Information Sciences 466 (2018) 129–151

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

I-nice: A new approach for identifying the number of clusters and initial cluster centres

Md Abdul Masud[a], Joshua Zhexue Huang[a,*], Chenghao Wei[a], Jikui Wang[a], Imran Khan[b], Ming Zhong[a]

[a] College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China
[b] Shenzhen Key Laboratory of Computational Intelligence, Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China

ARTICLE INFO

Article history:
Received 7 January 2017
Revised 4 July 2018
Accepted 11 July 2018
Available online 20 July 2018

Keywords:
Clustering algorithm
Initial cluster centres
Number of clusters

ABSTRACT

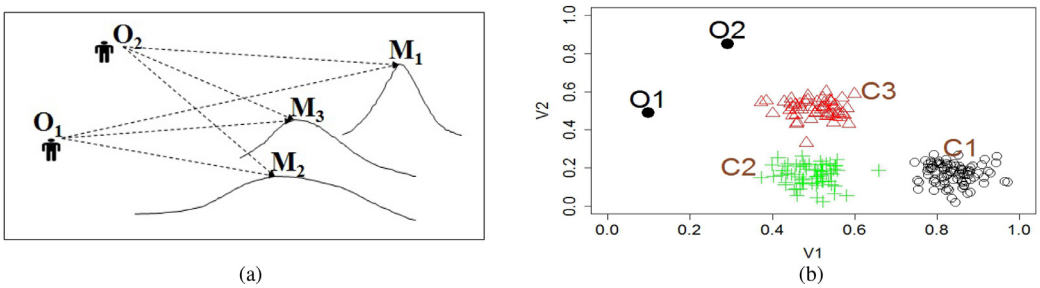
This paper proposes I-nice, which is a new method for automatically identifying the number of clusters and selecting the initial cluster centres in data. The method mimics a human being in observing peaks of mountains in field observation. The clusters in a dataset are considered as the hills in a field terrain. The distribution of distances between the observation point and the objects is computed. The distance distribution is modelled by a set of Gamma mixture models (GMMs), which are solved with the expectation-maximization (EM) algorithm. The best-fitted model is selected with an Akaike information criterion variant (AICc). In the I-niceSO algorithm, the number of components in the model is taken as the number of clusters, and the objects in each component are analysed with the *k*-nearest-neighbour method to find the initial cluster centres. For complex data with many clusters, we propose the I-niceMO algorithm, which combines the results of multiple observation points. Experimental results show that the two algorithms significantly outperformed two state-of-the-art methods (Elbow and Silhouette) in identifying the correct number of clusters in data. The results also show that I-niceMO improved the clustering accuracy and efficiency of the *k*-means clustering process.

1. Introduction

Clustering is one of the key techniques in data analysis. It is the process of dividing the data of objects into a set of clusters in which the objects in the same clusters are close to each other according to a similarity measure, whereas the objects in different clusters are far from each other. One problem in cluster analysis is that the number of clusters in the data to be analysed must be known in advance because many clustering algorithms require the number of clusters as an input parameter to run the algorithms. However, the number of clusters that exist in real data is usually unknown. Therefore, a number is often guessed in practical cluster analysis, which often results in unsatisfactory results. Although several methods for estimating the number of clusters in data have been developed [13,21,42,45,47], they either produce incorrect results or are difficult to use in real applications. Therefore, finding the correct number of clusters from real data remains a classical problem in cluster analysis. It is also an active research topic.

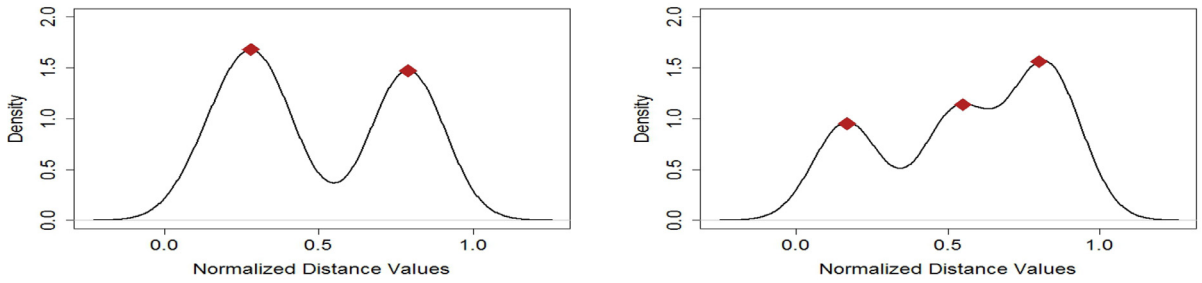
* Corresponding author.
E-mail addresses: masud@szu.edu.cn (M.A. Masud), zx.huang@szu.edu.cn (J.Z. Huang), chenghao.wei@yahoo.com (C. Wei), wjkweb@163.com (J. Wang), imran.khan@sustc.edu.cn (I. Khan), mingz@szu.edu.cn (M. Zhong).

https://doi.org/10.1016/j.ins.2018.07.034
0020-0255/



(a) (b)

Fig. 1. (a) Two observers observe three peaks of hills. (b) Two observation points are used to observe three clusters.



(a) (b)

Fig. 2. (a) Distance distribution with two peaks related to observation point O1. (b) Distance distribution with three peaks related to observation point O2.

In this paper, we propose an innovative approach to identifying the number of clusters in high-dimensional data. We consider a dataset as a terrain in which clusters are hills. We assign an observer to the terrain to observe and count the peaks of hills, which correspond to the dense regions of clusters and reflect the number of clusters in the data. Fig. 1(a) shows an example of the observation process in which three hills are situated at different distances from two observers who observe the number of peaks of hills in the terrain. We can mimic this scenario in counting the number of clusters in high-dimensional data. Fig. 1(b) illustrates an example of two-dimensional data that contain 3 clusters, which are denoted as C1, C2 and C3. The two black dots, which are denoted as O1 and O2, are two observation points. Whether an observer can see all peaks of hills depends on the location of the observer in the terrain because a hill may block the sight of the observer, thereby making it impossible to see the hills that are behind the front hill. Therefore, the location of the observer is important in finding all peaks.

To observe the dense regions of clusters in data, we calculate the distances of all points to the observation point to transform the high-dimensional data into one-dimensional distance data. The one-dimensional distance distribution carries the information of clusters in the original high-dimensional data. Therefore, it can be used to find the number of clusters. Fig. 2 shows two distributions of distances from observation points O1 and O2 in Fig. 1(b). The distance distribution of observation point O1 has two peaks, whereas the distance distribution of observation point O2 has three peaks, which are equal to the number of clusters in the data. According to Fig. 1(b), clusters C2 and C3 have the same distance to observation point O1. Thus, the peaks of these two clusters are merged into the left peak of Fig. 2(a). Therefore, only two clusters can be observed from this distance distribution. However, the three clusters have different distances to observation point O2. Thus, all three clusters are observed in the distance distribution of Fig. 2(b). Similarly, we can allocate multiple observers in the data space, and the largest number of peaks observed is taken as the number of clusters in the data. This is the method for finding the number of clusters that correspond to one observer. Alternatively, we can use the one-dimensional distance distributions from multiple observation points and combine the numbers of clusters found from the distributions. This method is useful for data with many clusters.

To automatically find the number of peaks from a distance distribution, we model the distance data with a Gamma mixture model (GMM) and use the expectation-maximization (EM) algorithm to solve the GMM. Since the number of components in the GMM is not known, we build multiple GMMs with different numbers of components and solve these models. Then, we use the second-order variant of the Akaike information criterion (AICc) to select the best-fitted model whose number of components is the observed number of peaks in the distance distribution. This approach is called I-nice, which is an abbreviation of **I**dentifying the **n**umber of clusters and **i**nitial cluster **ce**ntres.

In this paper, we propose the I-niceSO (**I**dentifying the **n**umber of clusters and **i**nitial cluster **ce**nitres with a **S**ingle **O**bservation) clustering algorithm, which uses the I-nice method to automatically find the number of clusters and the initial clustering centres in data for *k*-means clustering. In I-nice, multiple observation points are used to compute the distance distributions, and multiple best-fitted GMMs are selected from these distance distributions. The model with the largest

number of components, which is used as the number of clusters in the data, is selected as the final model. From the final GMM, I-niceSO can identify the objects in the peak of each component, each of which corresponds to a cluster in the data. By analysing these objects near the peak, an object in the dense region of the cluster is selected as the initial cluster centre. These selected initial cluster centres are used to improve the clustering performance of the *k*-means-type algorithm.

For complex data with many clusters, it will be difficult to use the final best-fitted GMM obtained by I-niceSO to identify the correct number of clusters in the data. For such data, we propose I-niceMO (**I**dentifying the **n**umber of clusters and **i**nitial cluster **ce**ntres with **M**ultiple **O**bservations), which is an extension of I-niceSO that makes use of multiple best-fitted GMMs to jointly identify the number of clusters and initial cluster centres. First, I-niceMO selects all best-fitted models from I-niceSO. For each model, it obtains the clusters of the objects identified by each component of the GMM and maps the objects of clusters to the original data space. Then, the dense regions of the clusters are identified with the *k*-nearest-neighbour method, and the objects in the densest regions are selected as the initial cluster centres from this model. After the initial cluster centre objects are selected from all best-fitted GMMs, the final set of initial cluster centres is obtained according to similarity analysis of these centre objects. These initial centres are estimated as the number of clusters in the data and used in *k*-means clustering to improve the clustering result.

Both I-niceSO and I-niceMO are parameter-free *k*-means-type clustering algorithms. Because the number of clusters is identified from one-dimensional distance data and the original data are clustered with *k*-means, the I-nice approach is efficient in clustering big data and easy to use in practice.

We conducted a series of experiments on the two I-nice algorithms with both synthetic and real-world datasets. We compared the results of I-nice with those of the Elbow and Silhouette methods, which are two popular methods for finding the number of clusters. The comparison results show that the I-nice algorithms significantly outperformed Elbow and Silhouette in finding the correct number of clusters for both synthetic and real-world data. The initial cluster centres selected by the I-nice methods also improved the clustering accuracy and efficiency of the *k*-means algorithm because the number of iterations was reduced significantly due to the high quality of the initial cluster centres.

The remainder of this paper is organized as follows: First, we present a brief overview of related work in Section 2. We introduce an innovative I-nice method and present the steps of I-niceSO, which is an I-nice algorithm that uses a single observation point, in Section 3. We present I-niceMO, which is an I-nice algorithm that uses multiple observation points, in Section 4. We present the experimental results of the I-nice method and other methods in Section 5. Finally, we discuss the I-nice approach and the conclusions of this work in Section 6.

