post-processing step. The updated weights are then used in the subsequent iteration until the convergence criterion is satisfied. To evaluate how well the proposed S-KM1 and S-KM2 algorithms perform, the following section contains some experimental results and comparisons.

4. Convergence Theorems for the S-KM2 Algorithm

The two proposed sparse k-means methods, S-KM1 and S-KM2, are discussed in Section 3. Based on the objective functions and their updating equations, we find that S-KM2 is much simpler than S-KM1. In S-KM1, determining the appropriate setting of the tuning parameter s is difficult, even when using the gap statistic. In this sense, we only consider convergence behaviors for the S-KM2 algorithm. We know that the S-KM2 algorithm determines the relevant features by assigning non-zero weights for key features and zero weights to features that are unimportant by using Lasso regularization as a penalty term. In this section, we formulate a convergence theorem for the S-KM2 algorithm in the following Theorem 5, which ensures a convergent subsequence in S-KM2 will ultimately lead to an optimal solution. First, we need to use Zangwill's convergence theorem [28] and the bordered Hessian matrix [29], as they offer a useful method for studying the convergence behaviors of clustering algorithms. An iterative algorithm was designed by Zangwill [28] to solve the problem as any closed point-to-set mapping $T : G \rightarrow Q(G)$, in which $Q(G)$ is the power set of G . We mention that the S-KM2 algorithm, which minimizes the closedness constraints T by ordinary continuity, is a point-to-point map. Therefore, Zangwill's convergence theorem is restated as follows.

Zangwill's convergence theorem [28]: Let $T : G \rightarrow G$ be the point-to-point map that produces a sequence $\{g_k\}_{k=0}^{\infty}$ by $g_{k+1} = T(g_k)$. Let the solution set be $\hat{G} \subset G$ and assume that:

1. The function $S : G \rightarrow R$ is continuous such that if $g \notin \hat{G}$, $S(T(g)) < S(g)$, and if $t \in \hat{G}$, $S(T(t)) \leq S(t)$.
2. For $G / \hat{G}$, the map S is continuous.

3. A compact set $\Phi \subseteq G$ contains all of the points g_k .

Following that, any convergent subsequence of $[g_k]_{k=0}^{\infty}$ that has a limit in the solution set $\hat{G}$ and $S(g_k)$ will eventually converge monotonically to $S(g)$ for some $g \in \hat{G}$. Let $N_{km} = \left\{ U = [u_{ik}]_{n \times c} \mid \forall k, i, u_{ik} \in \{0, 1\}, \sum_{k=1}^c u_{ik} = 1 \right\}$ and $U = [u_{ik}]_{n \times c} \in N_{km}$ where μ_{ik} is the membership of i^{th} point for the k^{th} cluster. Suppose the set $V = \{v_1, \dots, v_c\}$ be the c cluster centers and let $N_W = \left\{ W = [w_j]_{1 \times d} \mid \sum_{j=1}^d w_j = 1, w_j \in (0, 1), \forall j \right\}$, where w_j is the j^{th} feature weight. Let $\hat{G}_{S-KM2}$ be an algorithmic solution set for the S-KM2 as in

$$\hat{G}_{S-KM2} = \left\{ (\hat{U}, \hat{V}, \hat{W}) \mid \begin{array}{l} \forall V \neq \hat{V}, J_{S-KM2}(\hat{U}, \hat{V}, \hat{W}) < J_{S-KM2}(\hat{U}, V, \hat{W}); \\ \forall U \in N_{km}, U \neq \hat{U}, J_{S-KM2}(\hat{U}, \hat{V}, \hat{W}) < J_{S-KM2}(U, \hat{V}, \hat{W}); \\ \forall W \in N_W, W \neq \hat{W}, J_{S-KM2}(\hat{U}, \hat{V}, \hat{W}) < J_{S-KM2}(\hat{U}, \hat{V}, W) \end{array} \right\} \text{ with } \hat{U} = [\hat{u}_{ik}]_{n \times c},$$

$$\hat{\mu}_{ik} = \begin{cases} 1, & \text{if } \sum_{k=1}^c \sum_{j=1}^d w_j (x_{ij} - v_{kj})^2 = \min_{1 \leq l \leq c} \sum_{k=1}^c \sum_{j=1}^d w_j (x_{ij} - v_{lj})^2 \\ 0, & \text{otherwise} \end{cases}, \quad \hat{V} = (\hat{v}_1, \dots, \hat{v}_c), \quad \hat{v}_{kj} = \frac{\sum_{i=1}^n (u_{ik}) x_{ij}}{\sum_{i=1}^n (u_{ik})},$$

$$w_j = \begin{cases} \frac{-\sum_{i=1}^n \sum_{k=1}^c u_{ik} (x_{ij} - v_{kj})^2 - \lambda}{\left\| -\sum_{i=1}^n \sum_{k=1}^c u_{ik} (x_{ij} - v_{kj})^2 - \lambda \right\|_2}, & \text{when } \theta < -\lambda \\ 0, & \text{when } |\theta| \leq \lambda, \text{ where } \theta = \sum_{i=1}^n \sum_{k=1}^c \mu_{ik} (x_{ij} - v_{kj})^2 \\ \frac{-\sum_{i=1}^n \sum_{k=1}^c u_{ik} (x_{ij} - v_{kj})^2 + \lambda}{\left\| -\sum_{i=1}^n \sum_{k=1}^c u_{ik} (x_{ij} - v_{kj})^2 + \lambda \right\|_2}, & \text{when } \theta > \lambda \end{cases}.$$

Let $O: N_{km} \rightarrow [\mathbb{R}^d]^c$ and $O(U) = V = (v_1, \dots, v_c)$ where v_{kj} is determined by using Equation (17), $Q: [\mathbb{R}^d]^c \times N_W \rightarrow N_{km}$ with $Q(V, W) = U = [u_{ik}]_{n \times c}$ obtained by using Equation (16), and $R: [\mathbb{R}^d]^c \times N_{km} \rightarrow N_W$ with $R(U, V) = W = [w_j]_{1 \times d}$ determined by using Equation (18). Thus, we define the S-KM2 operator T in the following Definition 1.

Definition 1. The S-KM2 operator $T: N_{km} \times [\mathbb{R}^d]^c \times N_W \rightarrow N_{km} \times [\mathbb{R}^d]^c \times N_W$ is defined as $T = A_2 \circ A_1$ where $A_1: N_{km} \times [\mathbb{R}^d]^c \times N_W \rightarrow N_{km}$ with $A_1(U, V, W) = Q(V, W)$ and $A_2: N_{km} \rightarrow N_{km} \times [\mathbb{R}^d]^c \times N_W$ with $A_2(U) = (U, O(U), R(U, O(U)))$. Hence, we have $T(U, V, W) = (A_2 \circ A_1)(U, V, W) = A_2(A_1(U, V, W)) = A_2(Q(V, W)) = (Q(V, W), O(Q(V, W)), R(Q(V, W), O(Q(V, W)))) = (\hat{U}, \hat{V}, \hat{W})$.

Analyzing the Jacobian and Hessian matrices is a sufficient requirement for a strict minimizer of an objective function. In contrast, the Lagrange multipliers and a bordered Hessian matrix need to be evaluated as follows if specific constraints are considered.

Theorem 3 (the Lagrange theorem [29]). Let functions $f: E_f \rightarrow \mathbb{R}$, $E_f \subseteq \mathbb{R}^n$, and $z_i: E_{z_i} \rightarrow \mathbb{R}$, $E_{z_i} \subseteq \mathbb{R}^n$, $i = 1, \dots, t$, $t < n$, is continuous and differentiable partially. Let $x^{(0)} = \{x_1^{(0)}, \dots, x_n^{(0)}\} \in E_f$ be a local extreme value of a function f having the constraints

$z_i(x_1, \dots, x_n) = 0, \quad i = 1, \dots, t$. Let $L(x; \vartheta) = f(x_1, \dots, x_n) + \sum_{i=1}^t \vartheta_i z_i(x_1, \dots, x_n)$ and the

Jacobian $|J| = \begin{vmatrix} \frac{\partial z_1(x)}{\partial x_i} & \dots & \frac{\partial z_1(x)}{\partial x_i} \\ \vdots & \vdots & \vdots \\ \frac{\partial z_t(x)}{\partial x_i} & \dots & \frac{\partial z_t(x)}{\partial x_i} \end{vmatrix} \neq 0$ for $x^{(0)}$. Then, the gradient of $L(x; \vartheta)$ at $(x^{(0)}; \vartheta^{(0)})$ is 0. That is $\nabla L(x^{(0)}; \vartheta^{(0)}) = 0$.

Theorem 4 (local sufficient conditions [29]). Let $f: E_f \rightarrow \mathbb{R}$, $E_f \subseteq \mathbb{R}^n$, and $z_i: E_{z_i} \rightarrow \mathbb{R}$, $E_{z_i} \subseteq \mathbb{R}^n$, $i = 1, \dots, t$, $t < n$, be functions that are twice continuously partially differentiable and let $(x^{(0)}; \vartheta^{(0)})$ with $x^{(0)} \in E_f$ be a solution of $\nabla L(x^{(0)}; \vartheta^{(0)}) = 0$. Let

$$H_L(x; \vartheta) = \begin{pmatrix} 0 & \dots & 0 & \frac{\partial^2 L}{\partial \vartheta_1 \partial x_1} & \dots & \frac{\partial^2 L}{\partial \vartheta_1 \partial x_n} \\ \vdots & & \vdots & \vdots & & \vdots \\ 0 & \dots & 0 & \frac{\partial^2 L}{\partial \vartheta_1 \partial x_1} & \dots & \frac{\partial^2 L}{\partial \vartheta_1 \partial x_n} \\ \frac{\partial^2 L}{\partial x_1 \partial \vartheta_1} & \dots & \frac{\partial^2 L}{\partial x_1 \partial \vartheta_1} & \frac{\partial^2 L}{\partial x_1 \partial x_1} & \dots & \frac{\partial^2 L}{\partial x_1 \partial x_n} \\ \vdots & & \vdots & \vdots & & \vdots \\ \frac{\partial^2 L}{\partial x_n \partial \vartheta_1} & \dots & \frac{\partial^2 L}{\partial x_n \partial \vartheta_1} & \frac{\partial^2 L}{\partial x_n \partial x_1} & \dots & \frac{\partial^2 L}{\partial x_n \partial x_n} \end{pmatrix}$$

be the bordered Hessian with its leading principal minors $|H_t(x^{(0)}, \vartheta^{(0)})|$ of order $r = 2t + 1$, $2t + 2, \dots, n + t$ at $(x^{(0)}; \vartheta^{(0)})$. Thus, the following expressions can be obtained:

1. If all leading principal minors $|H_t(x^{(0)}; \vartheta^{(0)})|$, $2t + 1 \leq r \leq n + 1$, are with sign $(-1)^t$, then $x^{(0)} = \{x_1^{(0)}, \dots, x_n^{(0)}\}$ is a local minimum point of the function f for constraints $z_i(x) = 0$, $i = 1, \dots, t$.
2. If all leading principal minors $|H_t(x^{(0)}; \vartheta^{(0)})|$, $2t + 1 \leq r \leq n + 1$, alternate in signs, i.e., the signs of $|H_{n+t}(x^{(0)}; \vartheta^{(0)})| = |\overline{H}_L(x^{(0)}; \vartheta^{(0)})|$ are $(-1)^n$, then $x^{(0)} = (x_1^{(0)}, \dots, x_n^{(0)})$ is a local minima of the function f for constraints $z_i(x) = 0$, $i = 1, \dots, t$.
3. If neither condition of (1) nor (2) is met, then $x^{(0)}$ is not a local extreme point of the function f for constraints $z_i(x) = 0$, $i = 1, \dots, t$. In this instance, it is not deemed a violation of condition (1), when one or more leading principal minors have zero value.

