Multimedia Tools and Applications (2019) 78:33279–33296
https://doi.org/10.1007/s11042-019-7663-8

Initialization-similarity clustering algorithm

Tong Liu[1,2] · Jingting Zhu[2] · Jukai Zhou[2] · YongXin Zhu[3] · Xiaofeng Zhu[1,2]

Received: 6 March 2019 / Revised: 28 March 2019 / Accepted: 16 April 2019 /
Published online: 7 May 2019

Abstract
Classic *k*-means clustering algorithm randomly selects centroids for initialization to possibly output unstable clustering results. Moreover, random initialization makes the clustering result hard to reproduce. Spectral clustering algorithm is a two-step strategy, which first generates a similarity matrix and then conducts eigenvalue decomposition on the Laplacian matrix of the similarity matrix to obtain the spectral representation. However, the goal of the first step in the spectral clustering algorithm does not guarantee the best clustering result. To address the above issues, this paper proposes an Initialization-Similarity (IS) algorithm which learns the similarity matrix and the new representation in a unified way and fixes initialization using the sum-of-norms regularization to make the clustering more robust. The experimental results on ten real-world benchmark datasets demonstrate that our IS clustering algorithm outperforms the comparison clustering algorithms in terms of three evaluation metrics for clustering algorithm including accuracy (ACC), normalized mutual information (NMI), and Purity.

Keywords *k*-means clustering · Spectral clustering · Initialization · Similarity

1 Introduction

As an unsupervised learning technique, clustering is designed to divide all the samples into subsets with the goal to maximize the intra-subset similarity and inter-subset dissimilarity [32, 50, 58]. Clustering has been widely applied in biology, psychology, marketing, medicine, etc. [5, 21, 42, 46].

Clustering algorithms can be generally classified into two categories: non-graph-based approaches [60] and graph-based approaches [44], based on if the clustering algorithm

✉ Xiaofeng Zhu
S.Zhu@massey.ac.nz

[1] GuangXi Key Lab of Multi-Source Information Mining and Security, GuangXi Normal University, Guilin 541004, China

[2] School of Natural & Computational Sciences, Massey University, Auckland, New Zealand

[3] Hebei GEO University, Shijiazhuang 050000, China

constructs the similarity matrix. A non-graph-based approach conducts clustering directly on the original data without constructing any graph such as a similarity matrix to measure the similarity among sample points. The examples of non-graph-based algorithms include *k*-means clustering algorithm [30], locality sensitive hashing based clustering [1] and mean shift [9], etc. A graph-based approach first constructs a graph and then applies the clustering algorithm to partition the graph, including spectral clustering algorithm [35], k^+-isomorphism method [39], graph clustering framework based on potential game optimization [7], bag of visual graphs [44], and low-rank kernel learning for graph-based clustering [22], etc.

K-means clustering algorithm is a benchmarked and widely used non-graph-based clustering algorithm due to its simplicity and mathematical tractability [41, 61]. Specifically, *k*-means clustering algorithm first conducts initialization via randomly selecting k samples as the k centroids, and then assigns each sample to its nearest centroid according to a similarity measurement (e.g., Euclidean distance). After this, *k*-means clustering algorithm updates the k centroids followed by assigning each data to a cluster until the algorithm achieves convergence [19].

The result of *k*-means clustering algorithm depends on the initial guess of centroids. Randomly choosing the cluster centroid may not lead to a fruitful result. It is also hard to reproduce the results. The result of *k*-means clustering algorithm also depends on the similarity measure. Euclidean distance is often used in *k*-means clustering algorithm to determine the similarity or calculate the distance between samples. Euclidean distance measures unequally weighted underlying factors but does not account for factors such as cluster sizes, dependent features or density [12, 45]. *K*-means clustering algorithm is not good to indistinct or not well-separated datasets [12].

Many literature have solved the initialization problem of *k*-means clustering algorithm [11, 14, 25, 27, 33, 42]. For example, Duan et al. developed calculating the density to select the initial centroids [14]. Lakshmi et al. proposed to use nearest neighbors and feature means to decide the initial centroids [25]. Meanwhile, many researches addressed the similarity problem of *k*-means clustering algorithm [4, 34, 37, 39, 40, 54]. Cosine-Euclidean similarity matrix (CE) employs the cosine similarity of spectral information and classical Euclidean distance to construct a similarity matrix [54]. Low-rank representation (LRR) identifies the lowest rank representation among sample points that represent the data samples [29].

However, previous research focused on solving a part of these issues but has not focused on solving the initialization of clustering and the similarity measure in a unified framework. Fixing one of the two issues does not guarantee the best performance. Solving similarity and initialization issues of *k*-means clustering algorithm simultaneously can be considered as an improvement over the existing algorithms because it could lead to better outputs. So it is significant that our proposed clustering algorithm solves the initialization and the similarity issue simultaneously.

Our proposed Initialization-Similarity (IS) clustering algorithm aims to solve the above two issues in a unified way. Specifically, we set the initialization of the clustering using sum-of-norms (SON) regularization [28]. Moreover, the SON regularization outputs the new representation of the original samples. Our proposed IS clustering algorithm then learns the similarity matrix based on the data distribution. That is, the similarity is high if the distance of the new representation of the data points is small. Furthermore, the derived new representation is used to conduct *k*-means clustering. Finally, we employ an alternative strategy to solve the proposed objective function. Experimental results on real-world benchmark datasets demonstrate that our IS clustering algorithm outperforms the comparison clustering algorithms

in terms of three evaluation metrics for clustering algorithm including accuracy (ACC), normalized mutual information (NMI), and Purity.

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

- The fixed initialization of our IS clustering algorithm using the sum-of-norms regularization makes the clustering robust and reproduced. In contrast, the previous clustering algorithm uses randomly selected centroids initialization to conduct *k*-means clustering and then outputs unstable or varying clustering results [24].
- Previous spectral clustering algorithm uses spectral representation to replace original representation for conducting *k*-means clustering. To do this, spectral clustering algorithm first generates the similarity matrix and then conducts eigenvalue decomposition on the Laplacian matrix of the similarity matrix to obtain the spectral representation. This is obviously a two-step strategy which the goal of the first step does not guarantee the best clustering result. However, our IS clustering algorithm learns the similarity matrix and the new representation simultaneously. The performance is more promising when the two steps are combined in a unified way.
- Our experiment on ten public datasets showed that our proposed IS clustering algorithm outperforms both *k*-means clustering and spectral clustering algorithms. It implies that simultaneously addressing the two issues of *k*-means clustering algorithm is feasible and fitter.