2. Related works

Finding the number of clusters in data is a well-known classical problem in cluster analysis [6,13,22,26,33,41,42]. Determining the number of clusters in data is important because many clustering algorithms, such as the *k*-means and model-based clustering algorithms, take the number of clusters as an input parameter. It is also an important factor that affects the clustering results. However, there is still no unique method that can produce satisfactory results for all cases.

Hierarchical clustering methods determine the final number of clusters after clustering the data. A validation index is calculated from the clustering results for each candidate number of clusters. The final number of clusters is determined by the index value, which indicates the best clustering result among the alternative results [3,29,32,38].

In model-based clustering methods, as the number of clusters is an input parameter to the algorithm, a range of numbers of clusters is often used to build multiple models, and a model selection method is used to select the best-fitted model, which gives the final number of clusters in the data [6,18,19]. A method for selecting the number of components for a finite mixture model was proposed in [17].

Elbow and Silhouette are two popular methods frequently used to determine the number of clusters in data. The Elbow method calculates the average sum of within-cluster distances between objects and the cluster centres [40]. This value decreases as the number of clusters increases. When this value is plotted against the number of clusters, the number of clusters at the elbow (knee) location of the curve indicates the correct number of clusters in the data because the average sum of within-cluster distances decreases slowly as the number of clusters increases beyond the elbow number. The main problem with this method is that the curves of the average sum of within-cluster distances against the number of clusters in many datasets often do not have an elbow shape. Therefore, the correct number of clusters is not identifiable from the curves [27].

In the Silhouette method [35], given a clustering result, for each object, an index value is computed, which is the relative difference between the within-cluster distances of objects to this object and the distances of objects in other clusters to this object. The clustering result is evaluated based on the average of the index values of all objects. The larger the Silhouette index value is, the better the clustering results. The number of clusters is determined by the best clustering result, which is selected based on the Silhouette index. Silhouette index methods for cluster validation have been widely investigated [3,13,26]. However, validation-index-based solutions are not likely to provide consistent results across different clustering algorithms and data structures [7].

A recent study has proposed a density-based method for discovering the number of clusters from data [34]. This method is based on the assumption that high-density regions represent clusters. This assumption is similar to the assumption in this paper. However, the method in [34] requires manual adjustment of parameters to determine the final result. The method proposed in this paper models the whole process automatically.

After determining the number of clusters in the data, the selection of the initial cluster centres is an important step in the widely used k-means-type algorithms. There are many studies on methods for this purpose [8,14,25,28,36,47]. The mountain method [48] has been approximated to estimate the cluster centres. This method performs discretization by defining grids on the object space. The intersections of grid lines are considered as node points, which are used to compute the value of the mountain function from objects. The cluster centres are restricted to the node points. The mountain method was modified by the authors of [49], in which the correlation self-comparison algorithm is used to acquire a modified mountain function. To execute the correlation self-comparison procedure, the kernel-type density estimate of each object is computed on all other objects. The modified revised mountain function is computed to select the initial cluster centres. The computational complexity of repeated density estimation is high for high-dimensional data with many objects. A careful seeding method, namely, k-means++ [4], has been used to avoid the randomized seeding technique of the k-means algorithm. However, random selection is still commonly used in practice.

The method in this paper can find the number of clusters from the data before clustering and select the initial cluster centres from the dense regions in the data. In previous methods, finding the number of clusters and selecting the initial cluster centres are usually conducted in separate steps [8,14,22,38]. In this new method, we perform them in the same process.

3. I-nice method and I-niceSO algorithm

In this section, we present a new clustering algorithm, namely, I-niceSO, which is a k-means-type algorithm that can automatically identify the number of clusters in input data and select the initial cluster centres from dense regions. The algorithm is based on an innovative approach, namely, I-nice, to identifying the number of clusters by allocating observation points in the data space and modelling the distance distributions between objects in the data and the observation points with Gamma mixture models (GMMs). The GMMs are automatically solved using the EM algorithm, and the best-fitted model is selected using the Akaike information criterion. From the final selected GMM, I-niceSO identifies the number of clusters and the initial cluster centres. Furthermore, the identified number of clusters and selected initial cluster centres are used as input parameters to the k-means algorithm to cluster data. Since the number of clusters and the initial cluster centres are automatically found from input data, I-niceSO is a parameter-free clustering algorithm and very easy to use in practice.

3.1. Distance distributions of observation points

Let R^d be a data domain of d dimensions and $Y \subset R^d$ a set of N objects $\{Y_1, Y_2, \ldots, Y_N\}$ in R^d. Let $p \in R^d$ be a randomly generated point with a uniform distribution. Define p as an observation point to Y. Given a distance function $d(.)$ on R^d, we compute all distances between observation point p and $Y_i \in Y$ to transform $Y = \{Y_1, Y_2, \ldots, Y_N\}$ into a set of distances $X_p = \{x_1, x_2, \ldots, x_N\}$. Elements of X_p and Y have one-to-one relations. In other words, every element in Y has a corresponding element of distance value in X_p and vice versa. Therefore, X_p is the distance distribution of Y with respect to observation point p. Given a different observation point, we can compute a different distance distribution from Y.

Fig. 2 shows two distance distributions from the data in Fig. 1(b) with respect to observation points O1 and O2. The distance distributions of observation points contain information on clusters in the original data. That is, the peaks of a distance distribution reveal the dense regions in the data, thereby reflecting the number of clusters. Therefore, by identifying the number of peaks in the distance distribution, one indirectly identifies the number of clusters in the data.

3.2. Model distance distributions with a GMM

We consider a distance distribution that contains multiple peaks, i.e., the data contain more than one cluster. The distribution can be modelled as a GMM, where each component models a peak. A GMM is used instead of a Gaussian mixture model because the distance values are non-negative. GMMs are widely used in other areas [43,44,46]. The method for solving for the parameters of a GMM is well-developed [5].

Let $X_p = (x_1, x_2, \ldots\ldots, x_N)$ be a set of normalized distance values calculated from Y with respect to observation point p. The GMM of X_p is defined as

$$p(x|\theta) = \sum_{j=1}^{M} \pi_j g(x|\theta_j), \quad x \geq 0 \tag{1}$$

where θ is the vector of parameters of the GMM, i.e., $(\pi_1, \ldots, \pi_M, \ \theta_1, \ldots, \theta_M)$, and M is the number of the Gamma components in the mixture model. π_j is the mixing proportion, and θ_j are the parameters, including shape parameter α_j and scale parameter β_j. The density function of each Gamma component is

$$g(x|\alpha_j, \beta_j) = \frac{x^{\alpha_j - 1} e^{-(x/\beta_j)}}{\beta_j^{\alpha_j} \Gamma(\alpha_j)}, \quad 1 \leq j \leq M \tag{2}$$

where Gamma function $\Gamma(x)$ is defined as $\Gamma(x) = \int_0^\infty t^{x-1}e^{-t}dt$, which is a definite integral for $\Re[x] > 0$ (integral form of Euler). This density function must satisfy the constraints $\alpha_j > 0$ and $\beta_j > 0$, and the mixing proportion parameters are subject to the following constraints:

$$\pi_j \geq 0, \quad \sum_{j=1}^{M} \pi_j = 1 \tag{3}$$

Let $X_p = (x_1, x_2, \ldots\ldots, x_N)$ be a materialization of N random samples. The joint distribution of the i.i.d. random samples is given by

$$p(X_p|\theta) = \prod_{i=1}^{N} p(x_i|\theta) \tag{4}$$

3.3. Solve the GMM with EM

Given $X_p = (x_1, x_2, \ldots\ldots, x_N)$, the parameters of the GMM are solved by maximizing the log-likelihood function of

$$\mathcal{L}(\theta|X_p) = \sum_{i=1}^{N} \log \left(\sum_{j=1}^{M} \pi_j g(x_i|\alpha_j, \beta_j) \right) \tag{5}$$

The EM algorithm is used to solve Eq. (5) [15]. The initial values for solving the GMM with the EM algorithm are assigned as follows: For M components in the GMM, we set the initial value of mixing proportion π_j to $1/M$, and the initial values of the shape α_j and scale β_j parameters are obtained from the closed-form estimates using the method of moments. The first and second moments are mean $\bar{x}$ and variance s^2, respectively. We assign $\frac{\bar{x}^2}{s^2}$ and $\frac{s^2}{\bar{x}}$ as the initial approximations of α_j and β_j, respectively, for the EM algorithm in solving the GMM.

Given X_p, latent discrete random variables $Z = \{z_i\}$ are introduced to identify the elements of X_p in the components of the GMM. $z_i = j$ indicates that element x_i in X_p is assigned to component j of the GMM. However, the values of $Z = \{z_i\}$ are unknown in advance [44].

The EM algorithm iterates in two steps, namely, expectation and maximization, until the stopping criterion is satisfied. In the expectation step, given the parameters θ^n, which were estimated in the nth previous iteration, the membership weight $p(Z_i = j|x_i, \theta^n)$ of each element x_i in X_p in each component j is computed as

$$\begin{aligned} p(Z_i = j|x_i, \theta^n) &= \frac{p(Z_i = j|\theta^n)g(x_i|\theta_j^n)}{\sum_{t=1}^{M} p(Z_i = t|\theta^n)g(x_i|\theta_t^n)} \\ &= \frac{\pi_j g(x_i|\theta_j^n)}{\sum_{t=1}^{M} \pi_t g(x_i|\theta_t^n)} \end{aligned} \tag{6}$$

where π_j is the mixing proportion weight of component j, which was estimated in the previous iteration. After the membership weights of all elements of X_p have been calculated, the expected values of log-likelihood function (5) with respect to the latent random variables $Z = \{z_i\}$ are computed as follows:

$$\begin{aligned} Q(\theta|\theta^n, X_p) &= E_{Z|\theta^n, X_p}\{\mathcal{L}(\theta|X_p, Z)\} \\ &= \sum_{i=1}^{N} E_{Z_i|\theta^n, x_i}\{\log g(x_i|\theta_{z_i}, Z_i = z_i) + \log p(Z_i = z_i|\theta)\} \\ &= \sum_{i=1}^{N}\sum_{j=1}^{M} p(Z_i = j|x_i, \theta^n)(\log g(x_i|\theta_j) + \log \pi_j) \end{aligned} \tag{7}$$

In the maximization step, the expectation of log-likelihood function $Q(\theta|\theta^n, X_p)$ is maximized to obtain the new estimates of parameters θ^{n+1} in the model:

$$\theta^{n+1} = \arg\max_{\theta} Q(\theta|\theta^n, X_p) \tag{8}$$

These two steps iterate until the stopping criterion $(\theta^{n+1} - \theta^n) < Threshold$ is satisfied, where *Threshold* is pre-defined. $Q(\theta|\theta^n, X_p)$ is calculated as follows: From Eq. (7), we derive

$$Q(\theta|\theta^n, X_p) = \sum_{i=1}^{N}\sum_{j=1}^{M} \{p(Z_i = j|x_i, \theta^n) \log \pi_j + p(Z_i = j|x_i, \theta^n) \log(g(x_i|\theta_j)\} \tag{9}$$