Lemma 1. When the factors $U = \widehat{U}$ and $W = \widehat{W}$ are fixed, the objective function $J_{S-KM2}(\widehat{U}, V, \widehat{W})$ is minimized at $\widehat{v}_{kj}$ if and only if $\forall k, j, \widehat{v}_{kj} = \sum_{i=1}^n (\mu_{ik})^m x_{ij} / \sum_{i=1}^n (\mu_{ik})^m$.

Proof. We had already proved the necessary condition (“only if”) in Theorem 2. We take the first and second derivatives of $J_{S-KM2}(\hat{U}, V, \hat{W})$ w.r.t. v_{kj} for proving the “if” condition with fixing $U = \hat{U}$ and $W = \hat{W}$. We differentiate $J_{S-KM2}(\hat{U}, V, \hat{W})$ w.r.t. v_{kj} , therefore, we obtain the equations $\frac{\partial J_{S-KM2}}{\partial v_{kj}} = -2 \sum_{j=1}^d u_{ik} w_j (x_{ij} - v_{kj})$ and $\frac{\partial^2 J_{S-KM2}}{\partial v_{kj} \partial v_{tj}} = 2 \delta_{kt} I_d \sum_{j=1}^d u_{ik} w_j$ in addition $\delta_{kt} = \begin{cases} 1, & \text{if } k = t \\ 0, & \text{if } k \neq t \end{cases}$, where δ_{kt} is the kronecker index. The derivative of $J_{S-KM2}(\hat{U}, V, \hat{W})$ w.r.t. v_{kj} in term of Hessian matrix is $2 \cdot \text{diag} \left\{ I_d \sum_{j=1}^d u_{1i} w_j, \dots, I_d \sum_{j=1}^d u_{ci} w_j \right\}$. We can see the Hessian matrix is positive definite, and so the objective function $J_{S-KM2}(\hat{U}, V, \hat{W})$ is minimized at $\hat{v}_{kj}$ with $\hat{v}_{kj} = \sum_{i=1}^n (\mu_{ik})^m x_{ij} / \sum_{i=1}^n (\mu_{ik})^m$. $\square$

Lemma 2. The objective function $J_{S-KM2}(\hat{U}, \hat{V}, \hat{W})$ with $\sum_{j=1}^d w_j = 1$ is minimized at $\hat{W} = [\hat{w}_j]_{1 \times d}$ when the factors $V = \hat{V}$ and $U = \hat{U}$ are fixed if and only if

$$\hat{w}_j = \begin{cases} \frac{-\sum_{i=1}^n \sum_{k=1}^c u_{ik} (x_{ij} - v_{kj})^2 - \lambda}{\left\| -\sum_{i=1}^n \sum_{k=1}^c u_{ik} (x_{ij} - v_{kj})^2 - \lambda \right\|_2}, & \text{when } \theta < -\lambda \\ 0, & \text{when } |\theta| \leq \lambda, \text{ where } \theta = \sum_{i=1}^n \sum_{k=1}^c \mu_{ik} (x_{ij} - v_{kj})^2 \\ \frac{-\sum_{i=1}^n \sum_{k=1}^c u_{ik} (x_{ij} - v_{kj})^2 + \lambda}{\left\| -\sum_{i=1}^n \sum_{k=1}^c u_{ik} (x_{ij} - v_{kj})^2 + \lambda \right\|_2}, & \text{when } \theta > \lambda \end{cases}$$

Proof. The necessary condition “only if” had been proved already in Theorem 2. Consider the Lagrangian $J_{S-KM2}^*(U, V, W) = \sum_{i=1}^n \sum_{k=1}^c \sum_{j=1}^d w_j u_{ik} (x_{ij} - v_{kj})^2 + \lambda \sum_{j=1}^d |w_j| + \phi [\|w\|_2^2 - 1]$ of

$J_{S-KM2}(U, V, W)$, where $\|w\|_1 = \sum_{j=1}^d |w_j|$. Differentiate the Lagrangian $J_{S-KM2}^*(U, V, W)$

w.r.t. w_j to prove condition “if” for the fixed $V = \hat{V}$ and $U = \hat{U}$, then we obtain

$$\frac{\partial \hat{J}}{\partial w_j} = \frac{\partial}{\partial w_j} \left\{ \sum_{i=1}^n \sum_{k=1}^c \sum_{j=1}^d w_j u_{ik} (x_{ij} - v_{kj})^2 + \phi (\|w\|_2^2 - 1) \right\} + \frac{\partial}{\partial w_j} \lambda \|w\|_1.$$

The solution for the second partial derivative of $J_{S-KM2}^*(U, V, W)$ is by using the concept of sub differentials.

Consider the first partial derivative of $\partial \hat{J} / \partial w_j$ as following:

$$\frac{\partial \hat{J}}{\partial w_j} = Z^* = \sum_{i=1}^n \sum_{k=1}^c \sum_{j=1}^d u_{ik} (x_{ij} - v_{kj})^2 + 2\phi w_j.$$

$$\frac{\partial^2 \hat{J}}{\partial^2 w_j} = \frac{\partial}{\partial w_j} \left(\frac{\partial \hat{J}}{\partial w_j} \right) = \frac{\partial}{\partial w_j} \left[\sum_{i=1}^n \sum_{k=1}^c \sum_{j=1}^d u_{ik} (x_{ij} - v_{kj})^2 + 2\phi w_j \right], \quad \text{and} \quad \text{then}$$

$$\frac{\partial^2 \hat{J}}{\partial^2 w_j} = 0 + 2\phi = 2\phi. \text{ According to Theorem 1, we have } 2\phi = \left\| -\sum_{i=1}^n \sum_{k=1}^c \sum_{j=1}^d u_{ik} (x_{ij} - v_{kj})^2 \right\|_2^2$$

$$\text{and } \frac{\partial \hat{J}}{\partial \phi} = \|w\|_2^2 - 1, \quad \frac{\partial^2 \hat{J}}{\partial \phi \partial w_j} = 2w_j. \text{ Considering } \frac{\partial}{\partial w_j} \lambda \|w\|_1 = \lambda \frac{\partial}{\partial w_j} \left(\sum_{j=1}^d |w_j| \right) = \lambda \text{sign}(w_j),$$

$$\lambda \text{sign}(w_j) = \begin{cases} \lambda & ; \text{ when } w_j < 0 \\ [-\lambda, \lambda] & ; \text{ when } w_j = 0 \\ \lambda & ; \text{ when } w_j > 0 \end{cases} \quad \text{with } \lambda > 0, \quad \text{we have that}$$

$$\frac{\partial \hat{J}_j}{\partial w_j} = Z^* + \begin{cases} -\lambda & ; \text{ when } w_j < 0 \\ [-\lambda, \lambda] & ; \text{ when } w_j = 0, \text{ where } Z^* = \sum_{i=1}^n \sum_{k=1}^c \sum_{j=1}^d u_{ik} (x_{ij} - v_{kj})^2 + 2\phi w_j. \\ \lambda & ; \text{ when } w_j > 0 \end{cases}$$

$$\text{Therefore, } \frac{\partial^2 \hat{J}_j}{\partial^2 w_j} = 2\phi. \text{ By using the proof of Theorem 1, we have}$$

$$\|2\phi\| = \left\| -\sum_{i=1}^n \sum_{k=1}^c u_{ik} (x_{ij} - v_{kj})^2 \right\|_2^2 \text{ as } \|2\phi\| = 2\phi \text{ so that } \frac{\partial^2 \hat{J}_j}{\partial^2 w_j} = \left\| -\sum_{i=1}^n \sum_{k=1}^c u_{ik} (x_{ij} - v_{kj})^2 \right\|_2^2.$$

Therefore, the definition of bordered Hessian matrix w.r.t. w_j and ϕ is as follows:

$$\hat{H}_{ij}(w, \phi) = \begin{pmatrix} 0 & \frac{\partial^2 J_j}{\partial \phi \partial w_1} & \frac{\partial^2 J_j}{\partial \phi \partial w_2} & \dots & \frac{\partial^2 J_j}{\partial \phi \partial w_d} \\ \frac{\partial^2 J_j}{\partial \phi \partial w_1} & \frac{\partial^2 J_j}{\partial w_1 \partial w_1} & 0 & \dots & 0 \\ \frac{\partial^2 J_j}{\partial \phi \partial w_2} & 0 & \ddots & \dots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 J_j}{\partial \phi \partial w_d} & 0 & 0 & \dots & \frac{\partial^2 J_j}{\partial w_d \partial w_d} \end{pmatrix}.$$

Therefore, for only one given constraint, $t = 1$, hence $(-1)^1 = -1 < 0$. Next, we examine each of the top major minors:

$$\begin{aligned} |\hat{H}_3(\hat{w}; \hat{\phi})| &= 2w_1 \begin{vmatrix} 0 & 2w_1 & 2w_2 \\ \left\| -\sum_{i=1}^n \sum_{k=1}^c u_{li} (x_{ij} - v_{lj})^2 \right\|_2^2 & 0 & 0 \\ 2w_2 & 0 & \left\| -\sum_{i=1}^n \sum_{k=1}^c u_{2i} (x_{ij} - v_{2j})^2 \right\|_2^2 \end{vmatrix} \Bigg|_{\substack{w=\hat{w}_t^* \\ \phi=\hat{\phi}}} \\ &= - \left(4w_1^2 \left\| -\sum_{i=1}^n \sum_{k=1}^c u_{2i} (x_{ij} - v_{2j})^2 \right\|_2^2 + 4w_2^2 \left\| -\sum_{i=1}^n \sum_{k=1}^c u_{1i} (x_{ij} - v_{1j})^2 \right\|_2^2 \right) \end{aligned}$$