This section has laid the background of our research inquiry. The remainder of the paper is organized as follows: Section 2 discusses the existing relevant clustering algorithms. Section 3 introduces our IS clustering algorithm. Section 4 discusses the experiments we conducted and presents the results of our experiments. The conclusions, limitations and future research direction are presented in Section 5.

2 Related work

In this section, we review the relevant clustering algorithms including non-graph-based algorithms and graph-based algorithms.

2.1 Non-graph-based algorithms

Non-graph-based algorithms conduct clustering directly on the original data. *K*-means clustering algorithm is the most famous representative of non-graph-based algorithms. However, *k*-means clustering algorithm is not suitable for a dataset with an unknown number of clusters. *K*-means clustering algorithm is also sensitive to the initialization of the centroids [52]. Furthermore, the distance measure is very challenging for *k*-means clustering algorithm [45, 59].

Other algorithms based on non-graph-based algorithms include distribution-based algorithms, hierarchy-based algorithms, and density-based algorithms, etc. Popular distribution-based algorithms include Gaussian mixture model (GMM) [38] and distribution based clustering of large spatial databases (DBCLASD) [53], etc. The

distribution-based algorithms assume that the data generated from the same distribution belongs to the same cluster. However, not all the sample has several distributions and the parameters have a strong impact on the clustering results [52]. Hierarchy-based algorithms include robust clustering using links (ROCK) [17] and clustering using representatives (CURE) [18], etc. The hierarchy-based algorithms build a hierarchical relationship among samples to conduct clustering. The hierarchy-based algorithms also need to predefine the number of clusters. Density-based algorithms include Mean-shift [9] and ordering points to identify the clustering structure (OPTICS) [2]. The density-based algorithms are based on the assumption that the samples in the high-density region belong to the same cluster. However, the results of density-based algorithms would not be good if the density of samples is not even. Moreover, density-based algorithms are also sensitive to the parameters [52].

2.2 Graph-based algorithms

Instead of conducting clustering directly on the original samples, most graph-based clustering algorithms will first construct a graph and then apply a clustering algorithm to partition the graph. A node of the graph represents a sample and the edge represents the relationship among the samples. Graph representation represents the high-order relationship among samples which is easier to interpret the complex relationship inherent in the samples than to interpret it from the original samples directly. Spectral clustering algorithm is a typical example of graph-based algorithms. In the literature of graph-based algorithms, Cosine-Euclidean algorithm employs the cosine similarity of spectral information and classical Euclidean distance to construct a similarity matrix [54]. With the assumption that pairwise similarity values between elements are normally distributed and tight groups of highly similar elements likely belong to the same cluster, connectivity kernels (CLICK) algorithm recursively partitions a weighted graph into components using minimum cut computations [43]. Some graph-based algorithms construct hypergraph to represent a set of spatial data [8, 15], while other graph-based algorithms construct coefficient vectors of two samples to analyze the similarity between two samples [51]. For example, Low-Rank Representation (LRR) identifies the subspace structures from samples and then finds the lowest rank representation among samples to represent the samples [29]. Least Squares Regression (LSR) exploits data correlation and encourages a grouping effect for subspace segmentation [31]. Smooth representation (SMR) model introduces the enforced grouping effect conditions, which explicitly enforce in the sample self-representation model [20]. Chameleon uses a graph partitioning algorithm to cluster the samples into several relatively small sub-clusters, and then finds the genuine clusters by repeatedly combining these sub-clusters [23].

Graph-based clustering algorithms improve previous non-graph-based clustering algorithms on the representation of original samples. However, current graph-based clustering algorithms use a two-stage strategy which learns the similarity matrix and the spectral representation separately. The first stage goal of learning a similarity matrix does not always match the second stage goal of achieving optimal spectral representation, and thus not guaranteed to always outperform non-graph-based clustering algorithms. Moreover, most graph-based clustering algorithms still use non-graph-based clustering algorithms in the final stage and thus do not solve the initialization issue of non-graph-based clustering algorithms.

3 Proposed algorithm

3.1 Symbols

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, and italic letters as matrices, vectors, and scalars, respectively, and also summarize the symbols used in this paper in Table 1.

3.2 *K*-means clustering algorithm

K-means clustering algorithm is one of the most famous examples of the non-graph-based algorithm due to its simplicity. *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).

$$\min_{\mathbf{H}} \sum_{i=1}^{k} \sum_{j=1}^{\mathbf{c}_i} \|\mathbf{x}_i - \mathbf{h}_j\|^2 \tag{1}$$

Where C_i is the number of sample points in the i-th cluster. k is the number of clusters, while $\mathbf{h}_j$ is the j-th centroid. $\|\mathbf{x}_i - \mathbf{h}_j\|$ is the Euclidean distance between $\mathbf{x}_i$ and $\mathbf{h}_j$.

K-means clustering algorithm can be reformulated as the formulation of nonnegative matrix factorization as follows [48]:

$$\min_{\mathbf{H},\mathbf{F}} \|\mathbf{X} - \mathbf{F}\mathbf{H}\|_{\mathrm{F}}^2 \tag{2}$$

Where $\mathbf{F} \in \mathbb{R}^{n \times k}$ is the cluster indicator matrix of $\mathbf{X} \in \mathbb{R}^{n \times k}$ and $\mathbf{H} \in \mathbb{R}^{k \times \mathrm{d}}$ is the centroid matrix.

Based on both Eq. (1) and Eq. (2), it is obvious that different initialization methods may have different effects on the clustering results [36, 55]. This implies that it is difficult to reproduce the clustering results. Moreover, Eq. (2) also shows that the outcome of the clustering objective function only depends on Euclidean distance between the sample and the centroid, while Euclidean distance does not reveal other underlying factors such as cluster sizes, shape, dependent features or density [12, 45]. Thus the similarity measurement is an issue of *k*-means algorithm (Table 2).

To address the issue of *k*-means algorithm similarity measurement, spectral clustering algorithm uses spectral representation to replace original representation. To achieve this, spectral clustering algorithm first builds a similarity matrix and conducts eigenvalue

Table 1 Description of symbols used in this paper

Symbols	Description
$\mathbf{X}$	Sample 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 $\mathbf{X}$

Table 2 The pseudo code for *k*-means clustering algorithm [19]

Input: $\mathbf{X}$ (data matrix), k (the number of clusters)
Output: k centroid and the cluster indicator of each data point
Initialization:
Random selecting k centroid $\mathbf{h}_1, \mathbf{h}_2 \ldots \mathbf{h}_k$;
Repeat:
1. Assign each sample $\mathbf{x}_i$ to nearest cluster j using Euclidian distance;
2. Recalculating the new centroid $\mathbf{h}_1, \mathbf{h}_2 \ldots \mathbf{h}_k$;
Until convergence (the cluster indicator of each data points unchanged);

decomposition on its Laplacian matrix to obtain the reduced spectral representation. The pseudo code for spectral clustering algorithm is shown in Table 3.