Since the two terms in (9) have the same sign, they can be maximized independently. The first term has only variables π_j. Therefore, it can be optimized using Lagrange multipliers. The constraints on variables π_j are given in (3). The unconstrained function with Lagrange multiplier λ is written as follows:

$$\Lambda(\pi, \lambda) = \sum_{i=1}^{N}\sum_{j=1}^{M} p(Z_i = j|x_i, \theta^n) \log \pi_j + \lambda\left(\sum_{j=1}^{M}\pi_j - 1\right) \tag{10}$$

Take the first derivatives with respect to each π_j and λ and set them equal to zero. After simplification of the derivative equations, we find the formula for calculating π_j:

$$\hat{\pi}_j = \frac{1}{N}\sum_{i=1}^{N} p(Z_i = j|x_i, \theta^n) \tag{11}$$

To calculate the parameter $\theta_j = (\alpha_j, \beta_j)$ by optimizing the second term of Eq. (9), we take the first derivatives of the second term of Eq. (9) with respect to α_j and set them equal to zero as follows:

$$\frac{\partial}{\partial \alpha_j}\left\{\sum_{i=1}^{N}\sum_{j=1}^{M} p(Z_i = j|x_i, \theta^n) \log g\left(x_j|\alpha_j, \frac{1}{\alpha_j}\frac{\sum_{i=1}^{N} x_i p(Z_i = j|x_i, \theta^n)}{\sum_{i=1}^{N} p(Z_i = j|x_i, \theta^n)}\right)\right\} = 0 \tag{12}$$

After performing various mathematical manipulations, we obtain

$$\log(\hat{\alpha}_j) - \psi(\hat{\alpha}_j) = \log\left(\frac{\sum_{i=1}^{N} x_j p(Z_i = j|x_i, \theta^n)}{\sum_{i=1}^{N} p(Z_i = j|x_i, \theta^n)}\right) - \left(\frac{\sum_{i=1}^{N} p(Z_i = j|x_i, \theta^n) \log x_i}{\sum_{i=1}^{N} p(Z_i = j|x_i, \theta^n)}\right) \tag{13}$$

where $\psi(\alpha) = \frac{\partial \log(\Gamma(\alpha))}{\partial \alpha} = \frac{\Gamma'(\alpha)}{\Gamma(\alpha)}$ is called the Digamma function. The Euler-Maclaurin formula applies to the Digamma function $\psi(\alpha)$, which can be approximated by

$$\psi(\alpha) = \log(\alpha) - \frac{1}{2\alpha} - \frac{1}{12\alpha^2} + \frac{1}{120\alpha^4} - \frac{1}{252\alpha^6} + \dots \tag{14}$$

Eq. (14)is the asymptotic expansion of $\psi(\alpha)$.

Since Eq. (13) is a typical non-linear equation, which does not have a closed-form solution, a simple gradient scheme is a naive method for obtaining a solution [46]. In this paper, we use the nlm (non-linear minimization) function in R language to solve Eq. (13). The nlm function uses a Newton-type algorithm [16] to estimate the parameters. It also uses the method of moments to assign the initial approximation for the Newton optimization process. The Newton-type algorithm estimates $\hat{\alpha}_j$ by maximizing the likelihood function. This algorithm is fast, as it causes the results to converge in few iterations.

Then, we take the first derivatives of the second term of Eq. (9) with respect to β_j and set them equal to zero as follows:

$$\frac{\partial}{\partial \beta_j}\left\{\sum_{i=1}^{N}\sum_{j=1}^{M} p(Z_i = j|x_i, \theta^n) \log g(x_i|\theta_j)\right\} = 0 \tag{15}$$

After various manipulations, we obtain

$$\hat{\beta}_j = \frac{1}{\alpha_j}\frac{\sum_{i=1}^{N} x_i p(Z_i = j|x_i, \theta^n)}{\sum_{i=1}^{N} p(Z_i = j|x_i, \theta^n)} \tag{16}$$

With Eq. (16) and the estimated value of $\hat{\alpha}_j$ from Eq. (13), we can estimate $\hat{\beta}_j$. The estimated parameters, namely, $\hat{\alpha}_j$ and $\hat{\beta}_j$,obtained from the method are locally optimal because Newton's method is only locally convergent.

For each observation point p, $(Mmax - 1)$ GMMs are built based on distance data and fitted with the EM algorithm, where $Mmax$ is the maximum number of GMMs.

3.4. Select the best-fitted GMM with AICc

A set of fitted GMMs is investigated with a model selection criterion to select the best-fitted model. The Akaike information criterion (AIC) was the first model selection criterion and was introduced by Hirotogu Akaike in his seminal paper 1973, which was titled "Information Theory and an Extension of the Maximum Likelihood Principle" [2]. The second-order Akaike information criterion (AICc) [24,39] was derived to improve the accuracy of model selection with AIC. The Bayesian

information criterion (BIC) [37] is another popular model selection criterion, which overcomes the overfitting problem by introducing a penalty term for the number of parameters in the model. The prediction from the AICc-selected model is less biased than the prediction from the BIC-selected model [10].

In this paper, we use the second-order Akaike information criterion (AICc) to select the best-fitted model because AICc has an additional bias correction term for fitting small datasets. For large datasets, it is equal to the first-order AIC [10].

The second-order Akaike information criterion is calculated as follows:

$$AICc = -2\log\left(\mathcal{L}(\theta^*)\right) + 2q\left(\frac{N}{N-q-1}\right) \tag{17}$$

where $\mathcal{L}(\theta^*)$ is the maximum log-likelihood value, N is the number of objects in the data and q is the number of parameters; $q = 3\,\mathrm{M}$, where M is the number of components in the GMM, which include the shape and scale $(\alpha_j,\ \beta_j)$ parameters of each Gamma component and the component proportion π_j.

Using (17), we compute AICc for each GMM from the same observation point and select the GMM with the smallest value of AICc as the best-fitted GMM by that observation point. The number of components of the best-fitted model is considered as the number of clusters observed from the observation point. The AIC model selection criterion provides a trade-off between model underfitting and overfitting and between the bias and variance of models [10,11]. When additional Gamma components are added to the GMM, the first term on the right-hand side of (17) decreases and the second term increases as more parameters are added to q. If N becomes much larger than q, then $(N - q - 1) \to N$ and AICc converges to AIC in model selection. Therefore, AICc satisfies a general model selection criterion regardless of sample size.

3.5. Determine the number of clusters in the data

From different observation points, different numbers of peaks of clusters are observed depending on their locations, similar to humans observing the peaks of mountains in a terrain. If we take the count of one person as the result, we will take the result of the person who observes the largest number of peaks. In the same way, we use the largest number of clusters observed from the observation point as the correct number of clusters in the data.

With AICc, we select the best-fitted GMM for each observation point. The number of components in the selected model is taken as the number of clusters observed from the observation point. From the set of best-fitted models selected from multiple observation points, the final model is the one with the largest number of clusters. If multiple best-fitted models have the same largest number of clusters, the model with the largest maximum likelihood value is selected as the final model.

3.6. Select the initial cluster centres

After the final GMM with the largest number of clusters has been selected, we can use the partition of distance values by the components to identify a set of objects as the initial cluster centres for k-means clustering of the original high-dimensional data. This process is conducted as follows:

1. From the cluster of distance values by each Gamma component, choose a set of values from the dense region of the component.
2. Obtain the set of object IDs that correspond to the set of values.
3. Take the data records of these objects from the original high-dimensional data, and compute the dissimilarity matrix of the objects.
4. Select the object that has the smallest row-wise sum from the matrix. This object likely corresponds to the nearest position in the dense region. The dense region is obtained by choosing the one-third nearest neighbours of the selected object with the KNN method. Then, the remaining objects are considered as outliers.
5. Remove the outliers, and select one object with the largest k-nearest neighbours as the initial cluster centre.
6. Repeat the above steps on all Gamma components.

The set of objects, one from each component, is used as the initial cluster centres to run the k-means algorithm. Since the objects are initially selected from the dense regions of components in the one-dimensional GMM, the objects selected from the dense regions are likely to be located in the dense regions in the original data space. The selected candidate objects are further evaluated by the k-nearest-neighbour method to identify the object located in the high-density region. Therefore, they are likely to be close to the inherent cluster centres in the data. The experimental results have verified this.

3.7. I-niceSO algorithm

The methods and techniques described above are integrated into I-niceSO, which is a parameter-free clustering algorithm. The details of I-niceSO are listed in Algorithm 1. The input to I-niceSO is a high-dimensional dataset. No other input parameter is required. The outputs include the number of clusters in the input data and the clustering result.

First, the algorithm generates a default number of objects from a uniform distribution as observation points. For each observation point, a set of distances between the observation point and other objects are computed to form the distance vector of this observation point. The object IDs are maintained in the distance vector.

Algorithm 1: I-niceSO: Identifying the number of clusters and the initial centres using a single observation point.

Input: A high-dimensional dataset Y
Output: The number of clusters K, clustering result
Initialization:
Generate P observation points using a uniform distribution;
Set $Mmax$ as the maximum number of GMMs to build for each observation point ;
Select the best-fitted models:
for $p := 1$ *to* P **do**
  Compute the distance vector X_p between objects of Y and observation point p using the Euclidean distance function;
  for $M := 2$ *to* $Mmax$ **do**
    Model X_p to GMM(p, M) ;
    Initialize parameters (α, β) using the method of moments, and set $\pi = 1/M$ for GMM(p, M) ;
    Call the EM algorithm to solve GMM(p, M) ;
    Calculate the AICc(M) of GMM(p, M) ;
  The best-fitted model, which is denoted as GMM(p), with k components is selected using the minimum AICc value ;
Keep all selected models GMMs(p) with different numbers of components for all p ;
Identify the number of clusters:
Keep the GMM(p) with the largest number of components K ;
Set the GMM(p) as the final model GMM_{final} and K as the number of clusters ;
Select the initial cluster centres:
Separate the object IDs for each component from the final model GMM_{final} ;
for $c := 1$ *to* K **do**
  Select a subset of the object IDs in the neighbourhood of the peak point from component c ;
  Choose the subset of objects from the input data Y ;
  Select the object with the largest k-nearest neighbours as one initial cluster centre ;
Clustering:
Assign K and the K initial cluster centres to k-means to cluster input data Y ;
Output K and the object cluster IDs of Y.

For each distance vector, a set of GMMs are set in a range of numbers of components. The EM algorithm is called to solve these models. The AICc criterion is used to select the best-fitted model for each distance vector and determine the number of clusters found from the distance vector.

The final model with the largest number of clusters is selected from the best-fitted models of all distance vectors, and the number of clusters given by this final model is the number of clusters in the input data.