$$\begin{aligned}
|\widehat{H}_4(\widehat{w}; \phi)| &= \begin{vmatrix} 0 & 2w_1 & 2w_2 & 2w_3 \\ 2w_1 & \left\| -\sum_{i=1}^n \sum_{k=1}^c u_{li}(x_{ij} - v_{1j})^2 \right\|_2^2 & 0 & 0 \\ 2w_2 & 0 & \left\| -\sum_{i=1}^n \sum_{k=1}^c u_{2i}(x_{ij} - v_{2j})^2 \right\|_2^2 & 0 \\ 2w_3 & 0 & 0 & \left\| -\sum_{i=1}^n \sum_{k=1}^c u_{3i}(x_{ij} - v_{3j})^2 \right\|_2^2 \end{vmatrix} \\
&= \begin{pmatrix} 2w_1 \left\| -\sum_{i=1}^n \sum_{k=1}^c u_{2i}(x_{ij} - v_{2j})^2 \right\|_2^2 \left\| -\sum_{i=1}^n \sum_{k=1}^c u_{3i}(x_{ij} - v_{3j})^2 \right\|_2^2 - 2w_2 \left\| -\sum_{i=1}^n \sum_{k=1}^c u_{li}(x_{ij} - v_{1j})^2 \right\|_2^2 \left\| -\sum_{i=1}^n \sum_{k=1}^c u_{3i}(x_{ij} - v_{3j})^2 \right\|_2^2 \\ -2w_1 \left\| -\sum_{i=1}^n \sum_{k=1}^c u_{li}(x_{ij} - v_{1j})^2 \right\|_2^2 \left\| -\sum_{i=1}^n \sum_{k=1}^c u_{2i}(x_{ij} - v_{2j})^2 \right\|_2^2 \end{pmatrix} < 0 \\
|\widehat{H}_{T-H}(\widehat{w}; \phi)| &= \begin{vmatrix} 0 & 2w_1 & 2w_2 & \cdots & 2w_d \\ 2w_1 & \left\| -\sum_{i=1}^n \sum_{k=1}^c u_{ik}(x_{ij} - v_{1j})^2 \right\|_2^2 & 0 & \cdots & 0 \\ 2w_2 & 0 & \left\| -\sum_{i=1}^n \sum_{k=1}^c u_{ik}(x_{ij} - v_{2j})^2 \right\|_2^2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 2w_d & 0 & 0 & \cdots & \left\| -\sum_{i=1}^n \sum_{k=1}^c u_{ik}(x_{ij} - v_{dj})^2 \right\|_2^2 \end{vmatrix} \\
&= -\sum_{j=1}^d \prod_{\substack{p=1 \\ p \neq j}}^d \left\| -\sum_{i=1}^n \sum_{k=1}^c u_{ik}(x_{ij} - v_{1j})^2 \right\|_2^2 < 0. \quad \text{Thus, the objective function } J_{S-KM2}(\widehat{U}, \widehat{V}, \widehat{W}) \\
&\quad \substack{w=\widehat{w}; \\ \phi=\widehat{\phi}}
\end{aligned}$$

subject to $\sum_{j=1}^d w_j = 1$ is minimized at $\widehat{W} = [\widehat{w}_j]_{1 \times d}$. $\square$

Lemma 3. The objective function $J_{S-KM2}(\widehat{U}, \widehat{V}, \widehat{W})$ is continuous on $[\mathbb{R}^d]^c \times N_{km} \times N_W$.

Proof. By considering the fact that $[u_{ik} \rightarrow u_{ik}]$, $[v_{kj} \rightarrow \|x_{ij} - v_{kj}\|^2]$ and $[w_j \rightarrow \|w_j\|]$ are continuous, the total sum of objects generated by $[u_{ik} \rightarrow u_{ik}]$ and $[v_{kj} \rightarrow \|x_{ij} - v_{kj}\|^2]$ is also continuous, and the total sum of objects generated by $[u_{ik} \rightarrow u_{ik}]$ and $[w_j \rightarrow \|w_j\|]$ should be continuous too. Therefore, the objective function $J_{S-KM2}(\widehat{U}, \widehat{V}, \widehat{W})$ is continuous on $[\mathbb{R}^d]^c \times N_{km} \times N_W$. $\square$

Lemma 4. Let $\widehat{G}_{S-KM2}$ denote the set of solutions for the S-KM2 objective function $J_{S-KM2}(U, V, W)$. We can obtain that $J_{S-KM2}(Z(U, V, W)) = J_{S-KM2}(\widehat{U}, \widehat{V}, \widehat{W}) \leq J_{S-KM2}(U, V, W)$ for any $(U, V, W) \notin \widehat{G}_{S-KM2}$.

Proof. Let $(U, V, W) \notin \widehat{G}_{S-KM2}$. Then, we have that $J_{S-KM2}(Z(U, V, W)) = J_{S-KM2}(A_2 \circ A_1(U, V, W)) = J_{S-KM2}(\widehat{U}, \widehat{V}, \widehat{W})$

$$= \min_{u_{ik}} \left\{ \sum_{i=1}^n \sum_{k=1}^c \sum_{j=1}^d u_{ik} \widehat{w}_j (x_{ij} - \widehat{v}_{kj})^2 + \lambda \|\widehat{w}\|_1 \right\} \leq \sum_{i=1}^n \sum_{k=1}^c \sum_{j=1}^d u_{ik} \widehat{w}_j (x_{ij} - \widehat{v}_{kj})^2 + \lambda \|\widehat{w}\|_1$$

$= J_{S-KM2}(U, \widehat{V}, \widehat{W}) \leq J_{S-KM2}(U, \widehat{V}, W)$ (by Lemma 2) $\leq J_{S-KM2}(U, V, W)$ (by Lemma 1).
 Therefore, we obtain that $J_{S-KM2}(Z(U, V, W)) = J_{S-KM2}(\widehat{U}, \widehat{V}, \widehat{W}) \leq J_{S-KM2}(U, V, W)$
 for any $(U, V, W) \notin \widehat{G}_{S-KM2}$. $\square$

Lemma 5. The S-KM2 operator T for the objective function $J_{S-KM2}(U, V, W)$ operates continuously on $[\mathbb{R}^d]^c \times N_{km} \times N_W$.

Proof. Let $A_1(U, V, W) = Q(V, W) = \{Q_{11}(V, W), \dots, Q_{cn}(V, W)\}$. Since

$$\left[v_{kj} \rightarrow \|x_{ij} - v_{kj}\|^2 \right], \quad \left[w_j \rightarrow |w_j| \right] \quad \text{and}$$

$$\left\{ \sum_{k=1}^c \sum_{j=1}^d w_j (x_{ij} - v_{kj})^2 \rightarrow \min_{1 \leq l \leq c} \sum_{k=1}^c \sum_{j=1}^d w_j (x_{ij} - v_{lj})^2 \right\} \quad \text{are} \quad \text{continuous,}$$

$$Q_{ik}(V, W) = \begin{cases} 1, & \text{if } \sum_{k=1}^c \sum_{j=1}^d w_j (x_{ij} - v_{kj})^2 = \min_{1 \leq l \leq c} \sum_{k=1}^c \sum_{j=1}^d w_j (x_{ij} - v_{lj})^2 \\ 0, & \text{otherwise} \end{cases} = \mu \quad \text{is also}$$

continuous. Because the sum and product of continuous functions are also continuous.

Thus, $Q(V, W) = \{Q_{11}(V, W), \dots, Q_{cn}(V, W)\}$ is continuous too. Furthermore,

$$A_2(U) = \{U, O(U), R(U, O(U))\} \quad \text{where} \quad O(U) = \{O_{11}(U), \dots, O_{cd}(U)\}, \quad \text{and}$$

$$R(U, V) = \{R_1(U, V), \dots, R_d(U, V)\} \quad \text{with} \quad v_{kj} = O_{kj}(U) = \frac{\sum_{i=1}^n u_{ik} x_{ij}^h}{\sum_{i=1}^n u_{ik}} \quad \text{and}$$

$$R_j(U, V) = \begin{cases} \frac{-\sum_{i=1}^n \sum_{k=1}^c u_{ik} (x_{ij} - v_{kj})^2 - \lambda}{\left\| -\sum_{i=1}^n \sum_{k=1}^c u_{ik} (x_{ij} - v_{kj})^2 - \lambda \right\|_2}, & \text{when } \theta < -\lambda \\ 0, & \text{when } |\theta| \leq \lambda, \text{ where } \theta = \sum_{i=1}^n \sum_{k=1}^c \mu_{ik} (x_{ij} - v_{kj})^2 \\ \frac{-\sum_{i=1}^n \sum_{k=1}^c u_{ik} (x_{ij} - v_{kj})^2 + \lambda}{\left\| -\sum_{i=1}^n \sum_{k=1}^c u_{ik} (x_{ij} - v_{kj})^2 + \lambda \right\|_2}, & \text{when } \theta > \lambda \end{cases}.$$

Since $[u_{ik} \rightarrow u_{ik}^m]$ and $[u_{ik} \rightarrow u_{ik} x_{ij}]$ are continuous, $O_{kj}(U)$ and $R_j(U, V)$ with the quotient of two continuous functions are also continuous. Therefore, $O(U) = \{O_{11}(U), \dots, O_{cd}(U)\}$, and $R(U, V) = \{R_1(U, V), \dots, R_d(U, V)\}$ are continuous. Thus, $T = A_2 \circ A_1$ is continuous on $[\mathbb{R}^d]^c \times N_{km} \times N_W$. $\square$

Lemma 6. Let $(\text{conv}(X))^c$ be a c -fold Cartesian product of the convex hull X . Let $\{Q(V^{(0)}, W^{(0)}), V^{(0)}, W^{(0)}\}$ be the starting points of iterations with T . Then, $\{T^{(r)}(Q(V^{(0)}, W^{(0)}), V^{(0)}, W^{(0)})\} \in N_{km} \times (\text{conv}(X))^c \times N_W$ is compact in $N_{km} \times (\mathbb{R}^d)^c \times N_W$.

Proof. Let $\{Q(V^{(0)}, W^{(0)}), V^{(0)}, W^{(0)}\}$ be the starting points of iterations with T in which $Q(V^{(0)}, W^{(0)}) = \{Q_{11}(V^{(0)}, W^{(0)}), \dots, Q_{cn}(V^{(0)}, W^{(0)})\}$ with $u_{ik}^{(0)} = Q_{ik}(V^{(0)}, W^{(0)})$ can be describe as $Q_{ik}(V^{(0)}, W^{(0)}) = \begin{cases} 1, & \text{if } \sum_{k=1}^c \sum_{j=1}^{d_h} w_j^{(0)} (x_{ij} - v_{kj}^{(0)})^2 = \min_{1 \leq l \leq c} \sum_{k=1}^c \sum_{j=1}^{d_h} w_j^{(0)} (x_{ij} - v_{lj}^{(0)})^2 \\ 0, & \text{otherwise} \end{cases}$. Moving

forward in a recursive way, we have that $V^{(r)} \in (\text{conv}(X))^c, \forall r \geq 1$. It is clearly that

$$u_{ik}^{(1)} = Q_{ik}(V^{(1)}, W^{h(0)}) = \begin{cases} 1, & \text{if } \sum_{k=1}^c \sum_{j=1}^{d_h} w_j^{(0)} (x_{ij} - v_{kj}^{(1)})^2 = \min_{1 \leq l \leq c} \sum_{k=1}^c \sum_{j=1}^{d_h} w_j^{(0)} (x_{ij} - v_{lj}^{(1)})^2 \\ 0, & \text{otherwise} \end{cases} \in N_{km}$$

and $U^{(1)} \in N_{km}$. Therefore,

$$w_j^{(1)} = R_j(U^{(1)}, V^{(1)}) = \sum_{j'=1}^{d_h} \left\{ \frac{\sum_{h=1}^H (v_h^{(1)})^2 \sum_{k=1}^c \sum_{i=1}^n (u_{in}^{(1)}) (x_{ij}^h - z_{kj}^{h(1)})^2 + \lambda}{\sum_{h=1}^H (v_h^{(1)})^2 \sum_{k=1}^c \sum_{i=1}^n (u_{ik}^{(1)}) (x_{ij'}^h - z_{kj'}^{h(1)})^2 + \lambda} \right\} \in N_{W^h}$$

with

$$w_j^{(1)} = R_j(U^{(1)}, V^{(1)}) = \begin{cases} \frac{-\sum_{i=1}^n \sum_{k=1}^c (u_{ik}^{(1)}) (x_{ij} - v_{kj}^{(1)})^2 - \lambda}{\left\| -\sum_{i=1}^n \sum_{k=1}^c (u_{ik}^{(1)}) (x_{ij} - v_{kj}^{(1)})^2 - \lambda \right\|_2}, & \text{when } \theta < -\lambda \\ 0, & \text{when } |\theta| \leq \lambda, \text{ where } \theta = \sum_{i=1}^n \sum_{k=1}^c \mu_{ik} (x_{ij} - v_{kj}^{(1)})^2 = w_j^{(1)} \\ \frac{-\sum_{i=1}^n \sum_{k=1}^c (u_{ik}^{(1)}) (x_{ij} - v_{kj}^{(1)})^2 + \lambda}{\left\| -\sum_{i=1}^n \sum_{k=1}^c (u_{ik}^{(1)}) (x_{ij} - v_{kj}^{(1)})^2 + \lambda \right\|_2}, & \text{when } \theta > \lambda \end{cases}$$

and $W^{(1)} \in N_W$. Furthermore, moving in a recursive approach, $U^{(r)} \in N_{km}$ and $W^{(r)} \in N_W, \forall r \geq 1$. Thus, $T^{(r)}\{Q(V^{(0)}, W^{(0)}), V^{(0)}, W^{(0)}\} \in N_{km} \times (\text{conv}(X))^c \times N_W$ is compact in $N_{km} \times (\mathbb{R}^d)^c \times N_W$. $\square$

By assessing the conditions of Zangwill's convergence theorem according to Lemmas 3–6, we obtain the following convergence theorem, Theorem 5, for the S-KM2 algorithm.

