K-Initialization-Similarity Clustering Algorithm

Tong Liu and Gaven Martin

***Abstract*—As one of the most used clustering algorithms, K-means clustering algorithm has been applied in variety areas. Its clustering result depends on the predefined cluster number, the initialization, and the similarity measure. Previous research focused on solving parts of these issues but has not focused on solving them in a unified framework. However, fixing one of these issues does not guarantee the best performance. To improve K-means clustering algorithm, we propose the K-Initialization-Similarity (KIS) clustering algorithm to solve the issues of the K-means clustering algorithm in a unified way. Specifically, we propose to learn the similarity matrix based on the data distribution, to automatically output the cluster number using a robust loss function, and to fix the initialization by using sum-of-norms which outputs the new representation of the original samples. The proposed algorithms outperformed the state-of-the-art clustering algorithms on real data sets. Moreover, we theoretically prove the convergences of the proposed optimization methods for the proposed objective function.**

***Index Terms*—Clustering, K-means, Spectral clustering, Machine learning, Initialization, Similarity measure.**

I. INTRODUCTION

K-means clustering algorithm is considered one of the most used clustering algorithms. It has been successfully applied to broad areas such as artificial intelligence, machine learning, data mining, etc.

K-means clustering algorithm partitions the dataset into *K* distinct clusters in the following steps: First, it initializes cluster centers via randomly selecting *K* data points as the *K* cluster centers. Second, it assigns each data point to its nearest cluster center according to a similarity measure, e.g., Euclidean distance. Third, it revises the *K* cluster centers using the mean of assigned data points in each cluster. K-means clustering algorithm keeps repeating the last two steps until the algorithm achieves convergence [1, 2].

As one of the most famous and widely used clustering algorithm, K-means clustering algorithm still has its limitations. It is difficult to determine the cluster number *K* without prior knowledge. Different initializations may obtain completely different clustering results. K-means clustering results depend on the similarity measure such as Euclidean distance measure which does not account for the factors such as cluster sizes, dependent features or density [3, 4]. Thus K-means clustering algorithm is not good for indistinct or not well-separated data sets [5, 6]. Existing methods only solved some of these problems. All these issues of K-means clustering algorithm are important to be addressed to improve the performance of K-means clustering algorithm. Many literatures have solved some parts of these issues of K-means clustering algorithm [7-9]. For example, Duan et al. developed an algorithm to calculate the density to select the initial cluster centers [10]. Lakshmi et al. proposed to use nearest neighbors and feature means to decide the initial cluster centers [8]. Other works also addressed the issues of K-means clustering algorithm [11-14].

However, previous clustering algorithms only fixed parts of the issues of the K-means clustering algorithm. When a clustering algorithm addresses those problems separately, it is easily to be trapped into the sub-optimal results, which means it is hard to obtain a global optimal solution, for example, even if a best initial value is found or the best similarity matrix is found, but the final optimal results may not be obtained. Because the results of the individual steps are not obtained according to the requirements of the next steps. It would be significant if we could fix the issues of the initialization, cluster number determination and similarity measure problems of K-means clustering algorithm in a unified framework to achieve global optimal results.

Our proposed new K Initialization Similarity (KIS) algorithm is aimed to develop an improved K-means clustering algorithm while solving the issues of the cluster number determination, the initialization, the similarity measure in a unified way. Specifically, the cluster number is automatically generated by using a robust loss function, the initialization of the clustering using sum-of-norms (SON) regularization, the similarity matrix based on the data distribution. Furthermore, we employ the alternative strategy to solve the proposed objective function. The experimental results on real-world benchmark data sets also demonstrates that our KIS clustering algorithm outperforms the related clustering methods in terms of accuracy (Acc), the assessment evaluation metric for clustering algorithm [1].

We briefly summarize the contributions of our proposed KIS algorithm as follows:

- A unified way addresses the cluster number determination, initialization, and similarity measure issues around clustering.
- The cluster number is automatically generated using a robust loss function.
- The initialization is fixed by using the sum-of-norms regularization
- The similarity measure is generated based on the data distribution
- The proposed clustering algorithm outperforms comparison clustering algorithms. It implies that simultaneously addressing the three issues (cluster number determination, initialization, and similarity measure) is feasible and robust.

This section has laid the background of our research inquiry. The remainder of this paper is organized as follows. Section 2 discusses the existing relevant clustering algorithm. Section 3 introduces our KIS algorithm. Section4 discusses the experiments we conducted and present the results of our

experiments. The conclusions and future research direction are presented in Section 5.

II. Related Work

Clustering can be generally categorized into the prototype-based and the non-prototype-based approaches, based on whether the clustering algorithm is center-based or not.

A. Prototype-based clustering Algorithms

The prototype of the corresponding cluster is the center of the data points in each cluster. The prototype-based clustering algorithms assign data points to their closest prototypes, such that data points in the group are closer to the prototype of the cluster than to the prototype of any other group. K-means clustering algorithm is one of the most famous representatives of this kind of clustering approaches [15, 16]. It keeps recalculating the prototypes followed by assigning each data point to a cluster represented by a prototype until the algorithm achieves convergence [1]. There are numbers of other algorithms based on prototype clustering algorithms, e.g. K-medoids, COTCLUS, and Tabu search. K-medoids chooses the data points located near their prototypes to represent the clusters. The rest of remaining data points are clustered with the representative prototype to which they are the most similar based on the minimal sum of the dissimilarities between data points and their corresponding cluster prototypes [17]. Instead of using only one center for each class, COTCLUS, an improved prototype-based clustering algorithm, uses suitable prototypes from another cluster. It finds two prototypes from one cluster and replace them by two prototypes from the other cluster in such a way that maximum decreases the mean square error of the first clustering. It constructs a clustering from two suboptimal clustering results based on the belief that each suboptimal clustering has benefits regarding to containing some of the correct clusters [18]. After modifying centroids, it applies K-means clustering algorithm for final fine-tuning [18]. A Tabu based clustering algorithm employs the prototype driven approach of the K-means clustering algorithm with the guidance of Tabu search, which is a local or neighborhood search algorithm that accepts the worsening searches of no improving search is available and discourages the search from going back to previously visited search [19]. The K-medoids, COTCLUS, and Tabu search example like other K-means clustering algorithm need to specify the cluster number *K* before the execution of the algorithms.