A subset of the objects in the neighbourhood of the peak of each component in the final model is selected, and the k-nearest-neighbour method is used to select the object with the highest-density neighbourhood in the original dataset. The objects selected from all components of the final model are used as the initial cluster centres assigned to the k-means algorithm to cluster the input data.

4. I-niceMO: an I-nice algorithm that uses multiple observation results

I-niceSO determines the number of clusters from the best-fitted model of the distance distribution for one observation point. It works on data that have a few clusters. However, when a dataset contains many clusters, I-niceSO may miss clusters because the distance distribution for one observation point may not carry information on all clusters in the data. To overcome this problem, we can use the information on clusters carried by the distance distributions of all observation points. In this section, we present I-niceMO, which is an extension of I-niceSO, for using the results of the best-fitted GMMs from all observation points to identify the initial cluster centres and the number of clusters in the data.

4.1. Method for combining the results of multiple GMMs

Fig. 3 shows a dataset of 15 clusters in two-dimensional space. Three observation points are assigned to the dataset and shown as black dots 1, 2 and 3. Fig. 4(a), (b) and (c) are the distance distributions of the three observation points. According to visual inspection of the number of peaks in the distance distributions, none of them carry the full information of the 15 clusters in the data of Fig. 3. Therefore, I-niceSO does not perform well on this dataset.

Using I-niceSO, we find the best-fitted models of the three distance distributions. The distributions of all components in the best-fitted GMMs are shown in Fig. 4(d) , (e) and (f) . The first GMM from the distance distribution of observation point

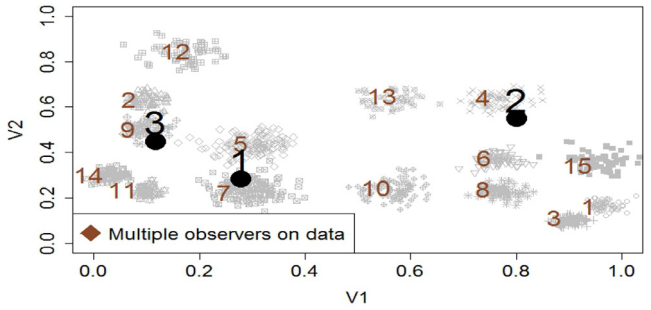


Fig. 3. Three observation points are located in the dataset with 15 clusters.

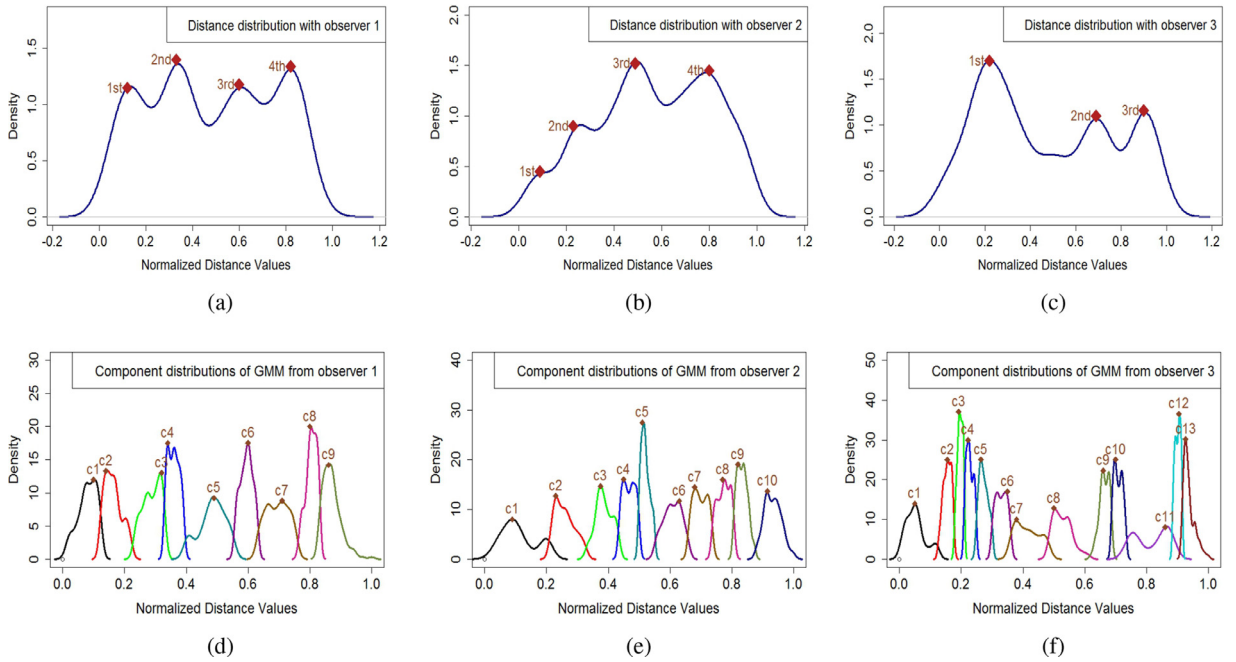


Fig. 4. Distance distributions for three observation points and distributions of individual components of their GMMs. (a) Distance distribution for observation point 1. (b) Distance distribution for observation point 2. (c) Distance distribution for observation point 3. (d) The component distributions of the selected GMM from the distance distribution in (a). (e) The component distributions of the selected GMM from the distance distribution in (b). (f) The component distributions of the selected GMM from the distance distribution in (c).

1 identifies 9 clusters. The other two GMMs identify 10 and 13 clusters. From none of the three observation points can all 15 clusters in the data be identified. Moreover, I-niceSO can identify more peaks than those that we can observe visually from the distance distributions of Fig. 4(d) , (e) and (f).

We can combine the information of three best-fitted GMMs to jointly determine the number of clusters in the data and the initial cluster centres. In doing so, we must investigate each component of the GMMs and its relation with the clusters in the data. We consider all objects in each component and use the *k*-nearest-neighbour method to find dense regions carried by these objects. We have 3 situations:

Case 1: One component is related to one cluster in the data.

Case 2: One component is related to more than one cluster in the data. This situation occurs when two or three clusters in the data have the same distances to the observation point.

Case 3: The same cluster in the data is observed by two or more components from different GMMs. In this case, the cluster can only be counted once.

Fig. 5 shows the distributions of objects in the nine components of observation point 1. Fig. 5(a), (b), (c),(e), (h) and (i) show that components 1, 2, 3, 5, 8 and 9 identify one cluster each. Fig. 5(d) shows that component 4 alone identifies 3 clusters. Fig. 5(f) and (g) show that components 6 and 7 identify 2 clusters each. In total, 13 clusters are identified from observation point 1, and 2 clusters are missed.

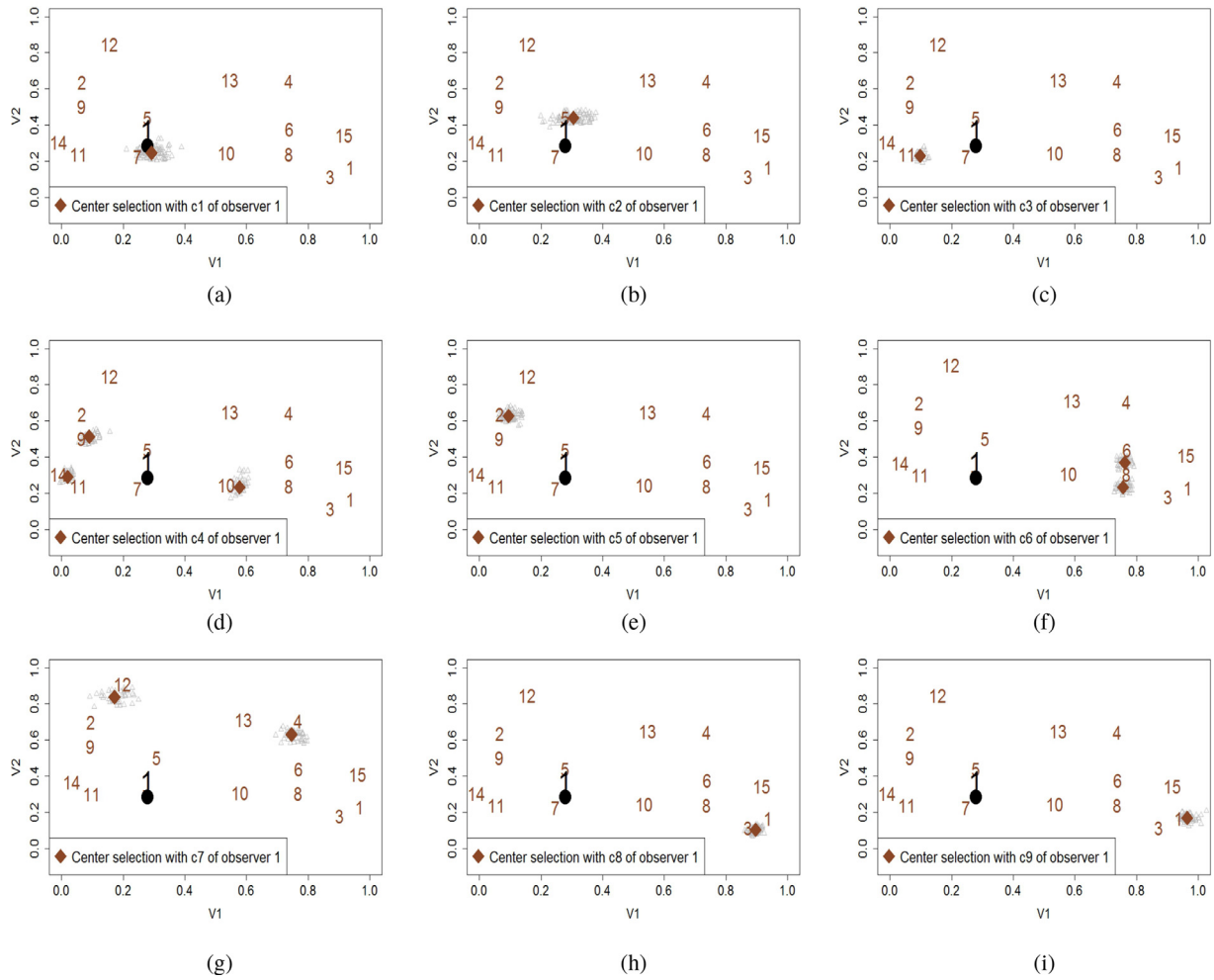


Fig. 5. Initial cluster centres identified by each component of the selected GMM from observation point 1. The numbers show the locations of the true cluster centres. (a) Component 1 selects one initial cluster centre from true cluster 7. (b) Component 2 selects one initial cluster centre from true cluster 5. (c) Component 3 selects one initial cluster centre from true cluster 11. (d) Component 4 selects three initial cluster centres from true clusters 9, 10 and 14. (e) Component 5 selects one initial cluster centre from true cluster 2. (f) Component 6 selects two initial cluster centres from true clusters 6 and 8. (g) Component 7 selects two initial cluster centres from true clusters 4 and 12. (h) Component 8 selects one initial cluster centre from true cluster 3. (i) Component 9 selects one initial cluster centre from true cluster 1.