Obviously, spectral clustering algorithm replacing original representation with spectral representation deals the issue of similarity measurement in *k*-means clustering algorithm. However, spectral clustering algorithm separately learns the similarity matrix and the spectral representation, as knowns as a two-stage strategy, where the goal of constructing the similarity matrix in the first stage does not aim at achieving optimal spectral representation, and thus not guaranteeing to always outperform *k*-means clustering algorithm.

3.3 Initialization-similarity clustering algorithm

This paper proposes a new clustering algorithm (i.e., Initialization-Similarity (IS)) to simultaneously solve the initialization issue of *k*-means clustering algorithm and the similarity issue of spectral clustering algorithm in a unified framework. Specifically, IS clustering algorithm uses the sum-of-norms regularization to investigate the initialization issue, and jointly learns the similarity matrix and the spectral representation to overcome the issue of the two-stage strategy of spectral clustering algorithm. To achieve our goal, we form the objective function of the IS clustering algorithm as follows:

$$\min_{\mathbf{S},\mathbf{U}} \frac{1}{2}\|\mathbf{X}-\mathbf{U}\|_{\mathbf{F}}^2 + \frac{\alpha}{2}\sum_{i,j=1}^{n} s_{i,j}\rho\left(\|\mathbf{u}_i-\mathbf{u}_j\|_2\right) + \beta\|\mathbf{S}\|_2^2, s.t., \forall i, s_{i,j}\geq 0, \mathbf{s}_i^T\mathbf{e} = 1 \qquad (3)$$

Where $\mathbf{S} \in \mathbb{R}^{n\times n}$ is the similarity matrix to measure the similarity among data points, and $\mathbf{U} \in \mathbb{R}^{n\times d}$ is the new representation of $\mathbf{X}$. $\rho(\|\mathbf{u}_i - \mathbf{u}_j\|_2)$ is an implicit function, as known as robust loss function, which has been used for avoiding the effect of noise and automatically generating cluster number in robust statistics.

Eq. (3) aims at learning the new representation $\mathbf{U}$ and fixes the initialization of clustering. Moreover, Eq. (3) learns the new representation $\mathbf{U}$ as well as considers the similarity among

Table 3 The pseudo code for simple spectral clustering algorithm

Input: $\mathbf{X}\in\mathbb{R}^{n\times d}$ (data matrix), k (the number of clusters)
Output: k centroid and the cluster indicator of each data point
- Computing $\mathbf{S} \in \mathbb{R}^{n\times n}$ to measure the similarity between any data point pair;
- Computing $\mathbf{L} = \mathbf{D} - \mathbf{S}$, where $\mathbf{D} = [d_{ij}]_{n\times n}$ is a diagonal matrix and $d_{ij} = \sum_{j=1}^{n} s_{ij}$;
- Generating spectral representation using the eigenvectors and eigenvalues of $\mathbf{L}$;
- Conducting k-means clustering on the spectral representation;

sample points, i.e., the higher the similarity $s_{i,j}$ between two samples, the smaller their corresponding new representation ($\mathbf{u}_i$ and $\mathbf{u}_j$) is. Furthermore, we learn the similarity matrix $\mathbf{S}$ based on the sample distribution, i.e., iteratively updated by the updated $\mathbf{U}$. This makes the new representation reasonable.

A number of robust loss functions have been proposed for avoiding the influence of noises and outliers in robust statistics [3, 56]. In this paper, we employ the Geman-McClure function [16] as follows

$$\rho\left(\left\|\mathbf{u}_p-\mathbf{u}_q\right\|_2\right)=\frac{\mu\left\|\mathbf{u}_p-\mathbf{u}_q\right\|_2^2}{\mu+\left\|\mathbf{u}_p-\mathbf{u}_q\right\|_2^2} \tag{4}$$

Eq. (4) is often used to measure how good a prediction model does in terms of being able to predict the expected outcome. The closer the distance is, the smaller value of $\|\mathbf{u}_p-\mathbf{u}_q\|_2$ is, and the higher the similarity $\boldsymbol{s}_{p,q}$ is. With the update of other parameters in Eq. (3), the distance $\|\mathbf{u}_p-\mathbf{u}_q\|_2$ for some p, q, will be very close, or even $\mathbf{u}_p=\mathbf{u}_q$. In this case, the clustering number will be less than n. In this way, the clusters will be determined.

In robust statistics, the optimization of the robust loss function is usually difficult or inefficient. To address this, it is normal for introducing an auxiliary variable $f_{i,j}$ and a penalty item $\varphi(f_{i,j})$ [6, 26, 57], and thus Eq. (3) is equivalent to:

$$\min_{\mathbf{S},\mathbf{U},\mathbf{F}} \frac{1}{2}\sum_{i=1}^{n}\|\mathbf{x}_i-\mathbf{u}_i\|_2^2+\frac{\alpha}{2}\sum_{i,j=1}^{n}s_{i,j}\left(f_{i,j}\|\mathbf{u}_i-\mathbf{u}_j\|_2^2+\boldsymbol{\varphi}\left(f_{i,j}\right)\right)+\beta\sum_{i=1}^{n}\|\mathbf{s}_i\|_2^2 s.t., \forall i, s_{i,j}\geq 0, \boldsymbol{s}_i^T\mathbf{e}=1 \tag{5}$$

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

Algorithm 1. The pseudo code for IS clustering algorithm.

Input: $\mathbf{X}\in\mathbb{R}^{n\times d}$ (n samples)
Output: a set of k clusters

Initialization:
$\mathbf{U}=\mathbf{X}$;
Repeat:
- Update $\mathbf{F}$ using Eq. (8);
- Update $\mathbf{S}$ using Eq. (12);
- Update $\mathbf{U}$ using Eq. (16);

Until $\mathbf{U}$ converges
- Apply k-means clustering algorithm on $\mathbf{U}$;

3.4 Optimization

Eq. (5) is not jointly convex on $\mathbf{F}$, $\mathbf{U}$, and $\mathbf{S}$, but is convex on each variable while fixing the rest. To solve the Eq. (5), the alternative optimization strategy is applied. We optimize each variable while fixing the rest until our algorithm converges. The pseudo-code of our IS clustering algorithm is given in Algorithm 1.