B. Non-Prototype-based clustering algorithms

Instead of conducting clustering based on the cluster centers by the prototype approaches, some non-prototype-based clustering algorithms use links or graph. Robust clustering using links (ROCK) [20]. ROCK clustering algorithm draws a number of data points randomly from the original data set as inputs along with the desired cluster number *K*. Instead of using distances to conduct clustering, ROCK uses the number of links which is defined as the number of common neighbors as the similarity measure [20]. But ROCK ignores the possible differences in the similarity measure of different clusters inside the same data set. Graph is also used for representing the high-order relationship among data points [21]. A graph is a set of nodes with connected edges which have weights associated with them [22]. Spectral clustering algorithm is an example of clustering algorithms using graph. It creates a similarity matrix first and then defines a feature vector. Then it runs the K-means clustering algorithm to conduct clustering [23]. It creates the spectral representation and conducts the final clustering in separate stages, and it requires the cluster number beforehand because it uses of the K-means clustering algorithm. Low-rank representation (LRR) identifies the subspace structures from data points and then finds the lowest rank representation among data points to represent the original data points [24]. A low-rank kernel learning graph-based clustering (LKLGC) algorithm is based on a multiple kernel learning with assumption that the consensus kernel matrix is a low-rank matrix and lies in the neighbourhood of the combined kernel matrix [25]. The spectral clustering algorithm is applied to get the final clustering results for LKLGC algorithm, hence it is a multi-stage clustering and the cluster number needs to be predefined [25]. A low-rank kernel learning for Graph-based Clustering (LKG) iteratively constructs graph and kernel learning which exploits the similarity of the kernel matrix and an optimal kernel from the neighboring candidate kernels [11]. It requires the cluster number beforehand. A hybrid representative selection based ultra-scalable spectral clustering (U-SPEC) constructs a sparse affinity sub-matrix by using a hybrid representative selection strategy and a K-nearest representatives approximation method, and then interprets the sparse sub-matrix as a bipartite graph, which is partitioned using transfer cut to obtain the clustering result [12]. It is a multi-stage clustering and cluster number is prerequisite.

Current prototype-based and non-prototype-based clustering algorithms do not simultaneously solve the initialization, similarity measure or cluster number issues of non-graph-based clustering algorithms.

III. K-Initialization-Similarity Clustering

Given a data matrix $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \in \mathbb{R}^{n \times d}$, where n and d, respectively, are the number of samples and features, we denote boldface uppercase letters, boldface lowercase letters,

TABLE I
Description of symbols used in this paper

Symbol	Description
$\mathbf{X}$	Data matrix
$\mathbf{x}$	A vector of $\mathbf{X}$
$\mathbf{x}_i$	The i-th row of $\mathbf{X}$
$x_{i,j}$	The element in the i-th row and j-th column of $\mathbf{X}$
$\|\mathbf{x}\|_2$	L_2-norm of $\mathbf{x}$
$\|\mathbf{X}\|_F$	The Frobenius norm or the Euclidean norm of $\mathbf{X}$
$\mathbf{X}^T$	The transpose of $\mathbf{X}$
K	Cluster number

and italic letters as matrices, vectors and scalars, respectively, and also summarize the symbols used in this paper in Table I.

A. Motivation

To find out how other algorithms improve K-means clustering algorithm, we investigate Spectral clustering algorithm, Robust Clustering using links, Low-rank kernel learning for graph-based clustering (LKG), and Ultra-scalable spectral clustering (U-SPEC) beside K-means clustering algorithm in detail.

K-means algorithm aims at minimizing the total intra-cluster variance represented by an objective function known as the squared error function shown in Eq. (1):

$$E = \sum_{j=1}^{K} \sum_{i \in C_j} \left\| i_l - w_j \right\|^2 \quad (1)$$

where K is number of clusters, $j \in \{1, \dots, K\}$, n is number of data points, $l \in \{1, \dots, n\}$, $(w_1, \dots, w_K)$ is the K prototypes. C_j is the j^{th} cluster. K-means clustering algorithm operates in the following steps: First, it initializes k prototypes $(w_1, \dots, w_K)$ via randomly selecting K data points from $l \in \{1, \dots, n\}$. Second, it assigns each i_l to the cluster C_j with w_j, each C_j is associated with w_j. Third, it updates the prototype w_j for each cluster C_j using the mean. K-means clustering algorithm keeps repeating the last two steps until the E doesn't change or change insignificantly [12].

Instead of using original data points, Spectral clustering algorithm conducts K-means clustering on spectral representation. To do this, Spectral clustering algorithm first creates a similarity matrix, and then constructs a diagonal degree matrix using the sum of all the weights on each row of the similarity matrix and a feature vector by computing the first K eigenvectors of its Laplacian matrix, which is the degree matrix subtracting the similarity matrix. Finally, it runs K-means clustering on these features to separate objects into K clusters [23]. Spectral clustering algorithm is a multi-step algorithm and it requires the cluster number to be predefined.

Robust clustering using links (ROCK) obtains a number of random data points from the original data, then uses the link agglomerative approach with a goodness measure, which determines which pair of points is merged at each step as shown in Eq. (2). Finally, the remaining data points are assigned to these clusters [20].

$$g(K_i, K_j) = \frac{\text{link}(K_i, K_j)}{(n_i + n_j)^{1+2f(\theta)} - n_i^{1+2f(\theta)} - n_j^{1+2f(\theta)}} \quad (2)$$

where $\text{link}(K_i, K_j)$ is the number of links between the two clusters. n_i and n_j are the number of points in each cluster. The function f satisfies the property that each item in K_i, has approximatelt $n_i^{f(\theta)}$ neighbors in the cluster. The reasoning behind using link is that the data points belonging to the same cluster most likely have a large number of common neighbours, thus more links. Hence the larger the number of links between data points, the greater likelihood they belong to the same cluster. But ROCK ignores the possible differences the similarity of different data points and it require the cluster number beforehand as well.