Fig. 6(a) shows all initial cluster centres identified from observation point 1. Using the same *k*-nearest-neighbour method to analyse the best-fitted GMMs from observation points 2 and 3, we can obtain all initial cluster centres from all components of these two GMMs. Fig. 6(b) and (c) show the results. From observation point 2, 12 clusters are identified, and from observation point 3, 13 clusters are identified. From observation point 1, true clusters 13 and 15 are missed. From observation point 2, true clusters 2, 11 and 15 are missed. From observation point 3, true clusters 4, 11, 13 and 14 are missed. However, all 15 true clusters are identified by the components of the three GMMs.

Fig. 6(d) shows the result of combining the initial cluster centres from the three GMMs. Two initial cluster centres are merged into one if they are close to each other. For example, in Fig. 6(c), the two initial cluster centres in true clusters 5 and 7 are merged into one in the final result.

4.2. Select the initial cluster centres

Unlike I-niceSO, which determines the number of clusters based on the number of components in the selected GMM of one observation point, the number of clusters cannot be estimated as the number of components in the selected GMMs because the number of components is usually less than the number of clusters in the data. Therefore, in I-niceMO, we select the initial cluster centres identified by each component of the GMMs and use the initial cluster centres to determine the number of clusters in the data.

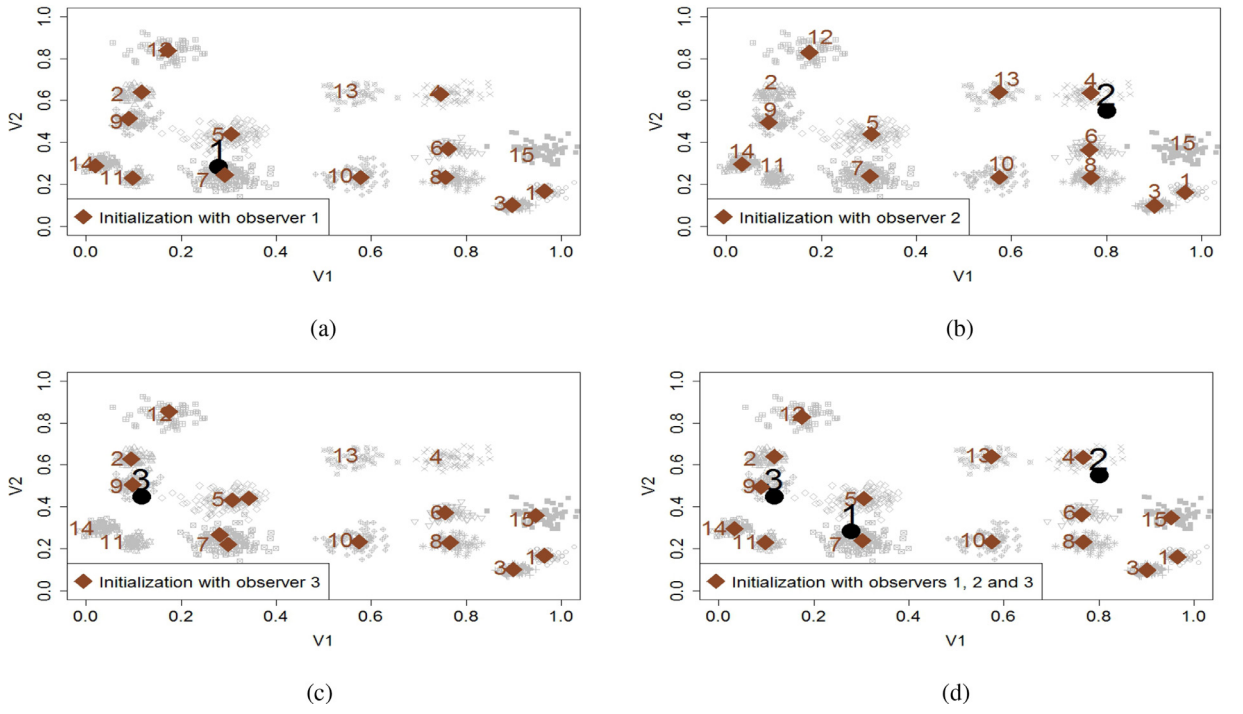


Fig. 6. (a) The initial cluster centres identified by the GMM from observation point 1. (b) The initial cluster centres identified by the GMM from observation point 2. (c) The initial cluster centres identified by the GMM from observation point 3. (d) The initial cluster centres obtained by combining the initial cluster centres identified by the three GMMs.

The initial cluster centres are identified by individual components as follows: For each component of a GMM, the set of object IDs is extracted and used to select the set of objects in the data. Then, the mutual distances between objects are computed, and the k-nearest-neighbour algorithm is applied to identify the dense region(s). For each dense region, one object with the largest density is selected as an initial cluster centre. A dense region is a small cluster of objects. One component can identify more than one dense region, which results in more than one initial cluster centre. For example, component 4 in Fig. 5(d) identifies 3 initial cluster centres. The initial cluster centres of all components from all selected GMMs constitute the candidate initial cluster centres in the data.

4.3. Determine the final set of initial cluster centres

Given the set of k candidate initial cluster centres, the dissimilarity matrix between the initial cluster centres is computed as follows:

$$d_{ij} = [(k_i - k_j)^T (k_i - k_j)]^{1/2}, \tag{18}$$

where k_i and k_j are two candidate initial cluster centres and d_{ij} is the Euclidean distance.

Five percent of the smallest distances are selected from the distance matrix and their mean is computed, as a threshold. The threshold is used to evaluate the k candidate initial cluster centres. We form a graph based on the k initial cluster centres and use the threshold to cut the graph. Any edge between two nodes in the graph that is greater than the threshold is removed from the graph. All isolated single nodes are taken as the initial cluster centres. For each group of connected nodes, one node is selected as an initial cluster centre, and the other nodes are discarded. All the selected nodes from the graph form the final set of initial cluster centres.

4.4. Determine the number of clusters

The number of initial cluster centres is the number of clusters in the data.

4.5. I-niceMO algorithm

The I-niceMO algorithm is shown in Algorithm 2. The steps before selecting initial cluster centres are the same as in I-niceSO. The algorithm also determines the number of clusters and the clustering result. It is a parameter-free clustering algorithm, which is easy to use.

Algorithm 2: I-niceMO: I-nice with multiple observation points.

Input: A high-dimensional dataset Y
Output: Initial K cluster centres, clustering result
Initialization:
Generate P observation points using the uniform distribution;
Set *Mmax* as the maximum number of GMMs to build for each observation point ;
Select the best-fitted models:
for *p := 1 to P* **do**
  Compute the distance vector X_p between Y and p using the Euclidean distance function ;
  for *M := 2 to Mmax* **do**
    Model X_p to GMM(p, M), and call the EM algorithm to solve GMM(p, M) ;
    Calculate AICc(M) of GMM(p, M) ;
  The best-fitted model GMM(p) with *c-max* components is selected using the minimum AICc ;
Keep all selected models GMMs(p) with different numbers of components for all p ;
Select the initial cluster centres:
for *p := 1 to P* **do**
  Keep selected model GMM(p) with estimated number of components *c-max* ;
  Separate the object IDs for each component of model GMM(p) ;
  for *c := 1 to c-max* **do**
    Select a subset of the object IDs in the neighbourhood of the peak point from component c ;
    Choose the subset of objects from the input data Y ;
    Apply the k-nearest-neighbour algorithm to identify dense region(s) ;
    Select the object with the largest density as an initial cluster centre from each dense region ;
  Keep the initial cluster centres for model GMM(p) ;
Integrate the initial cluster centres:
Keep the initial cluster centres k of all selected models GMMs(p) ;
Compute dissimilarity matrix $d(k)$ between centres k ;
Compute the mean from the 5% smallest values of $d(k)$ as the threshold;
for *i := 1 to k* **do**
  for *j := 1 to k* **do**
    if *((i!=j) && d(k_i, k_j) <threshold)* **then**
      g[i] = k[i]
    else
      s[i] = k[i]
Select as the initial cluster centre u a centre from each group of closest centres in g ;
Obtain K distinct initial cluster centres by adding s and u initial cluster centres ;
Identify the number of clusters:
The number of initial cluster centres is estimated as the number of clusters K ;
Clustering:
Assign K initial cluster centres to k-means to cluster input data Y ;
Output K and the object cluster IDs of Y.

5. Complexity analysis of the I-nice algorithms

Given a dataset Y with N objects, d dimensions and a default number P of observation points, the I-nice clustering algorithms cluster Y into K clusters in the following steps:

- Step 1: *Calculate P distance distributions.* Given an observation point, we must compute N distance values. The computational complexity is O(dN). For P observation points, the computational complexity in this step is O(dNP).
- Step 2: *Solve GMMs.* For each distance distribution, we must solve $(Mmax - 1)$ GMMs. Given a GMM with M components, the model is solved by the EM algorithm by the iteration of two steps: The expectation step calculates the membership weights and the expected value of the log-likelihood function with a computational complexity of O($2NM$). The computational complexity of the maximization step is $O((2+\rho)NM)$, where ρ is the average number of iterations in calculating $\hat{\alpha}_j$ in Eq. (13). Let μ be the average number of iterations in solving a GMM with EM. The computational complexity of solving $(Mmax - 1)$ GMMs is $O((Mmax-1)(4+\rho)\mu NM)$. The computational complexity of Step 2 is $O((Mmax-1)(4+\rho)\mu PNM)$.

Table 1
Configurations of the synthetic datasets.

Configuration No.	Instances (N)	Features (d)	Clusters (K)	Overlap (v)	No. of Datasets
1	100–1000	2	2–10	0.05–0.1	10
2	200–1000	2	2–10	0.1–0.8	10
3	110–500	2	2–7	0.005–0.01	10
4	180–500	3–4	3	0.007–0.01	10
5	120–500	3–4	4	0.003–0.01	10
6	120–560	3–4	5	0.005–0.01	10
7	120–550	3–4	6	0.009–0.01	10
8	150–1000	3–4	7	0.005–0.01	10
9	180–1000	3–4	8	0.009–0.01	10
10	180–1000	3–9	6–9	0.009–0.01	10
11	1200–2100	2	14–25	0.004–0.03	10
12	1000–15000	10–50	15–25	0.001–0.009	10

- Step 3: *Select the best-fitted models.* This step is straightforward. The AICc indices can be computed in Step 2. The number of clusters is determined in this step for I-niceSO.
- Step 4: *Select the initial cluster centres.* Let M be the average number of components in the P best-fitted GMMs. The average number of objects in each component is N/M. For each component, the computational complexity for finding the initial cluster centre(s) is $O(kN/M)$, where k is the number of objects in the k-nearest neighbourhood. The computational complexity of finding all initial cluster centre candidates is $O(kPN)$.
- Step 5: *Cluster data.* The complexity of this step is $O(KIdN)$, where K is the number of clusters and I is the number of iterations in the k-means clustering process.