Theorem 5. Let the data set $X = \{x_1, \dots, x_n\}$ be bounded in $\mathbb{R}^d$ with the objective function $J_{S-KM2}(U, V, W)$. Let $T: N_{km} \times (\mathbb{R}^d)^c \times N_W \rightarrow N_{km} \times (\mathbb{R}^d)^c \times N_W$ be the operator of S-KM2 as defined in Definition 1. Then, any convergent subsequence $\{T^{(r)}(Q(V^{(0)}, W^{(0)}), V^{(0)}, W^{(0)})\}$ from the S-KM2 algorithm shall tend to the optimal solution $(\hat{U}, \hat{V}, \hat{W})$ in $\hat{G}_{S-KM2}$ and the S-PCM2 sequence $\{T^{(r)}(Q(V^{(0)}, W^{(0)}), V^{(0)}, W^{(0)})\}$ will monotonically converge to the optimal solution $(\hat{U}, \hat{V}, \hat{W})$ in $\hat{G}_{S-KM2}$.

5. Clustering Measures, Results, and Comparisons

In this section, we demonstrate the capability and strength of the proposed S-KM1 and S-KM2 algorithms by undertaking several comprehensive experiments on synthetic and real datasets. We use four synthetic datasets and nine real datasets to perform the

experiments. The performances of the clustering results are assessed by making comparisons between the true labels (i.e., the labels of each data point) and the predicted labels (i.e., the labels assigned by the algorithm). The performance evaluation measures are the accuracy rate (AR), Jaccard index (JI) [30], Rand index (RI) [31], normalized mutual information (NMI) [32], Fowlkes–Mallows index (FMI) [33] and running time (RT). Let C and C' be the set of k clusters with $C = \{C_1, C_2, \dots, C_c\}$ in the original dataset and $C' = \{C'_1, C'_2, \dots, C'_c\}$ from the algorithm. The formal description of these measures are as follows.

Accuracy Rate (AR): The accuracy rate (AR) is measured by $AR = \sum_{i=1}^c h_i / n$, where, h_i denotes the correctly clustered data points in C' and n is the number of data points. The accuracy rate (AR) measures the percentage of clustering results with correctly identified points.

Jaccard Index (JI) [30]: To determine which members of two sets are similar and which are dissimilar, the members for the two sets are evaluated by applying the Jaccard index (JI) [30]. To compute the JI, we use the equation $JI = J(A, B) = |A \cap B| / |A \cup B|$ where A appears in both sets, and B can be seen in either set.

Rand Index (RI) [31]: The Rand index (RI) [31] evaluates how similar the cluster assignments are to one another by making pairwise comparisons. The RI always has a value between 0 and 1. A higher score indicates greater similarity. Let (X_i, X_j) symbolize a given pair of points from the dataset. If two points are paired together and both of them belong to the same cluster in C and C' , then there are a number of those pairs; likewise, let b denote the number of point pairs in which both points belong to two different clusters C and two different clusters in C' . If two points are in the same cluster in C but distinct clusters in C' , then there are c pairs of points, and d is the number of pairs of points in which both points are members of the same cluster in C' and two different clusters in C . The overall number $M = n(n-1)/2$ of possible point pairs within the dataset and the number n of data points are used to determine the RI, where $RI = (a + d) / M$. Similarly, the Rand Index measure can be defined as $RI = \text{No. of pairwise predictions} / \text{total No. of possible pairs}$. The percentage of accurate decisions the algorithm makes can be estimated by the Rand index. It is additionally determined by $RI = (TP + TN) / (TP + FP + FN + TN)$.

Normalized Mutual Information (NMI) [32]: Normalized mutual information (NMI) evaluates how well the cluster assignments agree with one another. Higher similarity is represented by a higher score. The joint and marginal probabilities are used to measure the level of agreement between clusters. The volume of data about a term's existence or lack thereof is determined by the NMI [32], which is necessary to determine its appropriate classification. To determine NMI, $NMI = I(X : Y) / ((H(X) + H(Y)) / 2)$ can be used, where $H(X)$ and $H(Y)$ are the marginal entropies of X and Y , respectively, and $I(X : Y)$ is a piece of mutual information between $H(X)$ and $H(Y)$.

Fowlkes–Mallows Index (FMI) [33]: The true positive, false positive, and false negative are used to generate the Fowlkes–Mallows Index (FMI) [33]. By counting the number of pairwise points that are assigned in the same cluster for the predicted and actual labels, or in a different cluster for TP, FP, and FN, it is possible to determine these terms. To compute FMI, we use $FMI = TP / \sqrt{(TP + FP)(TP + FN)}$. It is to be noted that the values above all indices measures lie between 0 and 1. Higher values exhibit that the similarity of clustering is high, and low values indicate that clusters are dissimilar.

Running Time (RT): Running time (RT) is the total average running time in seconds that an algorithm takes to reach convergence.

Based on the above clustering evaluation measures, we compare the effectiveness of our proposed S-KM1 and S-KM2 algorithms with the existing algorithms considered in this study. The comparisons in Yang and Benjamin [19] demonstrate that their proposed SPCM2 is better than the sparse k-means method proposed by Witten and Tibshirani [15] and the S-FCM algorithm proposed by Qiu et al. [18]. They chose SPCM2 over SPCM1 because of the savings made on run time. Thus, we should only compare the proposed S-KM1 and S-KM2 algorithms with the SPCM2 and (original) k-means. A detailed evaluation and comparisons between them are presented in the following section.

5.1. Evaluation and Comparison with Synthetic Datasets

In this section, we demonstrate a comprehensive investigation using four generated synthetic datasets to examine the performance of the aforementioned algorithms. Four synthetic experiments were conducted to comprehensively evaluate the behavior of the k-means, S-KM1, S-KM2, and SPCM2 algorithms in controlled clustering settings. The synthetic datageneration mechanism for the first three experiments follows the construction in Examples 1–3 and was adopted from the construction used by Yang and Benjamin [19], which was itself developed following the sparse clustering data generation framework of Qiu et al. [18]. In Example 1, three clusters were generated from three independent normal distributions, in which there were different means in the first five relevant features, while the remaining features were generated from a common zero-mean distribution and treated as irrelevant features. Examples 2 and 3 extend this setting by varying the number of features and observations to examine the effects of dimensionality and sample size, respectively. Example 4 uses a Gaussian mixture distribution to generate three clusters with $(n=450)$ observations and $(d=15)$ features, including three relevant and twelve irrelevant features. Each experiment was repeated 100 times with different initializations, and the average AR, JI, RI, NMI, FMI, and RT were used to compare the four clustering methods. We obtain both the ground truth label with its actual data labels and then made a comparison of each data point's ground truth label with its actual clustering label before assessing the results. We evaluated the effectiveness of algorithms by considering the six metrics of AR, JI [30], RI [31], NMI [32], FMI [33] and RT. The algorithms of k-means, S-KM1, S-KM2 and SPCM2 were repeatedly implemented with 100 simulations at different values of initials, and then we obtained the averages of AR, JI, RI, NMI, FMI and RT. These results will be shown in tables.

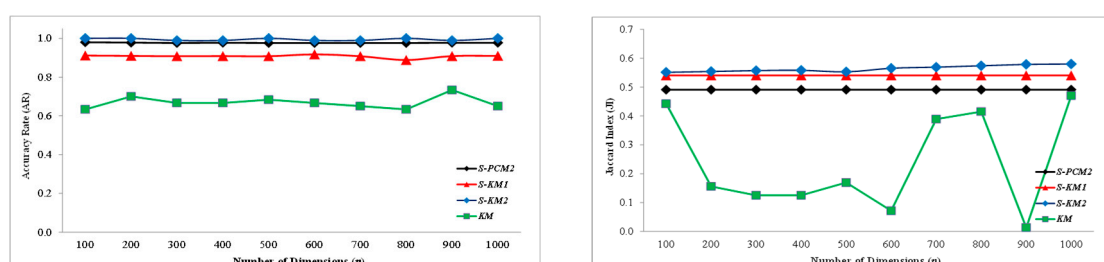
Example 1. The first dataset $X = \{x_1, \dots, x_n\}$ is generated from normal distributions with the n number of data points and d features, in which we use $X_{n \times d}$ to denote its data matrix. We follow a similar generating way to that of Yang and Benjamin [19]. The elements x_{ij} are independently distributed from three normal distributions in which these three normal distributions have different means with $-\mu$, 0 , and μ only in the initial q features. Specifically, data from the matrix $X_{n \times d}$ has the three clusters C_1, C_2 and C_3 , constructed based on the three normal distributions with means $-\mu$, 0 , and μ , respectively. Initially, they have q features, but they are produced by the identical normal distribution with a mean of 0 for the remaining $(d - q)$ attributes. Therefore, the three clusters C_1, C_2 and C_3 have different means $-\mu, 0$, and μ only in the first q ($q < d$) features. However, they have a mean of 0 for the remaining $(d - q)$ attributes which denote unimportant (sparse) features. In this dataset, we consider the number of data points to be $n = 60$ and the number of features to be $d = 25$. We set $\mu = 2.5$ and the first five features as relevant, and the

remaining ($d - q = 20$) attributes are insignificant. The value of the tuning parameter λ is obtained by the gap statistic using the technique of Tibshirani et al. [26] in S-KM1. The initial weights are attained by the expression $w_j = 1/\sqrt{d}$. We implement the k-means, S-KM1, S-KM2 and SPCM2 algorithms to demonstrate their performance and effectiveness and compare their results on clusters. Table 1 illustrates their clustering findings. The results in Table 1 are distinctive, among which the top-ranked values are shown in bold, while the values in second place are underlined. The results demonstrate that S-KM2 outperforms S-KM1 in AR, JI, RI, and FMI, but S-KM1 outperforms S-KM2 in NMI. S-KM2 outperforms k-means and SPCM2 in AR, JI, RI, NMI and FMI. For RT, k-means ran the fastest with the most saving time, and S-KM2 ran faster than S-KM1 and SPCM2. In general, S-KM1 took the most time, at 2.1859 sec. The proposed S-KM2 algorithm showed the best performance based on the average evaluation measures (AR, JI, RI and FMI) in Table 1, while the proposed S-KM1 algorithm had the second highest values in average evaluation measures (AR, RI) but the best NMI. The proposed S-KM2 also showed the minimum average total running time (RT), but due to the use of gap statistics, the proposed S-KM1 required some additional time (RT). Therefore, we conclude from Table 1 that our proposed S-KM2 clustering algorithm shows excellent performance.