Low-rank kernel learning for graph-based clustering (LKG) constructs graph and learns consensus kernel in a unified framework. Its low-rank kernel matrix is learnt by exploiting the similarity of the kernel matrix and seeking an optimal kernel from the neighboring of candidate kernels. The graph and kernel are iteratively enhanced by each other. LKG runs the spectral clustering algorithm to achieve the final clustering results.

$$\min_{Z,K,\mathbf{g}} \frac{1}{2} Tr\|K - 2KZ + Z^T K Z\|_F^2 + \alpha\rho(Z) + \beta\|K\|_* + \gamma\left\|K - \sum_i \mathrm{g}_i H^i\right\|_F^2,\ s.t.\ Z \geq 0, K \geq 0, \mathrm{g}_i \geq 0, \sum_i^r \mathrm{g}_i = 1 \quad (3)$$

where Z is self-expression coefficient, K is nonnegative kernel matrix, H is kernel matrix, the weight of kernel H^i is g_i, the constraints for g are from standard Multiple kernel learning method. The corresponding $\mathrm{g}_i$will be assigned a small value if a kernel is not appropriate. $\|K\|_*$ is the structure of the kernel matrix, where K will respect the correlations among data points with the cluster structure. The last term in Eq. (3) seeks an optimal kernel K in the neighborhood of $\sum_i \mathrm{g}_i H^i$. Z and K repeatedly learnt in a unified model. LKG reinforces the underlying connections between the optimal kernel learning and graph learning.

To improve the randomness and efficiency of K-means cluster algorithm, a hybrid representative selection based ultra-scalable spectral clustering (U-SPEC) was designed. It interprets the sparse sub-matrix as a bipartite graph, which is partitioned using transfer cut to obtain the clustering result. U-SPEC algorithm conducts in three phases. In the first phase, a hybrid representative selection strategy is applied by randomly selecting candidates and obtaining representatives from candidates via K-means. In the second phase, a coarse-to-fine method is used to approximate the K-nearest representatives for each data points, and to construct a sparse affinity sub-matrix between the data points and the representatives. The sparse affinity sub-matrix is represented by the Eq. (4). In the third phase, the sub-matrix is interpreted as a bipartite graph, which is partitioned to the final clusters.

$$B = \{b_{i,j}\}_{Nxp}, b_{i,j} = \begin{cases} \exp\left(-\frac{\|x_i - r_j\|^2}{2\sigma^2}\right), \text{if } r_j \in N_K(x_i) \\ 0, \text{ otherwise} \end{cases} \quad (4)$$

where $N_K(x_i)$ represents the set of K-nearest representatives of x_i and is the average Euclidean distance between the data points and their K-nearest representatives.

Previous clustering algorithms only fixed part of the issues of the K-means clustering algorithm. It would be significant for our KIS clustering algorithm fixing the issues of the initialization, cluster number determination and similarity measure problems of K-means clustering algorithm in a unified framework to achieve global optimal results.

B. K-Initialization-Similarity Clustering Algorithm

This paper proposes a new clustering algorithm (i.e., K-Initialization-Similarity (KIS)) to simultaneously solve the cluster number determination, the initialization issue, and the similarity measure issue of K-means clustering algorithm in a unified framework. Specifically, KIS clustering algorithm

generates the new representation of original data points, applies sum-of-square error estimation to minimize the difference between the original data and its new representative, learns the similarity matrix based on the data distribution, and generated the cluster number *K* by using the robust loss function. To achieve our goal, we form the objective function of the KIS clustering algorithm as follows:

$$\min_{\mathbf{S},\mathbf{U}} \frac{1}{2}\|\mathbf{X}-\mathbf{U}\|_F^2 + \frac{\alpha}{2}\sum_{i,j=1}^{n} s_{i,j}\rho(\|\mathbf{u}_i-\mathbf{u}_j\|_2) + \beta\|\mathbf{S}\|_2^2, \quad s.t., \forall i, s_{i,j} \geq 0, \mathbf{s}_i^T\mathbf{e} = 1 \tag{5}$$

where $\mathbf{X} \in \mathbb{R}^{n\times d}$ is the data matrix, $\mathbf{U} \in \mathbb{R}^{n\times d}$ is the new representation of $\mathbf{X}$, and $\mathbf{S} \in \mathbb{R}^{n\times n}$ is the similarity matrix to measure the similarity among data points, and $\rho\left(\|\mathbf{u}_i-\mathbf{u}_j\|_2\right)$ is a robust loss function, used for automatically generating clusters. The smaller the value of $\|\mathbf{u}_i-\mathbf{u}_j\|_2$ is, the closer the distance is, and the higher the similarity $\mathbf{s}_i$ and $\mathbf{s}_j$is. With the update of other parameters in Eq. (5), the distance $\|\mathbf{u}_i-\mathbf{u}_j\|_2$ for some i and j, will be very close, or even $\mathbf{u}_i = \mathbf{u}_j$. The clusters will be determined. $\boldsymbol{e} = [\mathbf{1}, \dots, \mathbf{1}]^T$.

Eq. (5) fixes the initialization of clustering, automatically learns the new representation **U** and the similarity matrix **S**, and generates the cluster number. The similarity matrix **S** learning is based on the data distribution, i.e., iteratively updated by the updated **U**. This produces an intelligent new representation of the original data matrix.

Moreover, Eq. (5) will keep the distance of indicator vectors similar if the data belongs to the same cluster, possibly making them equal. The distance of indicator vectors is as separated as possible if data belongs to the different clusters.