The most time-consuming process is the GMM solving process with the EM algorithm. However, since the GMMs are built from one-dimensional distance data, solving GMMs is fast.

6. Experiments

In this section, we present experimental results of I-niceSO and I-niceMO on both synthetic and real-world data to demonstrate the performance of the proposed algorithms in identifying the correct number of clusters in the data in comparison with two popular methods: Elbow and Silhouette. We also show the clustering quality of I-nice-generated initial cluster centres in comparison with the initial cluster centres generated by the k-means++ algorithm and the modified mountain clustering algorithm, randomly selected initial cluster centres, and the true class centres in the data. First, the characteristics of the synthetic and real-world datasets are described. Then, the experimental settings and evaluation methods are presented. Finally, the experimental results are analysed and discussed.

6.1. Datasets

Both synthetic data and real-world data were used in the experiments. We designed 12 configurations for generating synthetic datasets. Table 1 shows the details of these configurations, which depend on the number of instances N, the number of features d, the number of clusters K and the overlapping conditions of clusters v in the data. For each configuration, 10 datasets were generated by randomly selecting the control parameters within the given ranges. In total, 120 synthetic datasets were generated.

We generated various configurations for the diversification of datasets. Our objective is to generate datasets that range from simple to complex by increasing the number of instances, the number of features, and the number of clusters and setting various overlapping thresholds.

For the first two configurations, two-dimensional Gaussian distributions were used to generate different numbers of clusters in each dataset. The clusters were located on a circle at equal distance with a radius equal to the square root of the specified number of clusters. Twenty synthetic datasets were generated in these two configurations (see Table 1).

For the next ten configurations, namely, configurations 3–12, each dataset was generated as follows: Given the number of instances N, the number of features d, the number of clusters K, and overlap v, K vectors of d dimensions were randomly created. The elements of each vector were randomly selected integers between 1 and K inclusive, and no vector was equal to any other vector. From each vector, a cluster centre was calculated as the values of elements minus 0.5. In this way, K cluster centres were obtained. After that, the distances between cluster centres were computed. Half of the shortest distance between cluster centres was taken as the radius of the clusters. The variances in the diagonal of the covariance matrix were computed by squaring the radius and multiplying it with overlap v. Then, the cluster centre vector and the variance in the diagonal of the covariance matrix were used as the parameters of the multivariate Gaussian distribution to generate a set of data points in the d-dimensional space that forms a cluster in the Gaussian distribution. K Gaussian clusters were generated independently and merged into a single dataset with N instances. In this way, we obtained a dataset with N instances, d features and K clusters. This data generation method partitions the data space into K^d units, for instance, $K \times K$ unit squares

Table 2
Characteristics of the synthetic shape datasets.

Number	Datasets	Instances	Features	Classes	Source
1	Aggregation	788	2	7	[20]
2	Compound	399	2	6	[50]
3	Path-based	300	2	3	[12]
4	Spiral	312	2	3	[12]
5	Shape	2000	2	4	[9]

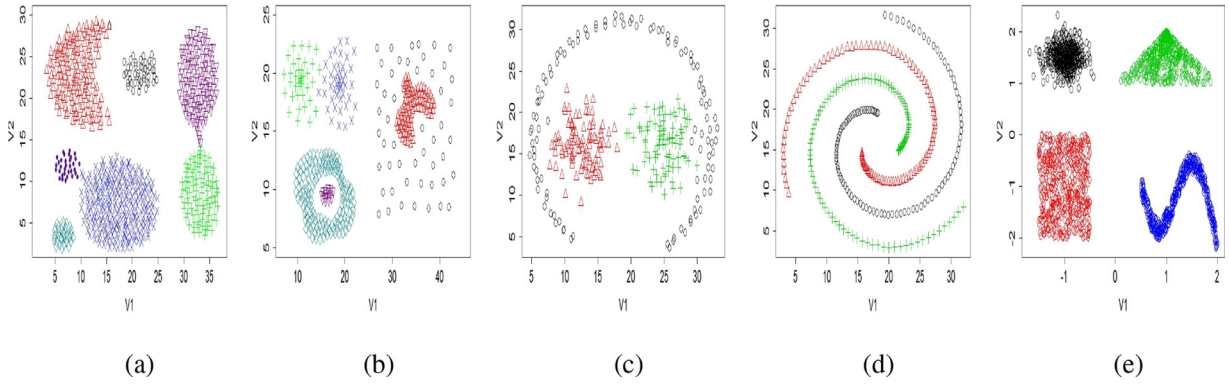


Fig. 7. Datasets of different shape structures. (a) Aggregation, (b) Compound, (c) Path-based, (d) Spiral, and (e) Shape.

Table 3
Characteristics of the twelve real-world datasets.

Number	Datasets	Instances	Features	Classes
1	Appendicitis	106	7	2
2	Banana	5300	2	2
3	Iris	150	4	3
4	Mammographic mass	961	6	2
5	Wine	178	13	3
6	Breast Tissue	106	10	6
7	Seed	210	7	3
8	Landsat satellite	6435	36	6
9	Glass	214	10	6
10	Ecoli	336	8	8
11	Texture	5500	40	11
12	Libras movement	360	91	15

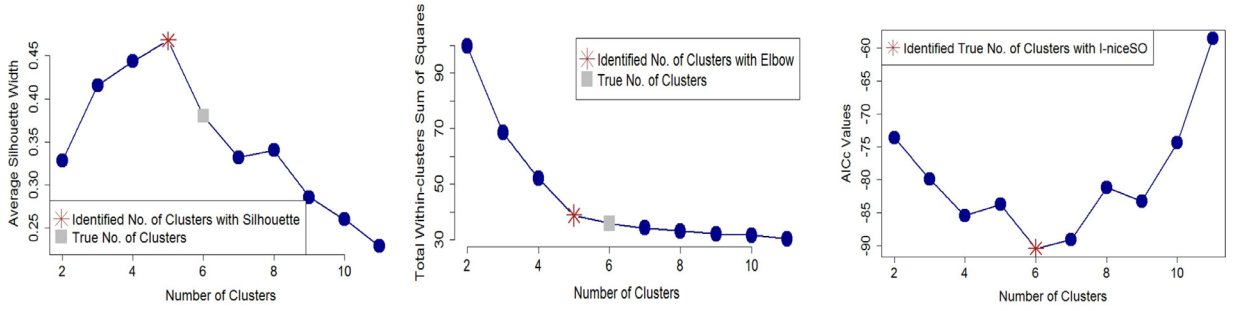
in two dimensions. Each cluster to be generated is allocated to a square with the square centre as the centre of the cluster. This method guarantees that clusters in the synthetic data are identifiable. However, the overlap factor makes the clusters not easy to recover from the synthetic data. The points in each cluster of a dataset were labelled so that the clustering results produced by the clustering algorithm can be compared with the true cluster labels to evaluate the clustering accuracy. One hundred synthetic datasets were generated in these ten configurations.

Moreover, we chose 5 shape datasets from different sources. The characteristics of these datasets are given in Table 2, and the structures of these datasets are illustrated in Fig. 7.

To evaluate the performances of I-niceSO and I-niceMO on real-world data, 12 real-world datasets were selected from the UCI machine learning repository [30] and the KEEL dataset repository [1]. These datasets are labelled with classes, which were taken as the true cluster labels, for comparison with the cluster labels generated by the proposed algorithms. The details of these real-world datasets are summarized in Table 3.

6.2. Experimental settings and evaluation methods

Two sets of experiments were conducted on both synthetic and real-world datasets. One was to evaluate the performances of I-niceSO and I-niceMO in finding the correct number of clusters in the data. Since the true numbers of clusters in both the synthetic and real-world datasets were known, we compared the numbers of clusters found by the proposed algorithms with the true number of clusters in each dataset. In addition, we compared the results of the proposed algorithms in terms of the number of clusters with the results of two well-known methods, namely, Elbow and Silhouette, and showed that the two proposed algorithms outperform the two popular methods.



(a) (b) (c)

Fig. 8. Curves generated by Silhouette, Elbow and I-niceSO on a synthetic dataset from configuration 10 to show the number of clusters. (a) The number of clusters indicated by the Silhouette curve is 5. (b) The number of clusters indicated by the Elbow position is 5. (c) The number of clusters indicated by the smallest AICc value is 6.

For example, Fig. 8 shows a comparison result of the three methods on a synthetic dataset with 6 clusters, which was generated from configuration 10 and is a complex synthetic dataset. Fig. 8(a) shows the curve of the average Silhouette width against the number of clusters. The highest average Silhouette width indicates the true number of clusters in the data. However, this Silhouette curve indicates that the number of clusters is 5, which is incorrect. Therefore, the Silhouette method failed on this dataset. Fig. 8(b) shows the curve of the sum of the within-cluster distances against the number of clusters. No clear Elbow position can be identified from this curve. The most likely position is 5, which is the same number as was found by the Silhouette method. Clearly, the Elbow position is not 6. Therefore, the Elbow method also failed. Fig. 8(c) shows the curve of the AICc values of I-niceSO against the number of clusters. The smallest AICc value indicates that the number of clusters is 6, which equals the true number of clusters in the data. Therefore, I-niceSO was successful in this case and able to find the correct number of clusters in this dataset, whereas the other two popular methods failed.

We conducted the same analysis on all synthetic and real-world datasets and compared the results of the three methods.

In these experiments, P is the number of observation points in the data space, M is the number of components in each GMM, and $Mmax$ is the maximum number of GMMs in each dataset. The value of P is set as 6. For each dataset, the value of $Mmax$ is set as the true number of clusters plus 5, and the value of M is selected from 2 to $Mmax$.

To produce the clustering results, the number of clusters and initial cluster centres identified by the proposed algorithms were used as input parameters of the k-means algorithm to cluster the dataset. In the second set of experiments, we evaluated the clustering results of I-niceSO and I-niceMO in comparison with other methods on synthetic and real-world datasets.

All the baseline methods for producing clustering results use the number of clusters as an input parameter and estimate the initial cluster centres for the clustering algorithm. The baseline methods are described as follows:

True centres: In this case, the number of clusters and label information were used to calculate the true centres. We computed the means of individual class-labelled data as the true class centres of the dataset. These true class centres were used as input parameters of the k-means algorithm to cluster the dataset.

***k*-means++:** The k-means++ seeding [4], which is the state-of-the-art algorithm, was used as the baseline method.

Random initialization: We conducted the experiment using randomly selected initial cluster centres in the k-means algorithm. In this case, we ran the k-means algorithm 10 times on each dataset with different sets of random initial cluster centres and computed the average clustering results and the number of iterations.