(i) **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**:

$$\min_{\mathbf{F}} \frac{\alpha}{2}\sum_{i,j=1}^{n} s_{i,j}\left(f_{i,j}\|\mathbf{u}_i-\mathbf{u}_j\|_2^2+\mu\left(\sqrt{f_{i,j}}-1\right)^2\right) \tag{6}$$

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

$$\frac{\alpha}{2}\left(s_{i,j}f_{i,j}\left\|\mathbf{u}_i-\mathbf{u}_j\right\|_2^2+s_{i,j}\left(\mu\left(f_{i,j}-2\sqrt{f_{i,j}}+1\right)\right)\right) \tag{7}$$

By conducting a derivative on Eq. (7) with respect to $f_{i,j}$, we get

$$f_{i,j}=\left(\frac{\mu}{\mu+\left\|\mathbf{u}_i-\mathbf{u}_j\right\|_2^2}\right)^2 \tag{8}$$

(ii) **Update S while fixing U and F.** While fixing **U** and **F**, the objective function Eq. (5) with respect to **S** is:

$$\min_{\mathbf{S}} \frac{\alpha}{2}\sum_{i,j=1}^{n}\left(s_{i,j}f_{i,j}\|\mathbf{u}_i-\mathbf{u}_j\|_2^2+s_{i,j}\left(\mu\left(\sqrt{f_{i,j}}-1\right)^2\right)\right)+\beta\sum_{i=1}^{n}\|\mathbf{s}_i\|_2^2 s.t., \forall i, s_{i,j}\geq 0, \mathbf{s}_i^T\mathbf{e}=1 \tag{9}$$

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

$$\min_{\mathbf{s}_i} \frac{\alpha}{2}\sum_{j=1}^{n} s_{i,j}\left(f_{i,j}\|\mathbf{u}_i-\mathbf{u}_j\|_2^2+\mu\left(\sqrt{f_{i,j}}-1\right)^2\right)+\beta\|\mathbf{s}_i\|_2^2 s.t., \forall_i, s_{i,j}\geq 0, \mathbf{s}_i^T\mathbf{e}=1 \tag{10}$$

Let $b_{i,j}=f_{i,j}\left\|\mathbf{u}_i-\mathbf{u}_j\right\|_2^2$ and $c_{i,j}=\mu\left(\sqrt{f_{i,j}}-1\right)^2$, Eq. (10) is equivalent to:

$$\min_{\mathbf{s}_i}\left\|\mathbf{s}_i-\frac{\alpha}{4\beta}\left(\mathbf{b}_i+\mathbf{c}_i\right)\right\|_2^2, s.t., \forall i, s_{i,j}\geq 0, \mathbf{s}_i^T\mathbf{e}=1 \tag{11}$$

According to Karush-Kuhn-Tucker (KKT) [47], the optimal solution $\mathbf{s}_i$ should be

$$\mathrm{s}_{i,j}=\max\left\{-\frac{\alpha}{2\beta}\left(\mathrm{b}_{i,j}+\mathrm{c}_{i,j}\right)\right\}+\theta,0\Big\}, j=1,\ldots,n \tag{12}$$

where $\theta=\frac{1}{\rho}\sum_{j=1}^{\rho}\left(\frac{\alpha}{2\beta}\left(\mathrm{b}_{i,j}+\mathrm{c}_{i,j}\right)+1\right)$, and $\rho=\max_{j}\left\{\omega_j-\frac{1}{j}\left(\sum_{r=1}^{j}\omega_r-1\right),0\right\}$ and ω is the descending order of $\frac{\alpha}{2\beta}\left(\mathrm{b}_{i,j}+\mathrm{c}_{i,j}\right)$.

(iii) 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**:

$$\min_{\mathbf{U}} \frac{1}{2}\sum_{i,j=1}^{n}\|\mathbf{x}_i-\mathbf{u}_i\|_2^2+\frac{\alpha}{2}\sum_{i,j=1}^{n}s_{i,j}f_{i,j}\|\mathbf{u}_i-\mathbf{u}_j\|_2^2 \tag{13}$$

Let $h_{i,j}=s_{i,j}f_{i,j}$. Eq. (13) is equivalent to:

$$\min_{\mathbf{U}} \frac{1}{2}\|\mathbf{X}-\mathbf{U}\|_F^2+\frac{\alpha}{2}\sum_{i,j=1}^{n}h_{i,j}\|\mathbf{u}_i-\mathbf{u}_j\|_2^2 \tag{14}$$

After conducting a derivative on Eq. (14) with respect to **U**, we get

$$\frac{1}{2}(-2\mathbf{X}+2\mathbf{U})+\frac{\alpha}{2}\left(\mathbf{L}\mathbf{U}+\mathbf{L}^T\mathbf{U}\right)=0 \tag{15}$$

Eq. (15) is solved to find **U**:

$$\mathbf{U}=(\mathbf{I}+\alpha\mathbf{L})^{-1}\mathbf{X} \tag{16}$$

3.5 Convergence analysis

In this section, we prove the convergence of our proposed IS clustering algorithm in order to prove our proposed algorithm can reach at least a locally optimal solution, so we use Theorem 1.

Theorem 1 IS clustering algorithm decreases the objective function value of Eq. (5) until it converges.

Proof By denoting $\mathbf{F}^{(t)}$, $\mathbf{S}^{(t)}$, and $\mathbf{U}^{(t)}$, respectively, are the results of the t-th iteration of **F**, **S**, and **U**, we further denote the objective function value of Eq. (5) in the t-th iteration as $\mathcal{L}\left(\mathbf{F}^{(t)},\mathbf{S}^{(t)},\mathbf{U}^{(t)}\right)$.

According to Eq. (8) in Section 3.4, **F** has a closed-form solution, thus we have the following inequality:

$$\mathcal{L}\left(\mathbf{F}^{(t)},\mathbf{S}^{(t)},\mathbf{U}^{(t)}\right)\geq\mathcal{L}\left(\mathbf{F}^{(t+1)},\mathbf{S}^{(t)},\mathbf{U}^{(t)}\right) \tag{16}$$

According to Eq. (12) in Section 3.4, **S** has a closed-form solution, thus we have the following inequality:

$$\mathcal{L}\left(\mathbf{F}^{(t+1)},\mathbf{S}^{(t)},\mathbf{U}^{(t)}\right)\geq\mathcal{L}\left(\mathbf{F}^{(t+1)},\mathbf{S}^{(t+1)},\mathbf{U}^{(t)}\right) \tag{17}$$

According to Eq. (16) in Section 3.4, **U** has a closed-form solution, thus we have the following inequality:

$$\mathcal{L}\left(\mathbf{F}^{(t+1)},\mathbf{S}^{(t+1)},\mathbf{U}^{(t)}\right)\geq\mathcal{L}\left(\mathbf{F}^{(t+1)},\mathbf{S}^{(t+1)},\mathbf{U}^{(t+1)}\right) \tag{18}$$

Finally, based on above three inequalities, we get

$$\mathcal{L}\left(\mathbf{F}^{(t)},\mathbf{S}^{(t)},\mathbf{U}^{(t)}\right)\geq\mathcal{L}\left(\mathbf{F}^{(t+1)},\mathbf{S}^{(t+1)},\mathbf{U}^{(t+1)}\right) \tag{19}$$

Eq. (19) indicates that the objective function value in Eq. (5) decreases after each iteration of Algorithm 1. This concludes the proof of Theorem 1.