Several robust loss functions have been proposed to avoid the influence of noise and outliers in robust statistics [26]. Here we employ the Geman-McClure function [27]:

$$\mathrm{P}\left(\|\mathbf{u}_p-\mathbf{u}_q\|_2\right) = \frac{\mu\|\mathbf{u}_p-\mathbf{u}_q\|_2^2}{\mu+\|\mathbf{u}_p-\mathbf{u}_q\|_2^2} \tag{6}$$

The literature of half-quadratic minimization and robust statistics explains the reason for selecting Geman–McClure loss function instead of other loss functions [28]. Eq. (6) measures how well a model predicts the expected outcome. The smaller the value of $\|\mathbf{u}_p-\mathbf{u}_q\|_2^2$ is, the closer the distance is, and the higher the similarity s_p and s_q is. With the update of other parameters in Eq. (6), the distance $\|\mathbf{u}_p-\mathbf{u}_q\|_2^2$ for some p and q, will be very close, or even $\mathbf{u}_p = \mathbf{u}_q$. The clusters will be determined.

The optimization of the robust loss function is challenging. To address this, it is normal practice to introduce an auxiliary variable $f_{i,j}$ and a penalty item $\varphi(f_{i,j})$, and thus Eq. (5) is rewritten to:

$$\min_{\mathbf{S},\mathbf{U},\mathbf{F}} \frac{1}{2}\sum_{i=1}^{n}\|\mathbf{x}_i-\mathbf{u}_i\|_2^2 + \frac{\alpha}{2}\sum_{i,j=1}^{n} s_{i,j}(f_{i,j}\|\mathbf{u}_i-\mathbf{u}_j\|_2^2 + \varphi(f_{i,j})) + \beta\sum_{i=1}^{n}\|\mathbf{s}_i\|_2^2 \;\; s.t., \forall i, s_{i,j} \geq 0, \boldsymbol{s}_i^T\boldsymbol{e} = 1 \tag{7}$$

where $\varphi(f_{i,j}) = \mu\left(\sqrt{f_{i,j}}-1\right)^2, i, j = 1 \dots n$

This objective function is still challenging to solve. An iterative optimization process is adopted to tackle this challenge. In the next section, we will show how iterative optimization is utilized to solving the problem.

C. Optimization

Eq. (7) is not jointly convex on **F**, **U**, and **S**, but is convex on each variable while fixing the rest. To solving the Eq. (7), the alternating optimization strategy is applied. We optimize each variable while fixing the rest until the algorithm converges. The pseudo-code of KIS clustering algorithm is given in Algorithm 1.

*1) Update **F** while fixing **S** and **U***

While **S** and **U** are fixed, the objective function can be rewritten in a simplified matrix form to optimize **F**:

$$\mathrm{Min}_{\mathbf{F}} \frac{\alpha}{2}\sum_{i,j=1}^{n} s_{i,j}(f_{i,j}\|\mathbf{u}_i-\mathbf{u}_j\|_2^2 + \mu\left(\sqrt{f_{i,j}}-1\right)^2) \tag{8}$$

Since the optimization of $f_{i,j}$ is independent of the optimization of other $f_{p,q}, i \neq p, j \neq q$, the $f_{i,j}$ is optimized first as shown in following Eq. (9)

$$\mathrm{Min}_{f_{i,j}} \frac{\alpha}{2}(s_{i,j}f_{i,j}\|\mathbf{u}_i-\mathbf{u}_j\|_2^2 + s_{i,j}\left(\mu\left(f_{i,j}-2\sqrt{f_{i,j}}+1\right)\right) \tag{9}$$

By conducting a derivative on Eq. (9) with respect to $f_{i,j}$, we get Eq. (10).

$$\frac{\alpha}{2}(s_{i,j}\|\mathbf{u}_i-\mathbf{u}_j\|_2^2 + s_{i,j}\mu - s_{i,j}\mu f_{i,j}^{-\frac{1}{2}}) = 0 \tag{10}$$

$$\Rightarrow \frac{\alpha}{2}\, s_{i,j}\|\mathbf{u}_i-\mathbf{u}_j\|_2^2 + \frac{\alpha}{2}\, s_{i,j}\mu - \frac{\alpha}{2}s_{i,j}\mu f_{i,j}^{-\frac{1}{2}} = 0 \tag{11}$$

$$\Rightarrow f_{i,j} = \left(\frac{\mu}{\mu+\|\mathbf{u}_i-\mathbf{u}_j\|_2^2}\right)^2 \tag{12}$$

*2) Update **S** while fixing **U** and **F***

While fixing **U** and **F**, the objective function Eq. (7) with respect to **S** is:

$$\mathrm{Min}_{\mathbf{S}} \frac{\alpha}{2}\sum_{i,j=1}^{n}(s_{i,j}f_{i,j}\|\mathbf{u}_i-\mathbf{u}_j\|_2^2 + s_{i,j}(\mu\left(\sqrt{f_{i,j}}-1\right)^2)) + \beta\sum_{i=1}^{n}\|\mathbf{s}_i\|_2^2, s.t., \forall i, s_{i,j} \geq 0, \mathrm{s}_i^T\mathrm{e} = \mathrm{I} \tag{13}$$

Since the optimization of $\mathbf{s}_\mathrm{I}$ is independent of the optimization of other $\mathbf{s}_j, i \neq \mathrm{j}, i, j = 1, \dots, \mathrm{n}$, the $\mathbf{s}_i$ is optimized first as shown in following:

$$\mathrm{Min}_{\mathbf{s}_i} \frac{\alpha}{2}\sum_{j=1}^{n} s_{i,j}(f_{i,j}\|\mathbf{u}_i-\mathbf{u}_j\|_2^2 + \mu\left(\sqrt{f_{i,j}}-1\right)^2) + \beta\|\mathbf{s}_i\|_2^2, \quad s.t., \forall i, s_{i,j} \geq 0, \mathrm{s}_i^T\mathrm{e} = 1 \tag{14}$$