Modified Mountain Clustering Algorithm (MMCA): The modified mountain clustering algorithm was proposed in [49] for the estimation of the initial cluster centres and the number of clusters. In this experiment, the number of clusters was given to the algorithm, and the algorithm was used to estimate the initial cluster centres for the k-means clustering process.

To measure the clustering results, two evaluation criteria were used in the experiments. They are defined as follows:

1. Purity: Purity is the percentage of the objects classified correctly [31]. Purity is computed as follows:

$$Purity = \frac{1}{N}\sum_{i=1}^{K} \max_j \left|C_i \cap Y_j\right| \tag{19}$$

where N is the number of objects in the dataset, K is the number of clusters, C_i is the set of objects in cluster i and Y_j is the set of objects in class j that has the maximum intersection with cluster i among all sets of classes. The range of purity is between 0 and 1.

2. Adjusted Rand Index (ARI): ARI evaluates the consistency between two partitions [23]. Let $C = C_1, C_2, \ldots C_{K^C}$ be a partition of N objects into K^C clusters and $Y = Y_1, Y_2, \ldots Y_{K^Y}$ be a partition of N objects into K^Y classes. Let N_{ij} be the number of objects in both cluster C_i and class Y_j, N_i be the number of objects in cluster C_i, and N_j be the number of objects in class Y_i.

Table 4
AIC and BIC values computed with the Gamma and log-normal models for better fitting of the distance distribution.

Datasets	Distribution	AIC	BIC
Appendicitis	Gamma	**−9.45**	**−4.23**
	Log-normal	16.83	22.14
Banana	Gamma	**−2793.03**	**−2779.88**
	Log-normal	−2080.35	−2067.20
Iris	Gamma	**−14.58**	**−8.57**
	Log-normal	14.07	20.08
Mammographic Mass	Gamma	**−440.95**	**−431.51**
	Log-normal	−387.96	−378.52
Wine	Gamma	**13.11**	**19.47**
	Log-normal	49.75	56.11
Breast Tissue	Gamma	**−281.54**	**−276.23**
	Log-normal	−276.15	−270.84
Seed	Gamma	**97.97**	**104.66**
	Log-normal	162.86	169.55
Landsat Satellite	Gamma	**−2442.44**	**−2428.90**
	Log-normal	−1698.23	−1684.69
Glass	Gamma	**−305.31**	**−298.58**
	Log-normal	−295.72	−289.00
Ecoli	Gamma	**−15.46**	**−7.83**
	Log-normal	31.07	38.70
Texture	Gamma	**−5673.32**	**−5660.10**
	Log-normal	−4980.52	−4967.29
Libras Movement	Gamma	**−296.59**	**−288.82**
	Log-normal	−211.30	−203.53

ARI is calculated as follows:

$$ARI = \frac{\sum_{ij}\binom{N_{ij}}{2} - \left[\sum_i \binom{N_i}{2}\sum_j \binom{N_j}{2}\right]/\binom{N}{2}}{\frac{1}{2}\left[\sum_i \binom{N_i}{2}\sum_j \binom{N_j}{2}\right] - \left[\sum_i \binom{N_i}{2}\sum_j \binom{N_j}{2}\right]/\binom{N}{2}} \tag{20}$$

The experiments were conducted on a computer with a 3.60 GHz Intel(R)Core(TM)i7-4790 CPU with 8 GB of memory running Windows 7 Professional and 128 GB of memory running Windows Server 2012 R2 Standard. The program was written in the R (3.2.2) language.

6.3. Experimental results and analysis

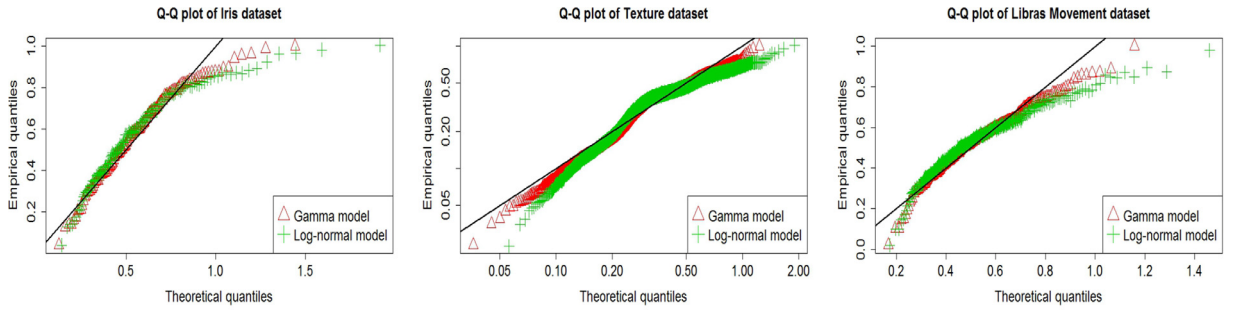
First, we present the fitting performances of the two distributions that describe the non-negative values. Then, we present the experimental results on identifying the correct number of clusters and the clustering accuracy on synthetic and real-world datasets. The proposed I-nice method transforms the high-dimensional data into one-dimensional non-negative distance data. Both Gamma and log-normal distributions use the non-negative data. We compared the fitting performances of the Gamma and log-normal models on the non-negative distance data.

We used maximum likelihood estimation to estimate the parameters of the Gamma and log-normal models and computed the AIC and BIC values from the fitted models. The Gamma model better fits the distance distributions than the log-normal model because the AIC and BIC values estimated with the Gamma model are smaller than those estimated with the log-normal model, as shown in Table 4.

In addition, we performed the quantile-quantile (Q-Q) test to visualize the goodness of fit with the Gamma and log-normal models on distance values from real-world datasets. When the quantiles from the data are plotted against the corresponding quantiles from a theoretical distribution, the Q-Q plot shows how closely the dataset fits the chosen theoretical distribution. The main advantage of the Gamma model is also demonstrated by a test in which the Q-Q plotted points from the Gamma model fall closely onto the identical line in Fig. 9.

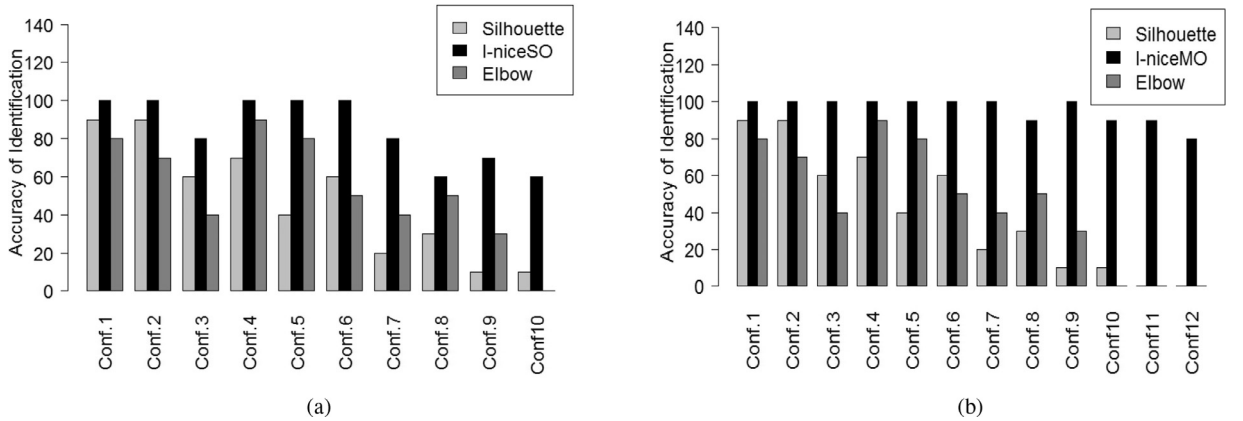
6.3.1. Performance in terms of finding the correct number of clusters

As in the example shown in Fig. 8, we tested I-niceSO, I-niceMO and two other methods on all synthetic datasets from configurations 1–10. For the datasets in each configuration, we counted the number of datasets for which the number of clusters was correctly determined by each method. The performance of each method was measured by the percentage of datasets for which the number of clusters was correctly found. Fig. 10(a) shows the performances of three methods on the datasets of configurations 1–10, which are given in Table 1. I-niceSO identified the correct numbers of clusters in the datasets



(a) (b) (c)

Fig. 9. Q-Q plot generated with the Gamma and log-normal models for better visualization of the distance distribution. (a) Q-Q plot generated with the Gamma and log-normal models on distance values from the Iris dataset. (b) Q-Q plot generated with the Gamma and log-normal models on distance values from the Texture dataset. (c) Q-Q plot generated with the Gamma and log-normal models on distance values from the Libras Movement dataset.



(a) (b)

Fig. 10. Performance comparisons of three methods in terms of accuracy in finding the correct numbers of clusters. (a) Results of the Silhouette, I-niceSO and Elbow methods on the datasets of configurations 1–10. (b) Results of the Silhouette, I-niceMO and Elbow methods on the datasets of configurations 1–12.

Table 5
Performances of I-niceSO, Elbow, Silhouette and I-niceMO in terms of finding the numbers of clusters from shape synthetic datasets.

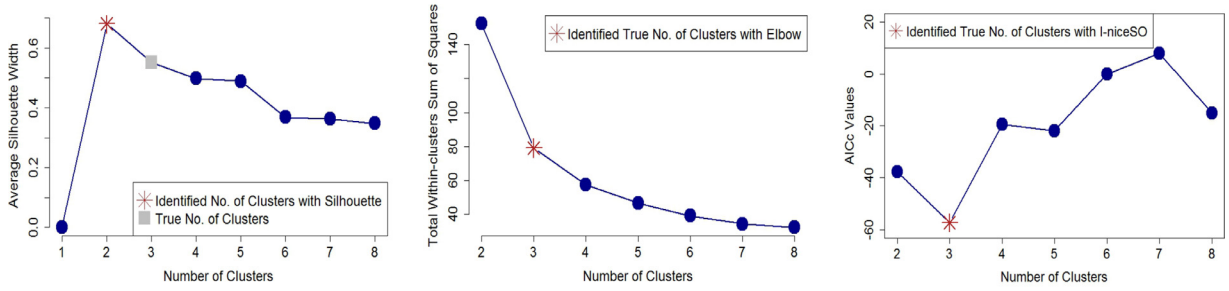
Datasets	Classes	I-niceSO	Elbow	Silhouette	I-niceMO
Aggregation	7	4	Not	4	7
Compound	6	3	2	2	6
Path-based	3	3	Not	3	3
Spiral	3	3	Not	8	3
Shape	4	4	4	4	4

in five configurations: 1, 2, 4, 5 and 6. For the datasets of the other five configurations, I-niceSO substantially outperformed Elbow and Silhouette.