Table 1. Comparison between k-means, S-KM1, S-KM2, and SPCM2 based on averages of AR, JI, RI, NMI, FMI, and RT for Example 1.

Evaluation Criteria	AR	JI	RI	NMI	FMI	RT
k-means	0.9500	0.2000	0.5556	0.3333	<u>0.9578</u>	0.0294
S-KM1	<u>0.9833</u>	0.4716	<u>0.9222</u>	0.9154	0.5433	2.1859
S-KM2	1.0000	0.9355	0.9778	<u>0.8791</u>	0.9667	<u>0.0427</u>
SPCM2	0.9670	<u>0.6240</u>	0.8100	0.7230	0.7730	0.0630

Example 2 (Continuation from Example 1 by varying dimensions d). We continue generating datasets using the method from Example 1. We generate the data matrix $X_{n \times d}$ for this experiment for the four clusters C_1, C_2, C_3 and C_4 from the four normal distributions with means $-\mu, 0, \mu$ and 2μ , respectively; however, they are produced by the same normal distribution as the first q attributes, with a mean of 0 for the remaining $(d - q)$ attributes. We illustrate the average $\mu = 2.5$ and generate datasets with feature dimension sizes from $d = 100$ to $d = 1000$. We then evaluate how well the k-means, S-KM1, S-KM2 and SPCM2 clustering algorithms perform under different feature dimensions from $d = 100$ to $d = 1000$. These comparisons of the four algorithms are shown in the form of line-segment graphs in Figure 2 for AR, JI, RI, NMI, FMI and RT. From Figure 2, we observe that the proposed S-KM2 algorithm performed significantly better than the k-means, S-KM1, and SPCM2 algorithms according to performance metrics AR, JI, RI, FMI, and NMI. On the other hand, we also find that the run time RT of S-KM2 is lower than that of S-KM1 and better than that of the k-means and SPCM2 algorithms. Generally, S-KM1 requires more time to finish the clustering process due to the gap statistic for the tuning parameter. To conclude, the proposed S-KM2 algorithm demonstrates better performance than the k-means, S-KM1, and SPCM2 algorithms.



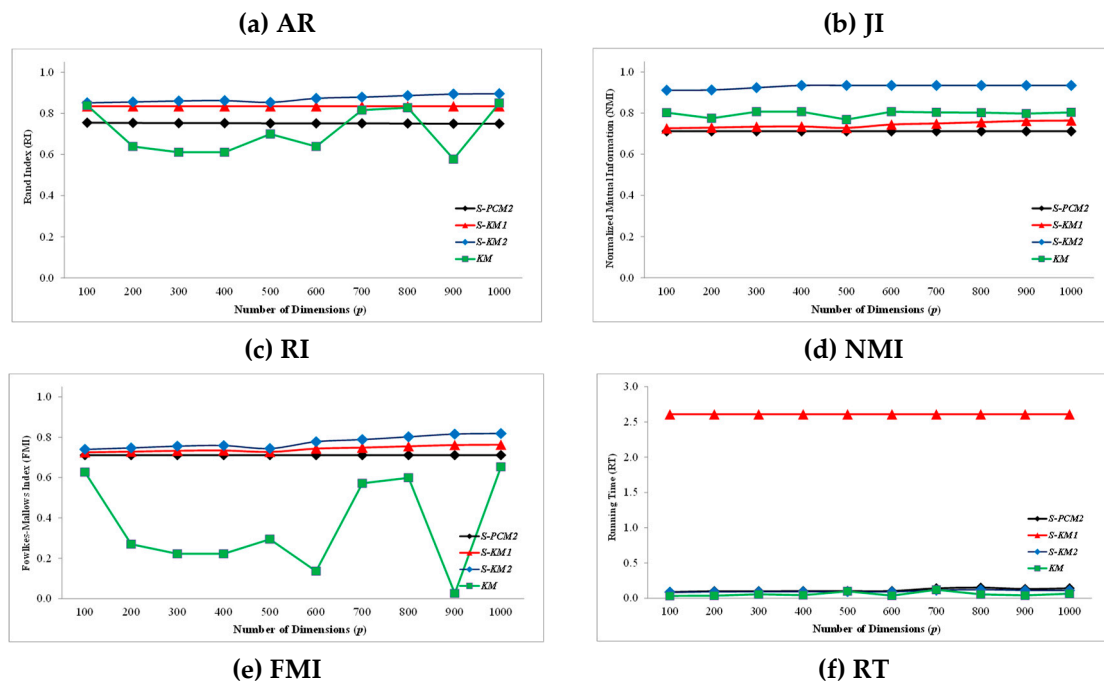
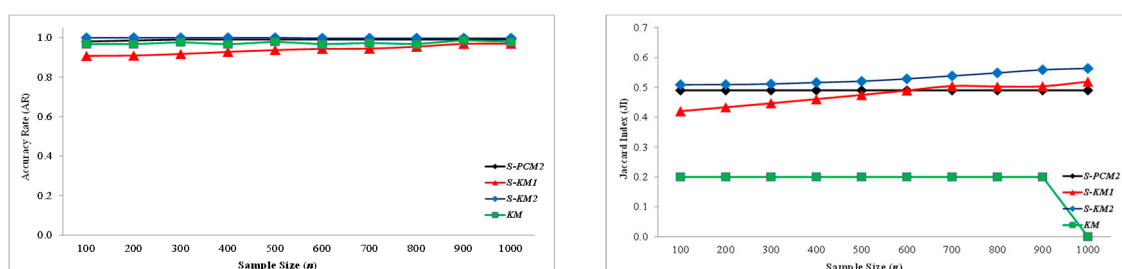


Figure 2. Comparisons of k-means, S-KM1, S-KM2 and SPCM2 in the form of line-segment graphs at different dimensions from $d = 100$ to $d = 1000$ on the averages of (a) AR, (b) JI, (c) RI, (d) NMI, (e) FMI, and (f) RT.

The apparently flat trajectories in Figure 2 reflect the small changes in the objective value as the iterative optimization approaches a stable solution. The trajectories therefore become visually indistinguishable after several iterations. The proposed algorithms continue their iterative updates until the specified stopping criterion is satisfied.

Example 3 (Continuation from Example 1 by varying the sample size n). We continue to generate datasets using the method outlined in Example 1. We generate the data matrix $X_{n \times d}$ for this experiment for the three clusters C_1, C_2 and C_3 , from the three normal distributions with means $-\mu, 0$, and μ , respectively; however, they have the same normal distribution as the initial q attributes, with a mean of 0 with respect to the remaining $(d - q)$ features. The mean is $\mu = 3.0$, and we generate datasets with data point sizes varying from $n = 100$ to $n = 1000$, with the fixed feature dimension $d = 1000$ and with $q = 25$. We then evaluate how well the k-means, S-KM1, S-KM2, and SPCM2 clustering algorithms perform with different data point sizes from $n = 100$ to $n = 1000$. We perform 100 simulations for these generated datasets on different initials. The line-segment graphs for AR, JI, RI, NMI, FMI and RT are shown in Figure 3. From Figure 3, we can see that the proposed S-KM2 algorithm performs better than the k-means, S-KM1, and SPCM2 algorithms based on the AR, JI, RI, FMI, and NMI metrics. S-KM2 has the lowest running time. We can conclude that the proposed S-KM2 algorithm performs better than the k-means, S-KM1, and SPCM2 algorithms.



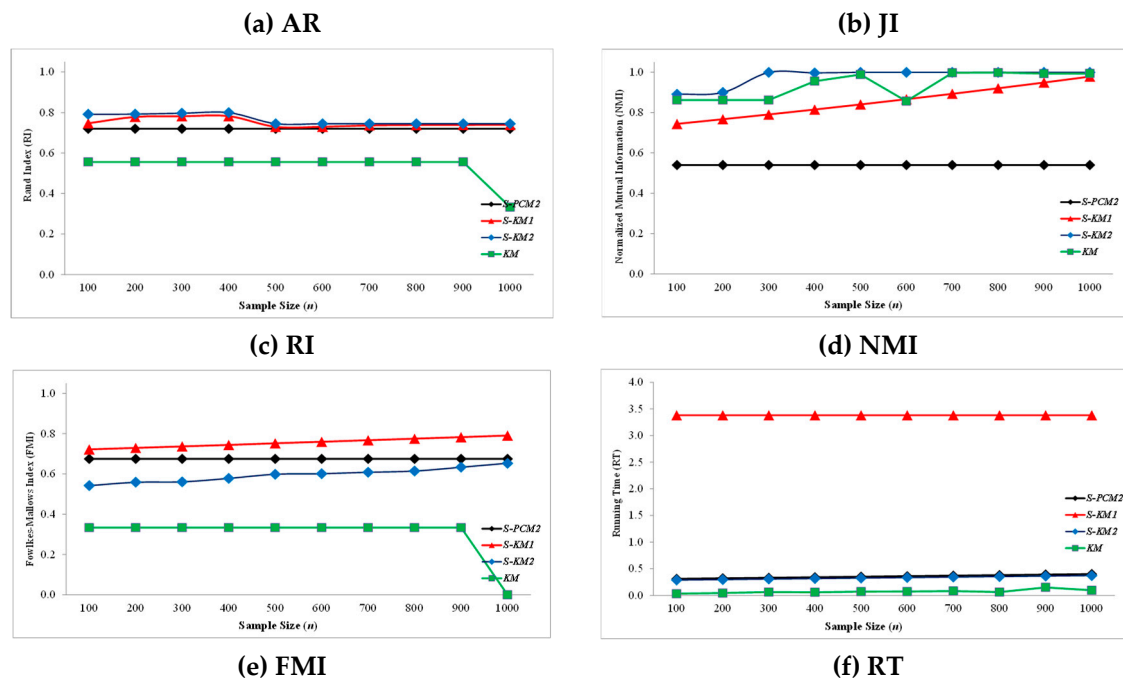


Figure 3. Comparisons of k-means, S-KM1, S-KM2, and SPCM2 in form of line-segment graphs at different sample sizes from $n = 100$ to $n = 1000$ on the averages of (a) AR, (b) JI, (c) RI, (d) NMI, (e) FMI, and (f) RT.