Let $b_{i,j} = f_{i,j}\|\mathbf{u}_i-\mathbf{u}_j\|_2^2$ and $c_{i,j} = \mu\left(\sqrt{f_{i,j}}-1\right)^2$, Eq. (14) is equivalent to:

$$\mathrm{Min}_{\mathbf{s}_i} \frac{\alpha}{2}\sum_{j=1}^{n} s_{i,j}b_{i,j} + \frac{\alpha}{2}\sum_{j=1}^{n} s_{i,j}c_{i,j} + \beta\|\mathbf{s}_i\|_2^2, s.t., \forall i, s_{i,j} \geq 0, \boldsymbol{s}_i^T e = 1 \tag{15}$$

$$\Rightarrow \min_{\mathbf{s}_i} \frac{\alpha}{2}\mathbf{s}_i^T\mathbf{b}_i + \frac{\alpha}{2}\mathbf{s}_i^T\mathbf{c}_i + \beta\|\mathbf{s}_i\|_2^2, \quad s.t., \forall i, s_{i,j} \geq 0, \mathbf{s}_i^T e = 1 \tag{16}$$

$$\Rightarrow \min_{\mathbf{s}_i} \mathbf{s}_i^T\mathbf{s}_i + 2\mathbf{s}_i\frac{\alpha}{4\beta}\mathbf{s}_i^T(\mathbf{b}_i+\mathbf{c}_i) + \frac{\alpha}{4\beta}\mathbf{s}_i^T(\mathbf{b}_i+\mathbf{c}_i)^T(\mathbf{b}_i+\mathbf{c}_i) - \frac{\alpha}{4\beta}\mathrm{s}_i^T(\mathbf{b}_i+\mathbf{c}_i)^T(\mathbf{b}_i+\mathbf{c}_i), s.t., \forall i, s_{i,j} \geq 0, \mathbf{s}_i^T e = 1 \tag{17}$$

$$\Rightarrow \min_{\mathbf{s}_i} \left\|\mathbf{s}_i + \frac{\alpha}{4\beta}(\mathbf{b}_i+\mathbf{c}_i)\right\|_2^2, s.t., \forall i, s_{i,j} \geq 0, \mathbf{s}_i^T\mathrm{e} = 1 \tag{18}$$

According to Karush-Kuhn-Tucker (KKT) [29], the optimal solution $\mathbf{s}_i$ should be

$$S_{i,j} = \max\{-\frac{\alpha}{4\beta}(b_{i,j} + c_{i,j})\} + \theta, 0\}, j = 1, \dots, n \quad (19)$$

where $\theta = \frac{1}{\rho}\sum_{j=1}^{\rho}\left(\frac{\alpha}{4\beta}(b_{i,j} + c_{i,j}) + 1\right)$, and ω is the descending order of $\frac{\alpha}{4\beta}(b_{i,j} + c_{i,j})$. and $\rho = {}^{max}_{j}\{\boldsymbol{\omega}_j - \frac{1}{j}\left(\sum_{r=1}^{j}\boldsymbol{\omega}_r - 1\right), 0\}$.

Algorithm 1. The pseudo code for KIS clustering algorithm

Input: $\mathbf{X} \in \mathbb{R}^{n \times d}$
Output: a set of *K* clusters

Initialization: **U** = **X**;
Repeat:
- Update **F** using Eq. (12)
- Update **S** using Eq. (19)
- Update **U** using Eq. (24)

Until U converges

*3) Update **U** while fixing **S** and **F***

While **S** and **F** are fixed, the objective function can be rewritten in a simplified form to optimize **U**:

$$\text{Min}_{\mathbf{U}} \ \frac{1}{2}\sum_{i,j=1}^{n}\|\mathbf{x}_i - \mathbf{u}_i\|_2^2 + \frac{\alpha}{2}\sum_{i,j=1}^{n} s_{i,j} f_{i,j}\left\|\mathbf{u}_i - \mathbf{u}_j\right\|_2^2 \quad (20)$$

Let $h_{i,j} = s_{i,j} f_{i,j}$. Eq. (3.23) is equivalent to:

$$\text{Min}_{\mathbf{U}} \ \frac{1}{2}\|\mathbf{X} - \mathbf{U}\|_F^2 \ + \frac{\alpha}{2}\sum_{i,j=1}^{n} h_{i,j}\left\|\mathbf{u}_i - \mathbf{u}_j\right\|_2^2 \quad (21)$$

$$\Rightarrow \text{min}_{\mathbf{U}} \ \frac{1}{2}tr(\mathbf{X}^T\mathbf{X} - 2\mathbf{U}^T\mathbf{X} + \ \mathbf{U}^T\mathbf{U}) \ + \frac{\alpha}{2}tr(\mathbf{U}^T\mathbf{L}\mathbf{U}) \quad (22)$$

After conducting a derivative on Eq. (22) with respect to **U**, we get Eq. (23).

$$\Rightarrow \frac{1}{2}(-2\mathbf{X} + 2\mathbf{U}) \ + \frac{\alpha}{2}(\mathbf{L}\mathbf{U} \ + \ \mathbf{L}^T\mathbf{U}) = 0 \quad (23)$$

$$\Rightarrow \mathbf{U} = (\mathrm{I} + \alpha\mathbf{L})^{-1}\mathbf{X} \quad (24)$$

D. Convergence Analysis

In this section, we prove the convergence of the proposed KIS clustering algorithm in order to prove the proposed algorithm can reach an optimal solution, so we apply Theorem 1.

Theorem 1. KIS clustering algorithm decreases the objective function value of Eq. (7) until it converges.

Proof.

By denoting $\mathbf{F}^{(t)}$, $\mathbf{S}^{(t)}$, and $\mathbf{U}^{(t)}$, the results of the t-th iteration of **F**, **S**, and **U** respectively, we further denote the objective function value of Eq. (7) in the t-th iteration as $\mathcal{L}(\mathbf{F}^{(t)}, \mathbf{S}^{(t)}, \mathbf{U}^{(t)})$.