4 Experiments

In this section, we evaluated the performance of our proposed Initialization-Similarity (IS) algorithm, by comparing it with two benchmark algorithms on ten real UCI datasets, in terms of three evaluation metrics.

4.1 Experiment setting

Dataset We used ten UCI datasets in our experiments, including the standard datasets for handwritten digit recognition, face datasets, and wine datasets, etc. We summarized them in Table 4.

Comparison algorithms Two comparison algorithms are classical clustering algorithms and their details were summarized below.

- *K*-means clustering algorithm (re)assigns samples to their nearest centroid and recalculates centroids iteratively with a goal to minimize the sum of distances between samples and centroid.
- Spectral clustering algorithm first forms the similarity matrix, and then calculates the first *k* eigenvectors of its Laplacian matrix to define feature vectors. Finally, it runs *k*-means clustering on these features to separate objects into *k* classes. There are different ways to calculate the Laplacian matrix. Instead of using simple Laplacian, we used normalized Laplacian $\mathbf{L} = \mathbf{D} \times \mathbf{L} \times \mathbf{D}$, which have better performance than using simple Laplacian [10].

For the above two algorithms, *k*-means clustering conducts clustering directly on the original data while spectral clustering is a two-stage based strategy, which constructs a graph first and then applies *k*-means clustering algorithm to partition the graph.

Experiment set-up In our experiments, firstly, we tested the robustness of our proposed IS clustering algorithm by comparing it with *k*-means clustering and spectral clustering algorithms using real datasets in terms of three evaluation metrics widely used for clustering research. Due to the sensitivity of *k*-means clustering to its initial centroids, we ran *k*-means clustering and spectral clustering algorithms 20 times and chose the average value as the final

Table 4 Description of ten benchmark datasets

Datasets	Samples	Dimensions	Classes
Digital	1797	64	10
MSRA	1799	256	12
Segment	2310	19	7
Solar	323	12	6
USPS	1854	256	10
USPST	2007	256	10
Waveform	5000	21	3
Wine	178	13	3
Wireless	2000	7	4
Yale	165	1024	15

result. Secondly, we investigated the parameters' sensitivity of our proposed IS clustering algorithm (i.e. α and β in Eq. (5)) via varying their values to observe the variations of clustering performance. Thirdly, we demonstrated the convergence of Algorithm 1 to solve our proposed objective function Eq. (5) via checking the iteration times when Algorithm 1 converges.

Evaluation measures To compare our IS clustering algorithm with related algorithms, we adopted three popular evaluation metrics of clustering algorithms including accuracy (ACC), normalized mutual information (NMI), and Purity [49]. ACC measures the percentage of samples correctly clustered. NMI measures the pairwise similarity between two partitions. Purity measures the percentage of each cluster containing the correctly clustered samples [13, 61]. The definitions of these three evaluation metrics are given below.

$$ACC = N_{correct}/N \tag{20}$$

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

$$NMI\ (A,B) = \frac{\sum_{i=1}^{C_A}\sum_{j=1}^{C_B} n_{ij}\log\left(n_{ij}n/n_i^A n_j^B\right)}{\sqrt{\sum_{i=1}^{C_A} n_i^A \log(n_i^A/n)\sum_{j=1}^{C_B} n_j^B \log\left(n_j^B/n\right)}} \tag{21}$$

where A, B represents two partitions of n samples into C_A and C_B clusters respectively.

$$Purity = \sum_{i=1}^{k}(S_i/n)P_i \tag{22}$$

where k represents number of clusters and n represents total number of samples. S_i represents the number of samples in the i-th cluster. P_i represents the distribution of correctly clustered sample.

4.2 Experimental results

We listed the clustering performance of all algorithms in Table 5, which showed that out IS clustering algorithm achieved the best performance on all ten datasets in terms of ACC and NMI, as well as outperformed k-means clustering algorithm on all ten datasets in terms of Purity. Our IS clustering algorithm outperformed spectral clustering algorithm on all eight datasets in terms of Purity but performed slightly worse than spectral clustering algorithm on three datasets USPT, USPST and Yale. The difference in Purity results between our IS clustering algorithm and the spectral clustering algorithm was only 1%. More specifically, our IS clustering algorithm increased ACC by 6.3% compared to k-means clustering algorithm and 3.3% compared to spectral clustering algorithm. Our IS clustering algorithm increased NMI by 4.6% compared to k-means clustering algorithm and 4.5% compared to spectral clustering algorithm. Our IS clustering algorithm increased Purity by 4.9% compared to k-means clustering algorithm and 2.9% compared to spectral clustering algorithm. Other observations were listed in the following sections.

Table 5 Performance of all algorithms on ten benchmark datasets

	ACC			NMI			Purity		
Datasets	*K*-means	Spectral	IS	*K*-means	Spectral	IS	*K*-means	Spectral	IS
Digital	0.73 ±0.06	0.77 ±0.03	**0.80**	0.73 ±0.02	0.72 ±0.01	**0.78**	0.76 ±0.04	0.78 ±0.02	**0.81**
MSRA	0.49 ±0.05	0.50 ±0.03	**0.57**	0.59 ±0.03	0.56 ±0.02	**0.63**	0.53 ±0.03	0.53 ±0.02	**0.58**
Segment	0.55 ±0.05	0.56 ±0.03	**0.63**	0.61 ±0.05	0.52 ±0.03	**0.63**	0.58 ±0.04	0.58 ±0.02	**0.64**
Solar	0.50 ±0.04	0.51 ±0.02	**0.55**	0.34 ±0.05	0.34 ±0.02	**0.42**	0.55 ±0.05	0.55 ±0.03	**0.61**
USPS	0.62 ±0.05	0.67 ±0.02	**0.70**	0.61 ±0.02	0.66 ±0.01	**0.70**	0.69 ±0.03	**0.75** ±0.02	0.74
USPST	0.66 ±0.05	0.70 ±0.02	**0.71**	0.61 ±0.01	0.66 ±0.02	**0.68**	0.71 ±0.02	**0.77** ±0.02	0.76
Waveform	0.50 ±0.00	0.51 ±0.00	**0.57**	0.36 ±0.00	0.37 ±0.00	**0.40**	0.53 ±0.00	0.51 ±0.00	**0.59**
Wine	0.65 ±0.07	0.69 ±0.02	**0.71**	0.43 ±0.01	0.42 ±0.04	**0.43**	0.69 ±0.01	0.69 ±0.02	**0.71**
Wireless	0.94 ±0.06	0.96 ±0.00	**0.97**	0.88 ±0.04	0.89 ±0.00	**0.91**	0.94 ±0.05	0.96 ±0.00	**0.97**
Yale	0.39 ±0.04	0.45 ±0.04	**0.46**	0.47 ±0.03	0.51 ±0.03	**0.51**	0.41 ±0.03	**0.47** ±0.04	0.46
Rank	3.0	2.0	**1.0**	2.4	2.4	**1.0**	2.4	1.8	**1.3**