Example 4 (Generated from Gaussian mixture distribution). The distribution of the Gaussian mixture $\sum_{k=1}^3 \alpha_k N(\mu_k, \Sigma_k)$ is employed to construct the datasets in this experiment, in which the parameters $\alpha_k = 1/3, k = 1, 2, 3, \mu_1 = [3, 3, 3]^T, \mu_2 = [6, 6, 6]^T$ and $\mu_3 = [9, 9, 9]^T$ apply, with the covariance matrices $\Sigma_k = 0.30I, k = 1, 2, 3$, where I is an identity matrix. We generate a three-cluster dataset with C_1, C_2 and C_3 containing $n = 450$ data points and $d = 15$ features. There are three features out of a total of fifteen features that are considered relevant by the Gaussian mixture distribution. The uniform distribution with a mean of 0 generates the other twelve features. We implement the k-means, S-KM1, S-KM2 and SPCM2 clustering algorithms on this synthetic data. We perform 100 simulations for this dataset on different initials, and their averages of AR, JI, RI, NMI, FMI and RT are shown in Table 2. The results in Table 2 are presented with the top-ranked values shown in bold and second-place values underlined. From Table 2, the results demonstrate that S-KM2 outperforms the k-means, S-KM1, and SPCM2 algorithms in all metrics: AR, JI, RI, NMI, FMI and RT. Furthermore, the total run time of S-KM2 (0.0175 (sec.)) is even faster than that of k-means (0.0439 (sec.)). To conclude, the proposed S-KM2 algorithm performs the best on this dataset.

Table 2. Comparison among k-means, S-KM1, S-KM2, and SPCM2 for Example 4 based on the average SPCM2 for AR, JI, RI, NMI, FMI and RT.

Evaluation Criteria	AR	JI	RI	NMI	FMI	RT
k-means	0.9565	0.2016	0.5570	0.9898	0.3356	<u>0.0439</u>
S-KM1	<u>0.9999</u>	0.2897	0.5733	<u>0.9998</u>	0.3600	2.4379
S-KM2	1.0000	0.9991	0.9957	1.0000	0.9874	0.0175
SPCM2	0.9950	<u>0.7410</u>	<u>0.8550</u>	0.8210	<u>0.8470</u>	0.1230

5.2. Evaluation and Comparison with Real Datasets

This section displays some additional illustrations that demonstrate the performance of and comparisons between k-means, S-KM1, S-KM2, and SPCM2 for actual datasets. The

real-world datasets considered in this study were obtained from the UCI Machine Learning Repository [34] and are selected to provide a range of sample sizes, feature dimensionalities, and cluster structures. The experimental comparison follows the sparse possibilistic clustering framework of Yang and Benjamin [19], with SPCM2 included as a competing method. To further examine the practical effectiveness and feature selection capability of the proposed S-KM1 and S-KM2 algorithms, we conduct experiments on a diverse collection of real-world datasets. The datasets are selected from different application domains and exhibit substantial variation in sample size, feature dimensionality, and number of clusters, thereby providing a more comprehensive evaluation of sparse clustering performance. We first consider the classical Iris dataset with Example 5 and then investigate different additional datasets in our experiments from the UCI Machine Learning Repository [34]: Breast cancer, vertebral, seeds, cardio, arrhythmia, sonar, WBC, and USPS within the Iris dataset are considered separately in Example 5. These datasets cover different application domains and, importantly for sparse clustering, exhibit substantial variation in sample size, feature dimensionality, and number of clusters. The breast cancer and WBC datasets represent medical applications; The vertebral dataset provides biomechanical measurements; the seeds dataset represents agricultural data; the cardio and arrhythmia datasets provide clinical/medical measurements; sonar represents signal-processing data; and USPS provides a high-dimensional handwritten digit recognition problem. The Iris dataset [35] contains 150 observations described by four measurements—sepal length, sepal width, petal length, and petal width—and comprises three iris species, with 50 observations from each species. The dataset in Example 6 provides considerably greater diversity. The breast cancer dataset contains 569 observations and 30 features with 2 clusters, while the vertebral dataset contains 310 observations, 6 features, and 2 clusters. The seeds dataset contains 210 observations described by 7 features and consists of 3 clusters. Cardio contains 1831 observations and 21 features, whereas arrhythmia contains 452 observations and 274 features, providing a relatively high-dimensional medical data setting. Sonar contains 208 observations and 60 features with 2 clusters, while WBC and USPS provide additional medical and image recognition applications. The USPS dataset contains 9298 handwritten digit observations represented by 256 pixel features and 10-digit clusters. To reduce the effect of random initialization, each experiment is repeated 100 times using different random seeds, and the corresponding average performance is reported. The resulting experiments therefore provide a joint assessment of clustering accuracy, computational efficiency, and the ability of S-KM1 and S-KM2 to identify and eliminate irrelevant features through sparse feature weights.

We first consider the experiment with the Iris dataset [35] of Example 5. We then consider the eight real datasets breast cancer, vertebral, seeds, cardio, arrhythmia, sonar, WBC, and USPS from the UCI Machine Learning Repository [34]. We show the feature selection strength for both S-KM1 and S-KM2 algorithms for these eight real datasets. We then present the performance of the k-means, S-KM1, S-KM2, and SPCM2 algorithms and make comparisons among them for these eight real datasets in Example 6.

Example 5. This example is based on the Iris dataset [35], which contains 150 points of data and the 4 main attributes of SL (sepal length), SW (sepal width), PL (petal length) and PW (petal width). Each of the three species of iris—Iris Setosa, Iris virginica and Iris versicolor—has (50) samples combined in the dataset. We initialize weights by $w_j = 1/\sqrt{d}$ and the parameter $\lambda = 1.2$ with $d = 4$ features, and we assume $\lambda = 1.2$ to be the optimal value of the tuning parameter that enables S-KM1 to become sparse (there are non-zero weights for the remaining features). The findings presented in Table 3 show top-ranked values in bold, while the second-highest values are underlined. The findings displayed in Table 3 demonstrate the superiority of the S-KM2 algorithm over all the other algorithms (k-means, S-KM1, and SPCM2). In terms of run time in seconds, k-

means is the fastest, at 0.0084, and the S-KM2 is also fast, with 0.0654. However, S-KM1 is the slowest algorithm. This is due to the involvement of the gap statistic, which selects the tuning parameter value, being similar to the Lasso-based k -means method suggested by Witten and Tibshirani [15].

Table 3. Performances of KM, S-KM1, S-KM2 and the average-based SPCM2 in terms of AR, JI, RI, NMI, FMI and RT in Example 5.

Algorithms	Performance Measures						Feature Weights			
	AR	JI	RI	NMI	FMI	RT	PL	PW	SL	SW
k-means	0.8867	0.1364	0.4933	0.7419	0.2400	0.0084	--	--	--	--
S-KM1	<u>0.9700</u>	<u>0.8072</u>	<u>0.9289</u>	<u>0.7433</u>	<u>0.8933</u>	1.3501	0.7421	0.2353	0.6276	0.000
S-KM2	0.9733	0.9231	0.9700	0.8642	0.9550	<u>0.0654</u>	0.0000	<u>0.0820</u>	<u>0.1311</u>	0.366
SPCM2	0.9620	0.6500	0.8100	0.6950	0.7930	0.0780	<u>0.0150</u>	0.0000	0.2400	0.000

We next investigate some real datasets with different dimensions in health and medical sciences from the UCI Machine Learning Repository [34]. These are the eight real datasets breast cancer, vertebral, seeds, cardio, arrhythmia, sonar, WBC, and USPS. These datasets have also been used by researchers and scientists in various fields. We apply our proposed S-KM1 and S-KM2 algorithms on these real datasets to explore their efficiency, effectiveness, and competence in relevant feature selection, and then we compare them with k -means and SPCM2. We conduct 100 simulations of clustering results using various seed values. We also show the feature selection strength of both the S-KM1 and S-KM2 algorithms in which both algorithms can produce sparsity to feature weights and have the ability to discard the irrelevant characteristics by assigning them exactly zero weights due to the inclusion of the Lasso penalty term.

Example 6. In this experiment, we use the eight real datasets breast cancer, vertebral, seeds, cardio, arrhythmia, sonar, WBC, and USPS [34]. We employ the proposed S-KM1 and S-KM2 algorithms on these real datasets to evaluate their performance and to explore their feature selection ability as well. In Table 4, we show results from k -means, S-KM1, S-KM2, and SPCM2 in the six evaluation criteria with (average) AR, JI, RI, NMI, FMI and RT, in which bold indicates the best values, and the underlined values are the second-placed values.

Table 4. Performances of KM, S-KM1, S-KM2 and SPCM2, based on average AR, JI, RI, NMI, FMI and RT for the eight different real datasets.

Datasets	Algorithms	AR	JI	RI	NMI	FMI	RT
Breast Cancer	k -means	0.8868	0.1368	0.4936	0.7423	0.2613	0.0097
	S-KM1	0.9957	<u>0.8332</u>	0.9147	<u>0.7223</u>	<u>0.9120</u>	3.8503
	S-KM2	0.9983	0.8335	0.9147	<u>0.7229</u>	0.9223	<u>0.0148</u>
	SPCM2	<u>0.9970</u>	0.8030	<u>0.8840</u>	0.7160	0.8910	0.0860
Vertebral	k -means	0.8867	0.1370	0.4889	0.7398	0.2615	0.0097
	S-KM1	0.9969	<u>0.4863</u>	<u>0.7980</u>	0.2168	<u>0.6670</u>	2.4831
	S-KM2	<u>0.9999</u>	0.6498	0.8566	<u>0.4662</u>	0.7848	<u>0.0121</u>
	SPCM2	1.0000	0.1310	0.5540	0.1550	<u>0.6690</u>	0.0790
Sonar	k -means	0.8869	0.1357	0.4973	0.7418	0.2667	0.0098
	S-KM1	<u>0.9998</u>	<u>0.5054</u>	0.7179	0.0398	<u>0.6982</u>	2.4750
	S-KM2	0.9987	0.5647	<u>0.6699</u>	<u>0.0472</u>	0.6990	<u>0.0140</u>
	SPCM2	1.0000	0.0530	0.5030	0.0400	0.6720	0.0890
Cardio	k -means	0.8866	0.1400	0.5100	0.7322	0.2713	0.0099
	S-KM1	<u>0.9999</u>	<u>0.5323</u>	<u>0.8330</u>	0.0541	<u>0.8891</u>	2.6320

Arrhythmia	S-KM2	0.9998	0.5426	0.8417	<u>0.0661</u>	0.8651	<u>0.0132</u>
	SPCM2	1.0000	0.0960	0.8260	0.0000	0.9090	0.0892
	k-means	<u>0.8867</u>	0.1332	0.4966	0.7289	0.2667	0.0089
	S-KM1	1.0000	0.4821	0.8752	<u>0.5620</u>	0.7986	2.6320
	S-KM2	1.0000	<u>0.4565</u>	<u>0.8321</u>	0.4332	<u>0.8233</u>	<u>0.0135</u>
Seeds	SPCM2	1.0000	0.0960	0.8260	0.0000	0.9090	0.0892
	k-means	0.8867	<u>0.5888</u>	0.4990	0.7337	0.1923	0.0087
	S-KM1	0.9760	<u>0.5888</u>	0.8274	0.4049	<u>0.7411</u>	2.8866
	S-KM2	<u>0.9732</u>	0.5972	<u>0.8317</u>	0.4233	0.7476	<u>0.0242</u>
	SPCM2	0.9720	0.5220	0.7410	<u>0.5680</u>	0.7000	0.0781
USPS	k-means	0.8863	0.1267	0.4770	0.7129	0.2563	0.0099
	S-KM1	<u>0.9456</u>	0.4027	<u>0.7153</u>	<u>0.4220</u>	0.5900	2.8670
	S-KM2	0.9733	<u>0.1954</u>	0.7223	0.2820	<u>0.4001</u>	<u>0.0133</u>
	SPCM2	0.9730	0.1730	0.6280	0.2600	0.3660	0.0789
	k-means	0.8779	0.1886	0.4881	0.7336	0.3568	0.0098
WBC	S-KM1	<u>0.9973</u>	0.8878	0.8896	0.1882	0.9338	2.6670
	S-KM2	0.9733	<u>0.7742</u>	<u>0.8123</u>	<u>0.3663</u>	<u>0.8990</u>	<u>0.0249</u>
	SPCM2	1.0000	0.4230	0.6470	0.3450	0.6620	0.0880

From Table 4, we observe that the S-KM2 algorithm performs better than k-means, S-KM1 and SPCM2 for the breast cancer, vertebral, sonar and cardio datasets, but the S-KM1 algorithm performs better than the k-means, S-KM2, and SPCM2 algorithms for seeds and USPS. Both the S-KM1 and S-KM2 algorithms perform better than k-means and SPCM2 for arrhythmia and WBC. However, the k-means algorithm is always the fastest in RT, and the S-KM2 algorithm remains in second place, close to k-means in terms of RT. In general, S-KM2 is superior to SPCM2 in all metrics LAR, JI, RI, NMI, FMI, and RT. To illustrate the behaviors of both S-KM1 and S-KM2 algorithms regarding feature selections (the number of relevant features with non-zero weights), we show results in Table 5. We observe that both S-KM1 and S-KM2 have good feature selection performance. In terms of overall performance, run time, and feature selection, we recommend that S-KM2 is used in clustering.