According to Eq. (12), **F** has a closed-form solution, thus we have the following inequality:

$$\mathcal{L}(\mathbf{F}^{(t)}, \mathbf{S}^{(t)}, \mathbf{U}^{(t)}) \geq \mathcal{L}(\mathbf{F}^{(t+1)}, \mathbf{S}^{(t)}, \mathbf{U}^{(t)}) \quad (25)$$

According to Eq. (19), **S** has a closed-form solution, thus we have the following inequality:

$$\mathcal{L}(\mathbf{F}^{(t+1)}, \mathbf{S}^{(t)}, \mathbf{U}^{(t)}) \geq \mathcal{L}(\mathbf{F}^{(t+1)}, \mathbf{S}^{(t+1)}, \mathbf{U}^{(t)}) \quad (26)$$

According to Eq. (24), **U** has a closed-form solution, thus we have the following inequality:

$$\mathcal{L}(\mathbf{F}^{(t+1)}, \mathbf{S}^{(t+1)}, \mathbf{U}^{(t)}) \geq \mathcal{L}(\mathbf{F}^{(t+1)}, \mathbf{S}^{(t+1)}, \mathbf{U}^{(t+1)}) \quad (27)$$

Finally, based on above three inequalities, we get

$$\mathcal{L}(\mathbf{F}^{(t)}, \mathbf{S}^{(t)}, \mathbf{U}^{(t)}) \geq \mathcal{L}(\mathbf{F}^{(t+1)}, \mathbf{S}^{(t+1)}, \mathbf{U}^{(t+1)}) \quad (28)$$

Equation. (28) indicates that the objective function value in Eq. (7) decreases after each iteration of Algorithm 1. This concludes the proof of Theorem 1.

IV. EXPERIMENTS

In this section, we evaluated the performance of the proposed K-Initialization-Similarity (KIS) algorithm, by comparing it with two benchmark algorithms on ten real UCI data sets, in terms of evaluation metric Accuracy.

A. Data Sets

We used ten UCI data sets in the experiments [30] including the standard data sets for email spam. wine quality, website fishing, and chess game data sets, etc. The details are summarized in Table II.

B. Comparison Algorithms

The five comparison algorithms are summarized below:

- K-means clustering algorithm randomly initializes the cluster center, then (re)assigns data points to their nearest cluster center and recalculates cluster centers iteratively until converge.
- Spectral clustering algorithm constructs the similarity matrix, and then defines the feature vectors. Finally, it runs K-means clustering algorithm.
- ROCK clustering algorithm randomly selects a number of data points from the original data and uses the number of links as the similarity measure.
- LKG constructs graph and low-rank kernel matrix by exploiting the similarity of the kernel matrix and an optimal kernel from the neighboring candidate kernels. The graph and kernel are iteratively enhanced by each other.
- U-SPEC constructs a sparse affinity sub-matrix by using a hybrid representative selection strategy and a K-nearest representatives approximation method.

C. Evaluation Measure

To assess the performance of the proposed algorithms with related algorithms, we adopted accuracy (ACC) which is a popular evaluation metric for clustering algorithms. ACC measures the percentage of samples correctly clustered [31].

TABLE II
DESCRIPTION OF TEN BENCHMARK DATA SETS

Dataset	Sample	Feature	Class
Isolet	7797	617	2
SpamBase	4601	57	2
Chess	3196	36	2
Banknote	1372	5	2
Diabetes	1151	19	2
Yeast	1484	8	10
Website	1353	9	2
Wine	1599	11	6

The definition of ACC is given below.

$$\mathrm{ACC} = \frac{N_{correct}}{N} \quad (29)$$

where $N_{correct}$ represents the number of correct clustered samples, and N represents total number of samples.

To rank the performance of different algorithms, we used dense ranking which the highest accuracy rate receives number 1, and the next accuracy rate receives the immediately following ranking number. Same accuracy rates receive the same ranking number. Thus if A ranks ahead of B and C (which compare equal) which are both ranked ahead of D, then A gets ranking number 1 ("first"), B gets ranking number 2 ("joint second"), C also gets ranking number 2 ("joint second") and D gets ranking number 3 ("Third").

D. Experiment Setup

In the experiments, first, we tested the robustness of the proposed KIS clustering algorithm by comparing it with K-means, Spectral, ROCK, LKG, and U-SPEC clustering algorithms using real data sets in terms of evaluation metric widely used for clustering research. Second, we investigated the parameters' sensitivity of the proposed KIS clustering algorithm (i.e. α and β in Eq. (7)) via varying their values to observe the variations of clustering performance. Third, we demonstrated the convergence of Algorithm 1 by checking the value of the proposed objective function Eq. (7) via iteration times.

E. Experimental Results Analysis

The performances of all algorithms are listed in Table III, which shows that the KIS algorithm achieved the best overall performance on each of the eight data sets in terms of ACC. More specifically, on the average ACC results of all eight data sets, the KIS algorithm increased it by 33.42%, 28.03%, 26.58%, 30.25%, and 25.93% respectively, compared to K-means clustering result, Spectral, ROCK, U-SPEC, and LPG. Other observations are listed below.

First, KIS, LKG, U-SPCE, and Spectral clustering algorithm outperformed K-means clustering algorithm. This implied that constructing the graph or learning a new representation of original data points improves the clustering performance. The reason could be that original data generally contains some noise or redundant information, which is often true in real data set and the noise and redundancy may corrupt the performance of clustering methods. In contrast, the non-prototype graph-based algorithms construct the new representation to conduct clustering, which can relieve the affection of noise and redundancy from original data, so the clustering performance can be improved.

Second, clustering algorithms using adaptive similarity measure, e.g. KIS clustering algorithm, performed better than nonadaptive clustering algorithms, e.g. K-means, Spectral, ROCK, U-SPCE that use the fixed similarity measurement to measure the similarity, our KIS employed an adaptive learning strategy to dynamically update the similarity matrix. In this way, our KIS can more accurately capture the intrinsic correlation of original data. This explains why our KIS easily outputs better clustering results than nonadaptive similarity clustering algorithms. This proves that the adaptive learning similarity leads to optimal clustering results, whereas the nonadaptive similarity measure clustering algorithms achieves sub-optimal results.