The highest score of each evaluation metric for each dataset is highlighted in bold font

First, one-step clustering algorithm, e.g. our IS clustering algorithm, performed better than two-step clustering algorithms, e.g. spectral clustering algorithm. The reason could be that the goals of the similarity matrix learning and the new representation are the optimal clustering results, whereas the two-step clustering algorithm achieves sub-optimal results.

Second, both one-step clustering algorithm, e.g. our IS clustering algorithm and two-step clustering algorithm, e.g. spectral clustering algorithm outperformed *k*-means clustering algorithm. This implied that constructing the graph or learning a new representation of original samples improved the clustering performance.

Parameters' sensitivity We varied parameters α and β in the range of $[10^{-2}, \ldots 10^{2}]$, and recorded the values of ACC, NMI and Purity of ten datasets clustering results for our IS clustering algorithm in Figs. 1, 2 and 3.

First, different datasets needed different ranges of parameters to achieve the best performance. For example, IS clustering algorithm achieved the best ACC (97%), NMI (91%) and Purity (97%) on dataset Wireless when both parameters α and β were 10. But for the dataset Digital, IS clustering algorithm achieved the best ACC (80%), NMI (78%) and Purity (81%) when β = 100 and α =0.1. This indicated that our IS clustering algorithm was data-driven.

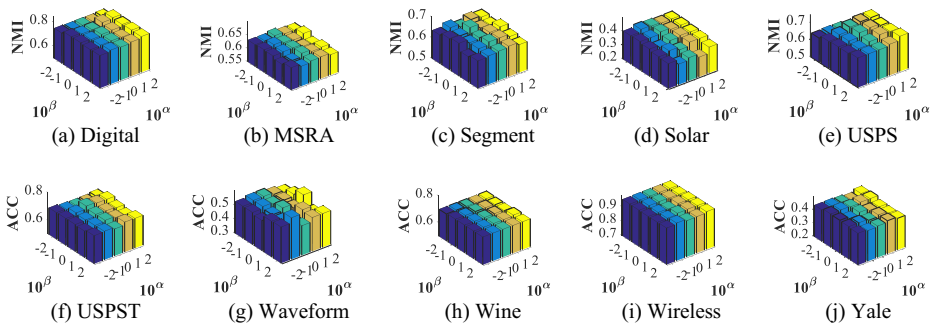


Fig. 1 ACC of our IS clustering algorithm with respect to different parameter settings

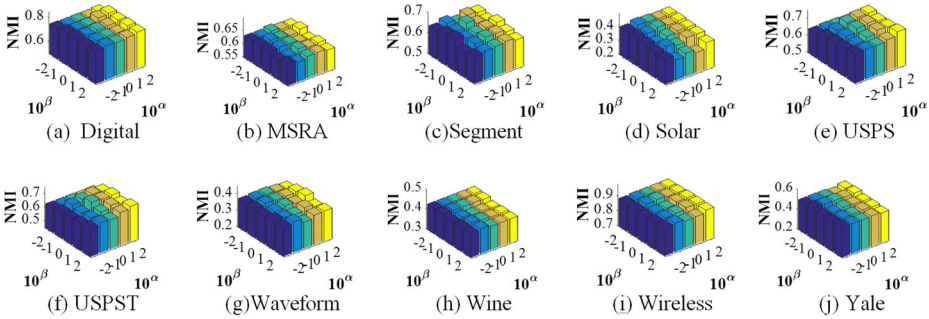


Fig. 2 NMI of our IS clustering algorithm with respect to different parameter settings

Second, the clustering ACC results had less than 3% average changes when the parameter α varied in the range of $[10^{-2}, \ldots 10^{2}]$ in eight out of ten datasets. The lowest average change was 1% (i.e., Wine and Wireless datasets) when the parameter α varied in the range of $[10^{-2}, \ldots 10^{2}]$. The biggest average change was 5% (e.g., Waveform dataset) when the parameter α varied in the range of $[10^{-2}, \ldots 10^{2}]$. This indicated that our IS clustering algorithm was not very sensitive to the parameter α.

Third, the clustering ACC results had less than 3% average changes when the parameter β varied in the range of $[10^{-2}, \ldots 10^{2}]$ in nine out of ten datasets. The lowest average change was 0 (Wine dataset) when the parameter β varied in the range $[10^{-2}, \ldots 10^{2}]$. The biggest average change was 5% (Waveform dataset) when the parameter β varied in the range of $[10^{-2}, \ldots 10^{2}]$. This indicated that our IS clustering algorithm was not very sensitive to the parameter β.

Fourth, even our IS clustering algorithm was not very sensitive on parameters α and β, the algorithm was slightly more sensitive on parameter α than it was on the parameter β.

Convergence Figure 4 showed the trend of objective values generated by our proposed algorithm 1 with respect to iterations. From Fig. 4, we can see that our algorithm 1 monotonically decreased the objective function value until it converged, when applying it to optimize the proposed objective function in Eq. (5). It is worth noting that the convergence rate of our algorithm 1 was relatively fast, converging to the optimal value within 20 iterations on all the datasets used.