Table 5. Comparison of S-KM1 and S-KM2 for remaining features based on different real datasets.

Real datasets	Data Points (<i>n</i>)	No. of Features (<i>d</i>)	No. of Clusters (<i>c</i>)	Non-Zero Feature Weights	
				S-KM1	S-KM2
Breast cancer	569	30	2	5	17
Vertebral	310	6	2	4	5
Sonar	208	60	2	13	36
Seeds	210	7	3	4	2
Cardio	1831	21	2	15	19
Lung	309	13	4	12	11
Red wine	1599	11	6	5	4
Arrhythmia	452	274	2	64	71

Example 7. In this experiment, we evaluate the efficiency of our proposed S-KM1 and S-KM2 algorithms by using the three large real datasets, namely GLI-85, USPS, and HARSP (Human Activity Recognition with Smartphones) [34]. We compare them with the SPCM2 method. The GLI-85 dataset has a large number of features, at 22,283, and the USPS and HARSP datasets have 9298 and 10,299 points, respectively, and 10 and 6 clusters, respectively. The USPS dataset contains 9298 16×16 pixel grayscale samples that were automatically scanned from envelopes by the U.S. Postal Service. The images are centered and normalized with a wide variety of font styles.

Machine learning algorithms often use the USPS dataset as a benchmark. A total of 9298 grayscale images are contained in this collection, with 10 clusters corresponding to digits 0 to 9. The dataset contains 256 features per image, in which each image consists of a 16×16 pixel grid. Human Activity Recognition with Smartphones (HARSP) is widely used for developing and evaluating machine learning models. The dataset contains 10,299 samples of sensor signals, 561 features, and 6 types of activities (clusters): walking, walking upstairs, walking downstairs, sitting, standing and laying. In cancer research, researchers use the GLI-85 dataset to analyze gene expression. The dataset contains 85 samples and about 22,283 gene expression features, 2 clusters of normal (non-cancerous) tissue, and glioma (cancerous) tissue. We evaluate the performance of our proposed S-KM1 and S-KM2 algorithms on these real datasets and investigate their efficiency by comparing them with SPCM2. In Table 6, we show the results from S-KM1, S-KM2 and SPCM2 based on the six evaluation criteria with (average) AR, RI, NMI, JI, FMI and RT, in which bold symbolizes the best value and underlined figures are second-placed values.

Table 6. Comparison between S-KM1, S-KM2, and SPCM2 for the three real datasets.

Real Data	Algorithms	AR	RI	NMI	JI	FMI	RT
GLI-85	S-KM1	<u>0.9992</u>	<u>0.7301</u>	0.0549	0.4175	0.5933	1.2522
	S-KM2	0.9998	0.7412	<u>0.0567</u>	0.4407	0.6118	0.5671
	SPCM2	0.9590	0.5430	0.0740	<u>0.4260</u>	<u>0.5990</u>	<u>0.7421</u>
USPS	S-KM1	<u>0.9989</u>	<u>0.6990</u>	0.0024	<u>0.1905</u>	<u>0.3999</u>	2.7442
	S-KM2	0.9999	0.7147	0.0027	0.2007	0.4021	2.1461
	SPCM2	0.9730	0.6280	<u>0.2600</u>	0.1730	0.3660	<u>2.5221</u>
HARSP	S-KM1	<u>0.9993</u>	<u>0.6289</u>	<u>0.0021</u>	<u>0.0902</u>	<u>0.2633</u>	1.5661
	S-KM2	0.9999	0.6436	0.0023	0.0926	0.2755	1.1205
	SPCM2	0.9882	0.4866	0.0001	0.0014	0.0029	<u>1.2166</u>

The results shown in Table 6 demonstrate that the S-KM2 method outperforms all other algorithms, such as S-KM1 and SPCM2. On the GLI-85 dataset, our proposed S-KM2 algorithm shows the best performance in terms of AR, RI, JI, FMI, and RT, with the exception of NMI, in which SPCM2 is the best. The AR and RI of S-KM1 are the second highest as well. For the average total run time in seconds, S-KM2 is the fastest, at 0.5671, and SPCM2 is also fast, at 0.7421. However, S-KM1 is the slowest, at 1.2522. Similarly, for the USPS and HARSP datasets, our proposed S-KM2 algorithm is better than the S-KM1 and SPCM2 algorithms based on AR, RI, NMI, JI, FMI, and RT, while S-KM1 shows second-best values. Regarding average RT in seconds, S-KM2 is the fastest, at 2.1461 and 1.1205, but SPCM2 has a smaller RT than S-KM1.

In the next example, we present how the parameters influence these clustering algorithms. We show that the tuning parameter s in S-KM1 has a large influence; thus, determining the appropriate setting of s is difficult. However, Witten and Tibshirani [15] suggest a very good estimate using the gap statistic. For this reason, we will not consider the parameters influence on S-KM1. We should consider these parameters influence for S-KM2 and SPCM2. In S-KM2, the only parameter is λ , which is exactly the same as the parameter γ in SPCM2, and so we use λ for both S-KM2 and SPCM2 in the following Example 8.

Example 8. In this experiment, we apply the S-KM2 and SPCM2 algorithms on four real datasets: Iris, leukemia, breast cancer and cardio [34]. We compare and examine their performance and influence using various parametric values of λ . We implement S-KM2 and SPCM2 for the Iris, leukemia, breast cancer and cardio datasets by using different parametric values of λ : 0.0001, 0.1000, 30.000 and 120.00. These clustering results and performance of S-KM2 and SPCM2 with different λ values are shown in Table 7. In Table 7, we show the results from S-KM2 and SPCM2

based on the six evaluation criteria with (average) AR, JI, RI, NMI, FMI and RT, in which bold text symbolizes the best values, and underlined values are the second-best values. We find that the performance of S-KM2 under different parametric values is very stable. That is, the performance of S-KM2 is not sensitive to the parameter λ . However, SPCM2 is more sensitive to the parameter λ . However, both S-KM2 and SPCM2 always give different RTs in seconds when the parametric values of λ are changed. To conclude, the results from Table 7 show that our proposed S-KM2 algorithm achieves better clustering performance than SPCM2 when using different parametric values of λ .

Table 7. Comparison of the clustering performance of S-KM2 and SPCM2 with different λ values.

RT	FMI	JI	NMI	RI	AR	λ	Algorithms	Real Datasets
0.0165	0.9600	0.9231	0.8642	0.9733	0.9997	0.0001	S-KM2	IRIS
0.0203	0.9600	0.9231	0.8642	0.9733	0.9997	0.1000		
0.0426	0.9600	0.9231	0.8642	0.9733	0.9997	30.000		
0.0411	0.9600	0.9231	0.8642	0.9733	0.9997	120.00		
0.0237	0.2867	0.1673	0.6194	0.5244	0.9978	0.0001	SPCM2	
0.0223	0.6200	0.4493	0.6194	0.7467	0.9978	0.1000		
0.0301	0.9200	0.8519	0.7925	0.9467	0.9995	30.000		
0.0303	0.9400	0.8868	0.8334	0.9600	0.9996	120.00		
0.1810	0.6690	0.5160	0.0778	0.7175	0.9976	0.0001	S-KM2	LEUKEMIA
0.1720	0.6690	0.5160	0.0778	0.7175	0.9976	0.1000		
0.1916	0.5556	0.3846	0.0003	0.7037	0.9921	30.000		
0.1612	0.5556	0.3846	0.0003	0.7037	0.9921	120.00		
0.1921	0.5762	0.3926	0.2810	0.6530	0.9450	0.0001	SPCM2	
0.2011	0.5762	0.3926	0.2810	0.6530	0.9450	0.1000		
0.1932	0.3889	0.2414	0.0606	0.5833	0.9948	30.000		
0.1722	0.4583	0.2973	0.0084	0.6389	0.9936	120.00		
0.0198	0.8541	0.7454	0.4672	0.9028	0.9997	0.0001	S-KM2	BREAST CANCER
0.0173	0.8541	0.7454	0.4672	0.9028	0.9997	0.1000		
0.0172	0.8541	0.7454	0.4672	0.9028	0.9997	30.000		
0.0169	0.8541	0.7454	0.4672	0.9028	0.9997	120.00		
0.0400	0.7010	0.4173	0.4173	0.8828	0.9997	0.0001	SPCM2	
0.4173	0.8243	0.7010	0.4173	0.8828	0.9997	0.1000		
0.0305	0.8295	0.7087	0.4261	0.8864	0.9997	30.000		
0.0282	0.8260	0.7036	0.4202	0.8840	0.9997	120.00		
2.9243	0.8494	0.7383	0.5095	0.8996	1.0000	0.0001	S-KM2	CARDIO
2.6095	0.8494	0.7383	0.5095	0.8996	1.0000	0.1000		
2.0056	0.8488	0.7373	0.5084	0.8992	1.0000	30.000		
2.9668	0.8488	0.7373	0.5084	0.8992	1.0000	120.00		
2.982	0.6504	0.4820	0.0000	0.7670	1.0000	0.0001	SPCM2	
2.7306	0.6504	0.4820	0.0000	0.7670	1.0000	0.1000		
2.7705	0.6504	0.4820	0.0000	0.7670	1.0000	30.000		
2.9743	0.6504	0.4820	0.0000	0.7670	1.0000	120.00		

Practical relevance and scope: The real data experiments demonstrate the practical relevance of the proposed S-KM1 and S-KM2 algorithms for clustering datasets containing potentially irrelevant or weakly informative variables. Unlike conventional k-means, which assigns equal importance to all features, the proposed methods jointly perform clustering and feature selection through sparse feature weights. Consequently, variables that contribute little to the underlying cluster structure can receive zero weights, reducing

their influence on the clustering solution and improving interpretability. The real datasets considered in this study cover several application domains. Specifically, the breast cancer and WBC datasets represent medical applications; vertebral contains biomechanical measurements; the seeds datasets represents agricultural data; cardio and arrhythmia involve clinical and medical measurements; sonar represents a signal-processing application; and USPS provides a high-dimensional handwritten digit recognition problem. These datasets therefore provide practical test cases for evaluating the ability of the proposed methods to identify informative variables while preserving meaningful cluster structures. The experimental results consequently provide practical evidence of the usefulness of S-KM1 and S-KM2 in addition to their theoretical properties and convergence analysis. More specialized applications, such as image segmentation and anomaly detection, are extensions of the proposed framework for future work; however, their detailed investigation requires domain-specific datasets, preprocessing procedures, and evaluation criteria.