Third, the proposed KIS clustering algorithm use the unified framework to simultaneously address the major issues of clustering algorithms. Addressing these issues in a unified way achieves one global goal leading to optimal clustering results, whereas the multi-stage clustering algorithms with separate goals in each stage achieve sub-optimal results. When a clustering algorithm addresses those problems separately, it is easily to be trapped into the sub-optimal results, for example, even if a best initial value or the best similarity matrix is found, the final optimal results may not be obtained. Because the results of the previous steps are not obtained according to the requirements of the final step.

F. Parameters' Sensitivity

We varied parameters α and β in the range of $[10^{-3}, \ldots 10^{3}]$, and recorded the values of ACC of eight data sets clustering results for KIS clustering algorithm in Fig. 1. The parameter α is used to tune the auxiliary variable F. The parameter β is used to tradeoff the importance of similarity matrix S.

The different data sets needed different ranges of parameters to achieve the best performance. For example, KIS clustering algorithm achieved the best ACC (85.79%) on data set Isolet when both parameters α is 10^{2} and β were 10^{3}. But for the data set Wine, KIS clustering algorithm achieved the best ACC (92.94%) when both $\beta = 10^{-3}$ and $\alpha = 10^{-3}$. This indicated that KIS clustering algorithm was data driven.

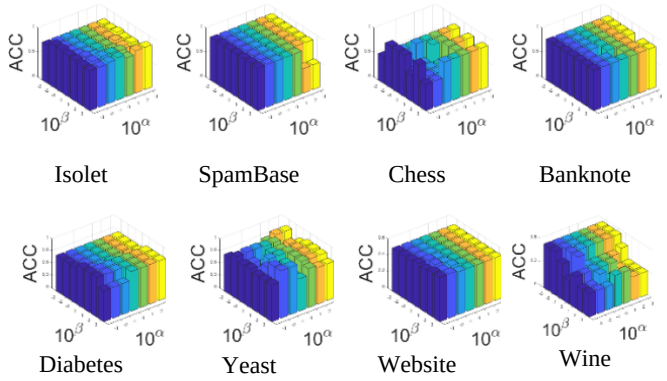

Fig. 1. ACC results of KIS algorithm with respect to different parameter settings

G.Convergence

Fig. 2 showed the trend of objective values generated by the proposed algorithm 1 with respect to iterations. The convergence curve indicates the change of the objective function value during the iteration process. From Fig. 2, we can see that the algorithm 1 monotonically decreased the objective function value until it converged, when applying it to optimize the proposed objective function in Eq. (7). That means that the value of the objective function stop changing or only change in a small range e.g. $\left|obj_{(t+1)} - obj_{(t)}\right| / obj_{(t)} \leq 10^{-9}$, where $obj_{(t)}$ represents the objection function value of Eq. (7) after the t-th iteration. In our proposed optimization algorithm, we have employed an alternating optimization strategy to optimize our objective function, i.e., iteratively updating each parameter until the algorithm converges. Thus, the optimal solution can be worked out by multiple iterations until the demand of minimizing the objective values is satisfied, which means the objective values decline to stable, as shown as the convergence lines. It is worth noting that the convergence rate of the algorithm 1 was relatively fast, converging to the optimal value within 20 iterations on all the data sets used. In other words, we can complete the optimization of our model in a fast speed.

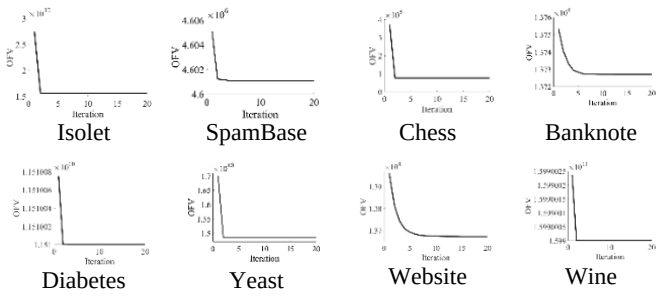

Fig. 2. Objective function values (OFVs) versus iterations for KIS algorithm

V. Conclusion

In this research we have proposed a new algorithm named K-Initialization-Similarity (KIS) which aims to solving the cluster number K determination, initialization, similarity measure issues of K-means clustering algorithm in a unified way. Specifically, we fixed the initialization by using the sum-of-norms regularization which outputted the new representation of original data points. The similarity matrix learning is based on the data distribution. The robust loss function is applied to automatically generate the cluster number *K*. The optimal performance is achieved when the separated issues are solved in a unified way. Experiment results on eight real-world benchmark data sets show that KIS outperforms the comparison clustering algorithms in terms of accuracy (ACC), the popular evaluation metric for clustering algorithm.

Although the proposed KIS clustering algorithm achieved good clustering results, we haven't considered imbalanced data sets. Hence, future research needs to improve our KIS clustering algorithm to automatically determine the clustering number K, fix the initialization, learn similarity in a unified way and have capability of handling imbalanced data.

TABLE III
ACC results of KIS algorithm on ten benchmark data sets (The highest score of each evaluation metric for each data set is highlighted in bold font)

Datasets	K-means	Spectral	ROCK	U-SPEC	LKG	KIS
Isolet	0.5065	0.5067	0.5103	0.5910	0.5410	**0.8579**
SpamBase	0.5915	0.5965	0.5947	0.6001	0.6062	**0.8660**
Chess	0.4959	0.5976	0.5207	0.5864	0.6677	**0.8403**
Banknote	0.4776	0.5904	0.4439	0.605	0.6276	**0.8216**
Diabetes	0.5130	0.5130	0.5317	0.530	0.5317	**0.6419**
Yeast	0.2353	0.3134	0.3383	0.2709	0.3127	**0.7663**
Website	0.4808	0.5212	0.5203	0.5795	0.5203	**0.5824**
Wine	0.3309	0.424	0.4259	0.4165	0.424	**0.9294**
Rank	5	4	4	3	2	**1**

References

[1] Hartigan, J.A. and M.A. Wong, *Algorithm AS 136: A k-means clustering algorithm.* Journal of the Royal Statistical Society. Series C (Applied Statistics), 1979. 28(1): p. 100-108.