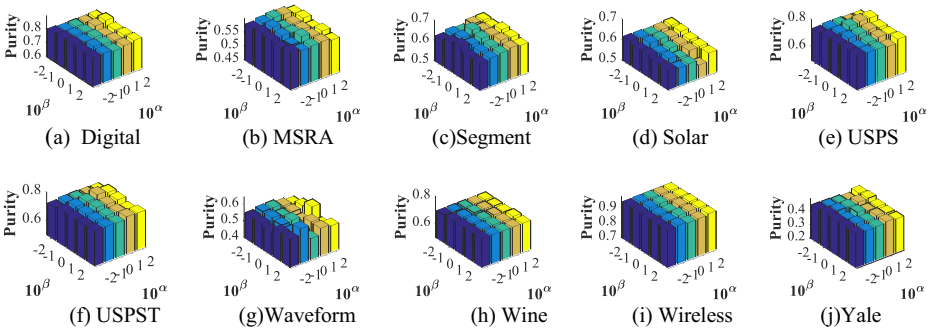


Fig. 3 Purity of our IS clustering algorithm with respect to different parameter settings

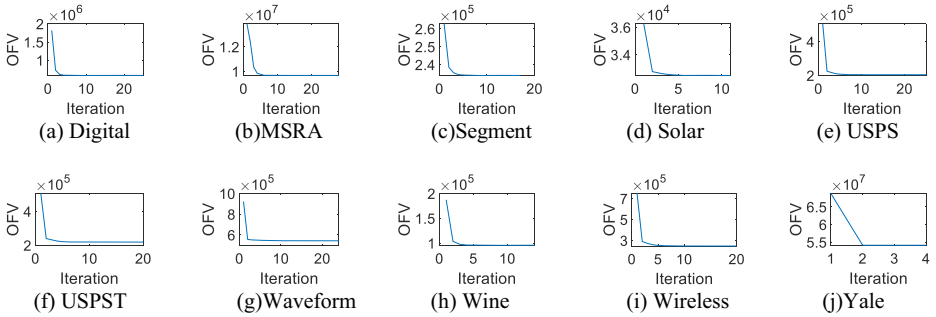


Fig. 4 Objective function values (OFVs) versus iterations

5 Conclusion

This paper has proposed a new Initialization-Similarity (IS) algorithm to solve the initialization and similarity issues in a unified way. Specifically, we fixed the initialization of the clustering using the sum-of-norms regularization which outputted the new representation of original samples. We then learned the similarity matrix and the new representation simultaneously. Finally, we conducted *k*-means clustering on the derived new representative. Extensive experimental results on real-world benchmark datasets showed that our IS clustering algorithm outperformed the related clustering algorithms. Furthermore, our IS clustering algorithm is not very parameter sensitive. The fixed initialization of our IS clustering algorithm using the sum-of-norms regularization makes the clustering robust.

Although our proposed IS clustering algorithm achieved significant clustering results, but we used *k*-means clustering in the final stage clustering. Similar to all *k*-means based clustering algorithms, this is the main limitation of our IS clustering algorithm. Hence, future research needs to develop new clustering algorithms to learn the clustering number *k,* initialization and similarity automatically in a unified way.

Funding This work was partially supported by the Research Fund of Guangxi Key Lab of Multi-source Information Mining & Security (MIMS18-M-01), the Natural Science Foundation of China (Grants No: 61876046 and 61573270); the Guangxi High Institutions Program of Introducing 100 High-Level Overseas Talents; the Strategic Research Excellence Fund at Massey University, and the Marsden Fund of New Zealand (Grant No: MAU1721).

References

1. Ahmed T, Sarma M (2018) Locality sensitive hashing based space partitioning approach for indexing multidimensional feature vectors of fingerprint image data. IET Image Process 12(6):1056–1064
2. Ankerst M, et al (1999) *OPTICS: ordering points to identify the clustering structure*. in *ACM Sigmod record*. p. 49–60
3. Barron JT (2017) *A more general robust loss function*. arXiv preprint arXiv:1701.03077
4. Bian Z, Ishibuchi H, Wang S (2019) Joint learning of spectral clustering structure and fuzzy similarity matrix of data. IEEE Trans Fuzzy Syst 27(1):31–44
5. Bin Y et al (2018) Describing video with attention-based bidirectional LSTM. IEEE transactions on cybernetics. https://doi.org/10.1109/TCYB.2018.2831447
6. Black MJ, Rangarajan A (1996) On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. Int J Comput Vis 19(1):57–91
7. Bu Z et al (2018) GLEAM: a graph clustering framework based on potential game optimization for large-scale social networks. Knowl Inf Syst 55(3):741–770

8. Cherng JS, Lo MJ (2001) *A hypergraph based clustering algorithm for spatial data sets*. in *ICDM,* p. 83–90
9. Comaniciu D, Meer P (2002) Mean shift: a robust approach toward feature space analysis. IEEE Trans Pattern Anal Mach Intell 24(5):603–619
10. Das A, Panigrahi P (2018) Normalized Laplacian spectrum of some subdivision-joins and R-joins of two regular graphs. AKCE International Journal of Graphs and Combinatorics 15(3):261–270
11. Deelers S, Auwatanamongkol S (2007) Enhancing K-means algorithm with initial cluster centers derived from data partitioning along the data axis with the highest variance. Int J Comput Sci 2(4):247–252
12. Doad PK, Mahip MB (2013) *Survey on Clustering Algorithm & Diagnosing Unsupervised Anomalies for Network Security.* International Journal of Current Engineering and Technology ISSN, p. 2277–410
13. Domeniconi C, Al-Razgan M (2009) Weighted cluster ensembles: methods and analysis. ACM Transactions on Knowledge Discovery from Data (TKDD) 2(4):17
14. Duan Y, Liu Q, Xia S (2018) *An improved initialization center k-means clustering algorithm based on distance and density* in *AIP*: 1955(1), p. 040–046
15. Estivill-Castro V, Lee I (2000) Amoeba: Hierarchical clustering based on spatial proximity using delaunay diagram. in ISSDH, p. 1–16
16. Geman S, McClure DE (1987) Statistical methods for tomographic image reconstruction. Bulletin of the International statistical Institute 52(4):5–21
17. Guha S, Rastogi R, Shim K (2000) ROCK: a robust clustering algorithm for categorical attributes. Inf Syst 25(5):345–366
18. Guha S, Rastogi R, Shim K (2001) Cure: an efficient clustering algorithm for large databases. Inf Syst 26(1):35–58
19. Hartigan JA, Wong MA (1979) Algorithm AS 136: a k-means clustering algorithm. J R Stat Soc: Ser C: Appl Stat 28(1):100–108
20. Hu H, et al (2014) *Smooth representation clustering*. in *CV PR*. p. 3834–3841
21. Jain AK (2010) Data clustering: 50 years beyond K-means. Pattern Recogn Lett 31(8):651–666
22. Kang Z et al (2019) Low-rank kernel learning for graph-based clustering. Knowl-Based Syst 163:510–517
23. Karypis G, Han E-H, Kumar V (1999) Chameleon: hierarchical clustering using dynamic modeling. Computer 32(8):68–75
24. Kuncheva LI, Vetrov DP (2006) Evaluation of stability of k-means cluster ensembles with respect to random initialization. IEEE Trans Pattern Anal Mach Intell 28(11):1798–1808
25. Lakshmi MA, Daniel GV, Rao DS (2019) *Initial Centroids for K-Means Using Nearest Neighbors and Feature Means*, in *SCSP*, p. 27–34
26. Lei C, Zhu X (2018) Unsupervised feature selection via local structure learning and sparse learning. Multimed Tools Appl 77(22):29605–29622
27. Likas A, Vlassis N, Verbeek JJ (2003) The global k-means clustering algorithm. Pattern Recogn 36(2):451–461
28. Lindsten F, Ohlsson H, Ljung L (2011) *Clustering using sum-of-norms regularization: With application to particle filter output computation*. in *SSP*, p. 201–201
29. Liu G et al (2013) Robust recovery of subspace structures by low-rank representation. IEEE Trans Pattern Anal Mach Intell 35(1):171–184
30. Lloyd S (1982) Least squares quantization in PCM. IEEE Trans Inf Theory 28(2):129–137
31. Lu CY, et al (2012) Robust and efficient subspace segmentation via least squares regression. in ECCV. p. 347–360
32. Moftah HM et al (2014) Adaptive k-means clustering algorithm for MR breast image segmentation. Neural Comput & Applic 24(7–8):1917–1928
33. Motwani M, Arora N, Gupta A (2019) *A Study on Initial Centroids Selection for Partitional Clustering Algorithms*, in *Software Engineering*. p. 211–220
34. Nie F, Wang X, Huang H (2014) *Clustering and projected clustering with adaptive neighbors*. in *SIGKDD,* p. 977–986
35. Park S, Zhao H (2018) Spectral clustering based on learning similarity matrix. Bioinformatics 34(12):2069–2076
36. Pavan KK, Rao AD, Sridhar G (2010) Single pass seed selection algorithm for k-means. J Comput Sci 6(1):60–66
37. Radhakrishna V et al (2018) A novel fuzzy similarity measure and prevalence estimation approach for similarity profiled temporal association pattern mining. Futur Gener Comput Syst 83:582–595
38. Rasmussen CE (2000) *The infinite Gaussian mixture model*. in *NIPS*, p.554–560
39. Rong H et al (2018) A novel subgraph K^+-isomorphism method in social network based on graph similarity detection. Soft Comput 22(8):2583–2601
40. Satsiou A, Vrochidis S, Kompatsiaris I (2018) A Hybrid Recommendation System Based on Density-Based Clustering. in INSCI 2018
41. Saxena A et al (2017) A review of clustering techniques and developments. Neurocomputing 267:664–681
42. Shah SA, Koltun V (2017) Robust continuous clustering. Proc Natl Acad Sci 114(37):9814–9819
43. Sharan R, Shamir R (2000) *CLICK: a clustering algorithm with applications to gene expression analysis*. in *ICISMB*. 8(307), p. 307–316
44. Silva FB et al (2018) Graph-based bag-of-words for classification. Pattern Recogn 74:266–285