Validation and practical applicability: The exceptionally high performance observed for some datasets can be interpreted in the context of their underlying clustering structures rather than as evidence of universally near-perfect performance. In the synthetic experiments, the cluster structures and relevant/irrelevant features were explicitly controlled, allowing the feature selection and clustering capabilities of the proposed methods to be examined under known conditions. First, the high performance observed in some synthetic experiments is attributable to the controlled nature of their data-generating mechanisms. In these experiments, the cluster structure and the distinction between relevant and irrelevant features were explicitly specified. For example, in the Gaussian mixture experiment, only three of the fifteen features were relevant, whereas the remaining twelve features were generated as irrelevant variables. Such a controlled setting is intentionally designed to assess whether a sparse clustering method can recover the underlying cluster structure while identifying informative features. Consequently, near-perfect values in this setting should be interpreted as evidence of successful recovery of the predefined structure rather than as evidence of universal near-perfect performance. Second, to reduce the influence of random initialization, each experiment was repeated 100 times using different initial values/random seeds, and the average performance was reported. Third, the evaluation was extended beyond synthetic data to a diverse collection of real-world datasets obtained from the UCI Machine Learning Repository. These datasets include breast cancer, vertebral, seeds, cardio, arrhythmia, sonar, WBC, and USPS, and they differ substantially in sample size, dimensionality, and application domain. For instance, arrhythmia contains 452 observations and 274 features, whereas USPS contains 9298 observations represented by 256-pixel features. The datasets therefore provide substantially more heterogeneous settings than the controlled synthetic experiments and include medical, biomechanical, agricultural, signal-processing, and image recognition applications. Importantly, this study concerns unsupervised clustering rather than supervised prediction. Since our study uses an unsupervised clustering framework, the ground truth labels were not used during clustering or feature weight estimation. They were used only after clustering to evaluate the obtained cluster assignments using AR, JI, RI, NMI, and FMI. Thus, the reference labels did not serve as training information for the proposed algorithms. Thus, the reference labels were used only for the final evaluation of clustering performance and were not incorporated into the optimization, feature weight estimation, or cluster assignment procedures. The proposed method therefore operates in a fully unsupervised manner. The same experimental conditions were applied to all competing methods to ensure a fair and consistent comparison. Further validation on independent, noisy, and domain-specific datasets would therefore be valuable for assessing the broader generalizability of the proposed methods.

6. Conclusions

In this article, we proposed two clustering algorithms, S-KM1 and S-KM2, by incorporating the Lasso L_1 penalty so that they can effectively select relevant attributes by applying non-zero weights to the most significant attributes and zero weights to the insignificant ones. Furthermore, we demonstrated the necessary conditions for their optimization through updating equations. We made comparisons between S-KM1, S-KM2, k-means, and SPCM2 on the following performance criteria: AR, RI, NMI, JI, and FMI. The proposed algorithm demonstrates better performance on both the synthetic and real datasets. The S-KM2 exhibits optimal performance in terms of the RT criterion. The findings of this article demonstrate that S-KM2 performs better on both artificial and real datasets, but it is also capable of providing good clustering structure. The results illustrate that our proposed S-KM2 algorithm can identify significant attributes by assigning non-zero weights to them while discarding unimportant attributes by assigning zero weights to them. Although the proposed S-KM2 demonstrates good performance in clustering, it needs to have a given number of clusters a priori. Our future work will aim to advance the study of S-KM2 without providing the number of clusters so that it can automatically find an optimal number of clusters during iterations. On the other hand, we will further extend the proposed algorithms to clustering multi-view datasets. The experimental results demonstrate that S-KM1 and S-KM2 provide competitive clustering performance and effective feature selection compared with the methods considered in this study. In particular, S-KM2 shows consistently strong performance across the evaluated synthetic and real datasets. These findings support the effectiveness of the proposed approaches for the datasets investigated, while broader comparisons with additional recent sparse clustering methods remain an important direction for future research.

Author Contributions: Conceptualization, M.-S.Y. and S.P.; methodology, M.-S.Y. and S.P.; validation, M.-S.Y.; formal analysis, S.P.; investigation, M.-S.Y. and S.P.; resources, M.-S.Y.; data curation, S.P.; writing—original draft preparation, S.P.; writing—review and editing, M.-S.Y.; supervision, M.-S.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: All data generated or analyzed during this study are included in this article.

Acknowledgments: This work was supported in part by the National Science and Technology Council, Taiwan, under grant NSTC 114-2118-M-033-002.

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

References

1. Backhaus, K.; Erichson, B.; Gensler, S.; Weiber, T. *Multivariate Analysis: An Application-Oriented Introduction*; Springer: Wiesbaden, Germany, 2021.
2. Guyon, I.; Elisseeff, A. An introduction to variable and feature selection. *J. Mach. Learn. Res.* **2003**, *3*, 1157–1182.
3. Bouveyron, C.; Girard, S.; Schmid, C. High dimensional data clustering. *Comput. Stat. Data Anal.* **2007**, *52*, 502–519.
4. Ikotun, A.M.; Ezugwu, A.E.; Abualigah, L.; Abuhaija, B.; Heming, J. K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. *Inf. Sci.* **2023**, *622*, 178–210.
5. Yang, M.S.; Sinaga, K.P. Federated multi-view k-means clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* **2025**, *47*, 2446–2459.
6. Bezdek, J.C. *Pattern Recognition with Fuzzy Objective Function Algorithms*; Plenum Press: New York, NY, USA, 1981.
7. Hussain, I.; Sinaga, K.P.; Yang, M.S. Unsupervised multi-view fuzzy c-means clustering. *Electronics* **2023**, *12*, 4467.
8. Krishnapuram, R.; Keller, J.M. A possibilistic approach to clustering. *IEEE Trans. Fuzzy Syst.* **1993**, *1*, 98–110.
9. Yu, H.; Wu, Y.; Zheng, H.; Luo, Q.; Zhang, L. Possibilistic c-means clustering approach based on a novel weighted-kernel distance for imbalanced images with minority targets in sparsely distribution. *Eng. Appl. Artif. Intell.* **2025**, *154*, 110902.

10. Nataliani, Y.; Yang, M.S. Powered Gaussian kernel spectral clustering. *Neural Comput. Appl.* **2019**, *31*, 557–572.
11. Cheng, D.; Jiang, X.; Xia, S.; Wang, G. Fast spectral clustering via pseudo-label-based granular-ball division for large-scale data. *IEEE Trans. Knowl. Data Eng.* **2026**, *38*, 2807–2817.
12. Hong, C.; Yu, J.; Tao, D.; Wang, M. Image-based three-dimensional human pose recovery by multiview locality-sensitive sparse retrieval. *IEEE Trans. Ind. Electron.* **2015**, *62*, 3742–3751.
13. Yu, J.; Tan, M.; Zhang, H.; Rui, Y.; Tao, D. Hierarchical deep click feature prediction for fine-grained image recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *44*, 563–578.
14. Tibshirani, R. Regression shrinkage and selection via the Lasso. *J. R. Stat. Soc. Ser. B* **1996**, *58*, 267–288.
15. Witten, D.M.; Tibshirani, R. A framework for feature selection in clustering. *J. Am. Stat. Assoc.* **2010**, *105*, 713–726.
16. Kondo, Y.; Salibian-Barrera, M.; Zamar, R. RSKC: An R package for a robust and sparse k-means clustering algorithm. *J. Stat. Softw.* **2016**, *72*, 1–26.
17. Castro, E.A.; Pu, X. A simple approach to sparse clustering. *Comput. Stat. Data Anal.* **2017**, *105*, 217–228.
18. Qiu, X.; Qiu, Y.; Feng, G.; Li, P. A sparse fuzzy c-means algorithm base on sparse clustering framework. *Neurocomputing* **2015**, *157*, 290–295.
19. Yang, M.S.; Benjamin, J.B.M. Sparse possibilistic c-means clustering with Lasso. *Pattern Recognit.* **2023**, *138*, 109348.
20. Hartigan, J.A.; Wong, M.A. Algorithm AS 136: A k-means clustering algorithm. *J. R. Stat. Soc. Ser. C (Appl. Stat.)* **1979**, *28*, 100–108.
21. 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.
22. 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.
23. Ma, M.; Yavuz, M.; Hudnall, M.; Wang, Q. C2slm: A correlation-based clustering-assisted sparse learning model for electric vehicle market demand forecasting. *Pattern Recognit.* **2026**, *170*, 111984.
24. Tutsoy, O.; Sumbul, H.E. Enhanced multiclass cardiovascular disease detection via Kernel-PSO feature selection: A comparative benchmark of gradient-driven and bio-inspired machine learning algorithms. *Swarm Evol. Comput.* **2026**, *105*, 102405.
25. Dong, X.; Nie, F.; Tian, L.; Wang, R.; Li, X. Unsupervised discriminative feature selection with $\ell_{2,0}$ -norm constrained sparse projection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2025**, *47*, 8321–8335.
26. Tibshirani, R.; Walther, G.; Hastie, T. Estimating the number of clusters in a data set via the gap statistic. *J. R. Stat. Soc. Ser. B (Stat. Methodol.)* **2001**, *63*, 411–423.
27. Williams, L.F., Jr. A modification to the half-interval search (binary search) method. In *Proceedings of 14th Annual Southeast Regional Conference, ACM-SE 14*; Association for Computing Machinery: New York, NY, USA, 1976; pp. 95–101.
28. Zangwill, W.I. *Nonlinear Programming: A Unified Approach*; Prentice-Hall: Englewood Cliffs, NJ, USA, 1969.
29. Werner, F.; Sotskov, Y.N. *Mathematics of Economics and Business*; Routledge, Taylor & Francis Group: London, UK; New York, NY, USA, 2006.
30. Jaccard, P. Distribution de la flore alpine dans le bassin des Dranses et dans quelques régions voisines. *Bull. Société Vaudoise Sci. Nat.* **1901**, *37*, 241–272.
31. Rand, W.M. Objective criteria for the evaluation of clustering methods. *J. Am. Stat. Assoc.* **1971**, *66*, 846–850.
32. Cover, T.M.; Thomas, J.A. *Elements of Information Theory*; Wiley: New York, NY, USA, 1991.
33. Fowlkes, E.B.; Mallows, C.L. A method for comparing two hierarchical clusterings. *J. Am. Stat. Assoc.* **1983**, *78*, 553–569.
34. Blake, C.L.; Merz, C.J.U. CI Repository of Machine Learning Databases, A Huge Collection of Artificial and Real-World Data Sets, 1998. Available online: <https://archive.ics.uci.edu/ml/datasets.html> (accessed on 3 March 2025).
35. Fisher, R.A. The use of multiple measurements in taxonomic problems. *Ann. Eugen.* **1936**, *7*, 179–188.

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.