[2] Jeong, Y., et al., K-means data clustering with memristor networks. Nano letters, 2018. 18(7), p. 4447-4453.

[3] Femi, P.S. and S.G. Vaidyanathan. Comparative Study of Outlier Detection Approaches. in ICIRCA. 2018. IEEE. p. 366-371.

[4] Buczkowska, S., N. Coulombel, and M. de Lapparent, *A comparison of euclidean distance, travel times, and network distances in location choice mixture models.* Networks and spatial economics, 2019: p. 1-34.

[5] Doad, P.K. and M.B. Mahip, *Survey on Clustering Algorithm & Diagnosing Unsupervised Anomalies for Network Security.* International Journal of Current Engineering and Technology ISSN, 2013: p. 2277-410.

[6] Saradha, T.M.P.D.A., *An Improved K-means Cluster algorithm using Map Reduce Techniques to mining of inter and intra cluster datain Big Data analytics.* International Journal of Pure and Applied Mathematics, 2018. **119**(7): p. 679-690.

[7] Shah, S.A. and V. Koltun, *Robust continuous clustering.* Proceedings of the National Academy of Sciences, 2017. **114**(37): p. 9814-9819.

[8] Lakshmi, M.A., G.V. Daniel, and D.S. Rao, *Initial Centroids for K-Means Using Nearest Neighbors and Feature Means*, in *Soft Computing and Signal Processing*. 2019, Springer. p. 27-34.

[9] Motwani, M., N. Arora, and A. Gupta, *A Study on Initial Centroids Selection for Partitional Clustering Algorithms*, in *Software Engineering*. 2019, Springer. p. 211-220.

[10] Duan, Y., Q. Liu, and S. Xia. *An improved initialization center k-means clustering algorithm based on distance and density*. in *AIP*, https://doi.org/10.1063/1.5033710

[11] Yan, Q., et al., *A discriminated similarity matrix construction based on sparse subspace clustering algorithm for hyperspectral imagery.* Cognitive Systems Research, 2019. **53**: p. 98-110.

[12] Bian, Z., H. Ishibuchi, and S. Wang, *Joint Learning of Spectral Clustering Structure and Fuzzy Similarity Matrix of Data.* IEEE Transactions on Fuzzy Systems, 2019. **27**(1): p. 31-44.

[13] Rong, H., et al., *A novel subgraph K^+-isomorphism method in social network based on graph similarity detection.* Soft Computing, 2018. **22**(8): p. 2583-2601.

[14] Fränti, P. and S. Sieranoja, *How much can k-means be improved by using better initialization and repeats?* Pattern Recognition, 2019. **93**: p. 95-112.

[15] Xu, D. and Y. Tian, *A comprehensive survey of clustering algorithms.* Annals of Data Science, 2015. **2**(2): p. 165-193.

[16] Song, J., F. Li, and R. Li. *Improved K-means Algorithm Based on Threshold Value Radius*. in *IOP Conference Series: Earth and Environmental Science*. 2020. IOP Publishing. doi:10.1088/1755-1315/428/1/012001

[17] Saraswathi, S. and M.I. Sheela, *A comparative study of various clustering algorithms in data mining.* International Journal of Computer Science and Mobile Computing, 2014. **11**(11): p. 422-428.

[18] Rezaei, M., *Improving a Centroid-Based Clustering by Using Suitable Centroids from Another Clustering.* Journal of Classification, 2019: p. 1-14.

[19] Lu, Y., et al., *A Tabu Search based clustering algorithm and its parallel implementation on Spark.* Applied Soft Computing, 2018. **63**: p. 97-109.

[20] Guha, S., R. Rastogi, and K. Shim, *ROCK: A robust clustering algorithm for categor*ical attributes. Information systems, 2000. 25(5): p. 345-366.

[21] Kang, Z., et al., Robust Graph Learning from Noisy Data. IEEE transactions on cybernetics, 2020. 50 (5): p. 1833-1843.
[22] Togninalli, M., et al. Wasserstein weisfeiler-lehman graph kern*els*. in *NIPS*. 2019. p. 6436-6446.
[23] Zhu, X., et al., *Low-rank sparse subspace for spectral clustering.* IEEE Transactions on Knowledge and Data Engineering, 2019. **31**(8): p. 1532-1543.
[24] Liu, G., et al., *Robust recovery of subspace structures by low-rank representation.* IEEE transactions on pattern analysis and machine intelligence, 2013. **35**(1): p. 171-184.
[25] Kang, Z., et al., *Low-rank kernel learning for graph-based clustering.* Knowledge-Based Systems, 2019. **163**: p. 510-517.
[26] Zheng, W., et al., *Unsupervised feature selection by self-paced learning regularization.* Pattern Recognition Letters, 2018: p. 438-446
[27] Geman, S. and D.E. McClure, *Statistical Methods for Tomographic Image Reconstruction.* Bulletin of the International statistical Institute, 1987. **52**(4): p. 5-21.
[28] Nikolova, M. and R.H. Chan, *The equivalence of half-quadratic minimization and the gradient linearization iteration.* IEEE Transactions on Image Processing, 2007. **16**(6): p. 1623-1627.
[29] Voloshinov, V.V., *A generalization of the Karush–Kuhn–Tucker theorem for approximate solutions of mathematical programming problems based on quadratic approximation.* Computational Mathematics and Mathematical Physics, 2018. **58**(3): p. 364-377.
[30] Dua, D. and C. Graff, *UCI Machine Learning Repository, University of California, Irvine, School of Information and Computer Sciences. 2019*. 2019.
[31] Zhu, X., et al., *One-step multi-view spectral clustering.* IEEE Transactions on Knowledge and Data Engineering, 2018. **31**(10): p.2022-2034.