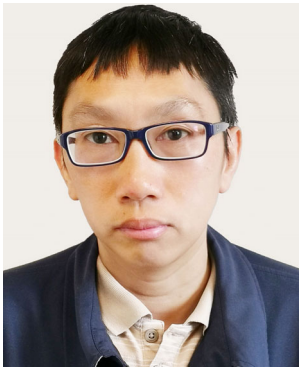
45. Singh A, A Yadav, Rana A (2013) *K-means with Three different Distance Metrics.* International Journal of Computer Applications, **67**(10)
46. Song J et al (2018) From deterministic to generative: multimodal stochastic RNNs for video captioning. IEEE transactions on neural networks and learning systems. https://doi.org/10.1109/TNNLS.2018.2851077
47. Voloshinov VV (2018) A generalization of the Karush–Kuhn–Tucker theorem for approximate solutions of mathematical programming problems based on quadratic approximation. Comput Math Math Phys 58(3): 364–377
48. Wang J, et al (2015) Fast Approximate K-Means via Cluster Closures, in MDMA. p. 373–395
49. Wang C et al (2018) Multiple kernel clustering with global and local structure alignment. IEEE Access 6: 77911–77920
50. Wong KC (2015) *A short survey on data clustering algorithms*. in *ISCMI*
51. Wu S, Feng X, Zhou W (2014) Spectral clustering of high-dimensional data exploiting sparse representation vectors. Neurocomputing 135:229–239
52. Xu D, Tian Y (2015) A comprehensive survey of clustering algorithms. Annals of Data Science 2(2):165–193
53. Xu X, et al. (1998) *A distribution-based clustering algorithm for mining in large spatial databases*. in *ICDE,* p. 324–331
54. Yan Q et al (2019) A discriminated similarity matrix construction based on sparse subspace clustering algorithm for hyperspectral imagery. Cogn Syst Res 53:98–110
55. Zahra S et al (2015) Novel centroid selection approaches for KMeans-clustering based recommender systems. Inf Sci 320:156–189
56. Zheng W et al (2018) Unsupervised feature selection by self-paced learning regularization. Pattern Recogn Lett. https://doi.org/10.1016/j.patrec.2018.06.029
57. Zheng W et al (2018) Dynamic graph learning for spectral feature selection. Multimed Tools Appl 77(22): 29739–29755
58. Zhou X et al (2018) Graph convolutional network hashing. IEEE transactions on cybernetics. https://doi.org/10.1109/TCYB.2018.2883970
59. Zhu X et al (2017) Graph PCA hashing for similarity search. IEEE Transactions on Multimedia 19(9):2033–2044
60. Zhu X et al (2018) Low-rank sparse subspace for spectral clustering. IEEE Trans Knowl Data Eng. https://doi.org/10.1109/TKDE.2018.2858782
61. Zhu X et al (2018) One-step multi-view spectral clustering. IEEE Trans Knowl Data Eng. https://doi.org/10.1109/TKDE.2018.2873378

Tong Liu is a faculty member at the School of Natural and Computational Sciences, Massey University, Auckland, New Zealand. She holds a Master of Science degree in Computer Science from Massey University, New Zealand. Her research interest includes big data, data mining, machine learning, artificial intelligence, software engineering, and application of IT in industry.

Jingting Zhu is a Ph.D. candidate at the School of Natural and Computational Sciences, Massey University, New Zealand. He holds a Master's degree in Computer Science from Kunming University of Science and Technology, China. His research interest includes data analysis, and multimedia application.

Jukai Zhou is a Master student at SNCS of Massey University, New Zealand. His research interests include data mining, machine learning and Big Data computing.

Yongxin Zhu is a Master's student at the Department of Computer Science and Technology, Hebei GEO University, China. He holds a Bachelor's Degree in Computer Science from Taiyuan Institute of Technology, China. His research interest includes machine learning, and natural language processing.



Xiaofeng Zhu is a faculty member at Massey University, New Zealand. His current research interests include large-scale multimedia retrieval, feature selection, sparse learning, data preprocess, and medical image analysis.