Fig. 10(b) shows the performances of I-niceMO and two other methods on the datasets of all 12 configurations. I-niceMO significantly outperformed Elbow and Silhouette. I-niceMO also outperformed I-niceSO on the datasets generated with configurations 3, 7, 8, 9, and 10. For datasets from more complex configurations 11 and 12, I-niceMO achieved an accuracy of almost 90%. However, the other methods could not produce correct results.

Table 5 lists the 5 shape datasets shown in Fig. 7 and the numbers of clusters identified by the four methods. The Elbow method only identified the correct number of clusters in one dataset. Silhouette found the correct numbers of clusters in two datasets. I-niceSO identified the correct numbers of clusters in three datasets. I-niceMO found the correct numbers of clusters in all 5 datasets.

In addition, we compared the performances of four methods on real-world datasets. Fig. 11 shows the results in terms of finding the correct number of clusters from dataset Iris by I-niceSO and two other methods. Both Elbow and I-niceSO correctly found that the number of clusters is 3, which equals the number of classes in dataset Iris. The Silhouette method



(a) (b) (c)

Fig. 11. Curves generated by Silhouette, Elbow and I-niceSO on the Iris dataset to show the number of clusters. (a) The number of clusters indicated by the Silhouette curve is 2. (b) The number of clusters indicated by the Elbow position is 3. (c) The number of clusters indicated by the smallest AICc value is 3.

Table 6
Performances of I-niceSO, Elbow, and Silhouette and I-niceMO in terms of finding the numbers of clusters for 12 real-world datasets.

Datasets	Classes	I-niceSO	Elbow	Silhouette	I-niceMO
Appendicitis	2	2	Not	2	2
Banana	2	2	Not	8	2
Iris	3	3	3	2	3
Mammographic Mass	2	2	Not	2	2
Wine	3	3	2	2	3
Breast Tissue	6	6	3	2	6
Seed	3	3	3	2	3
Landsat Satellite	6	2	3	3	6
Glass	6	6	Not	4	6
Ecoli	8	2	3	3	8
Texture	11	14	Not	2	11
Libras Movement	15	15	Not	11	15

found that the number of clusters was 2, which is smaller than the number of classes in the dataset. Therefore, the Silhouette method failed on this dataset.

Table 6 shows the performances of four methods on all real-world datasets listed in Table 3. Column Classes lists the number of classes in each dataset, which was considered as the true number of clusters. I-niceSO found the correct number of clusters for 9 of 12 real-world datasets. Elbow did not perform well on these real-world datasets. It only correctly found the numbers of clusters for two datasets. It found the incorrect numbers of clusters for 4 datasets. More seriously, it could not identify the Elbow positions on six datasets, which are marked as “Not”. The Silhouette method found the numbers of clusters for all real-world datasets. However, it only succeeded on two datasets. The numbers of clusters for the other ten datasets were incorrect. I-niceMO identified the correct numbers of clusters for all real-world datasets. From this table, both the Elbow and Silhouette methods performed poorly on these twelve real-world datasets, while I-niceSO found the correct numbers of clusters for 75% of these real-world datasets and I-niceMO for 100%.

We found that the MMCA method cannot determine the correct number of clusters for most of the datasets. Therefore, it is not comparable to the proposed method in identifying the correct number of clusters.

6.3.2. Performance in terms of improvement of clustering with I-nice-identified initial cluster centres

We used both synthetic and real-world data to investigate the improvement of clustering performance by the I-niceSO and I-niceMO algorithms by using I-nice-identified initial cluster centres. The results show that the use of the automatically selected initial cluster centres can increase the clustering accuracy and efficiency of the *k*-means clustering process.

We generated 11 synthetic datasets from configurations 1–3 with no more than 10 clusters for evaluating the clustering performance of I-niceSO and 12 synthetic datasets from configurations 11 and 12 with the number of clusters between 15 and 25 for evaluating the clustering performance of I-niceMO. The datasets are recorded in Tables 7 and8, respectively.

Figs. 12 and 13 show the comparisons of the *k*-means clustering results on cluster datasets Data11 and DS1 using randomly selected initial cluster centres and I-niceSO and I-niceMO for clustering the same datasets using I-nice-selected initial cluster centres. Data11 has 4 clusters, and DS1 has 15 clusters. The true number of clusters in the data was input to the *k*-means algorithm.

Figs. 12(a) and 13(a) show the initial cluster centres randomly selected from the datasets. These initial cluster centres do not cover all true clusters in the datasets. These are not advantageous locations at which to start the *k*-means clustering process. The clustering result of *k*-means is dependent on the initial cluster centres. Figs. 12(b) and 13(b) show the clustering

Table 7
Comparison of clustering results of I-niceSO and other methods on 11 synthetic datasets generated with configurations 1–3.

Datasets	K	Purity					ARI				
		I-niceSO	True Centres	Kmeans++	Random	MMCA	I-niceSO	True Centres	Kmeans++	Random	MMCA
Data1	2	0.980	0.980	0.980	0.970	0.980	0.921	0.921	0.921	0.883	0.921
Data2	5	1.000	1.000	1.00	0.814	1.00	1.000	1.000	1.00	0.738	1.00
Data3	3	0.932	0.932	0.932	0.932	0.932	0.808	0.808	0.808	0.808	0.808
Data4	3	0.963	0.963	0.963	0.963	0.963	0.893	0.893	0.893	0.893	0.893
Data5	5	0.993	0.993	0.990	0.933	0.990	0.987	0.987	0.976	0.887	0.976
Data6	6	1.000	1.000	1.000	0.847	1.000	1.000	1.000	1.000	0.779	1.000
Data7	10	1.000	1.000	0.902	0.802	1.000	1.000	1.000	0.865	0.750	1.000
Data8	8	1.000	1.000	0.886	0.876	1.000	1.000	1.000	0.840	0.826	1.000
Data9	7	0.995	0.995	0.995	0.852	0.995	0.990	0.990	0.990	0.802	0.990
Data10	9	1.000	1.000	0.771	0.887	1.000	1.000	1.000	0.728	0.847	1.000
Data11	4	0.988	0.988	0.988	0.972	0.988	0.971	0.971	0.971	0.940	0.971

Table 8
Comparison of clustering results of I-niceMO and other methods on 12 synthetic datasets with configurations 11–12.

Datasets	K	Purity					ARI				
		I-niceMO	True Centres	Kmeans++	Random	MMCA	I-niceMO	True Centres	Kmeans++	Random	MMCA
DS1	15	0.995	0.995	0.922	0.845	0.938	0.988	0.988	0.863	0.730	0.924
DS2	17	0.955	0.998	0.847	0.844	0.911	0.913	0.997	0.748	0.757	0.849
DS3	25	0.930	0.969	0.904	0.880	0.855	0.907	0.947	0.819	0.792	0.761
DS4	15	0.976	0.998	0.865	0.888	0.940	0.927	0.995	0.765	0.813	0.871
DS5	16	0.976	0.994	0.822	0.861	0.806	0.943	0.988	0.731	0.828	0.731
DS6	17	0.965	0.988	0.953	0.949	0.923	0.938	0.979	0.920	0.891	0.851
DS7	18	0.984	0.998	0.827	0.912	–	0.970	0.997	0.752	0.871	–
DS8	19	0.970	0.996	0.960	0.926	0.825	0.948	0.989	0.899	0.877	0.737
DS9	20	0.972	0.996	0.943	0.934	–	0.946	0.993	0.908	0.858	–
DS10	22	0.948	0.998	0.922	0.887	–	0.928	0.996	0.890	0.835	–
DS11	24	0.989	0.998	0.964	0.951	–	0.972	0.998	0.918	0.881	–
DS12	25	1.000	1.000	0.910	0.802	–	1.000	1.000	0.857	0.706	–

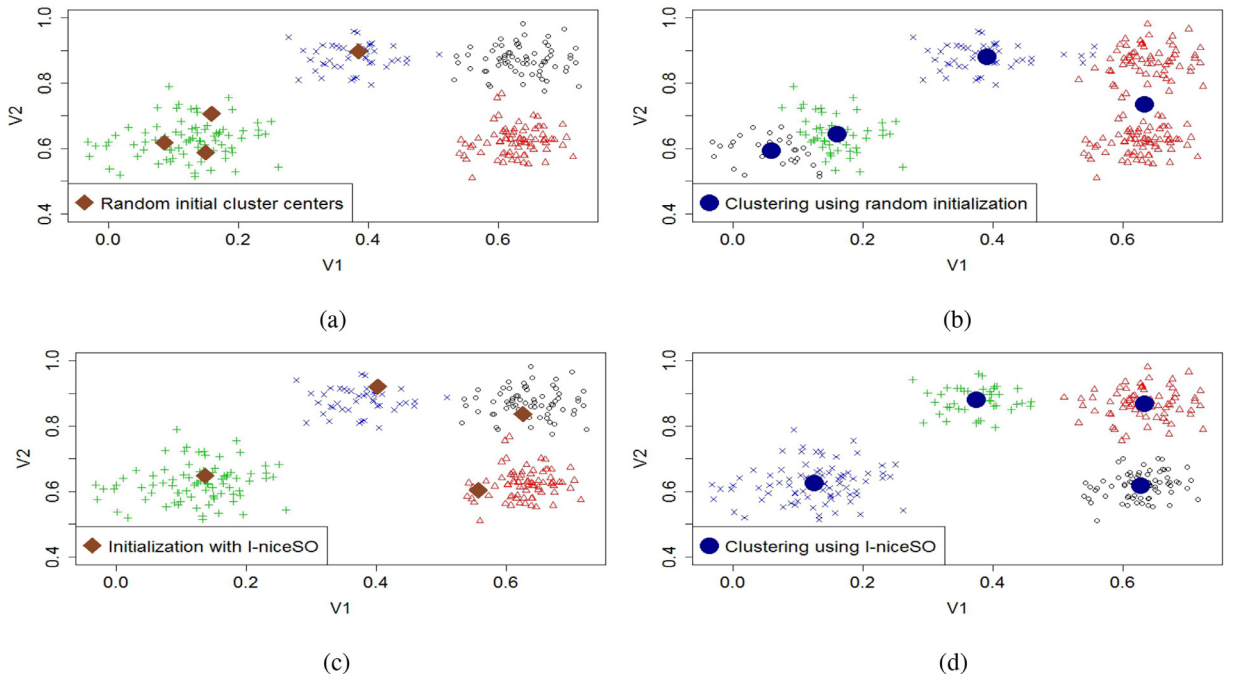


Fig. 12. Comparison of clustering results with randomly selected initial cluster centres and I-niceSO-identified initial cluster centres on dataset Data11 from configuration 3. (a) Randomly selected initial cluster centres. (b) Clustering results using random initial centres. (c) The initial cluster centres selected by I-niceSO. (d) Clustering results using I-niceSO-generated initial centres.

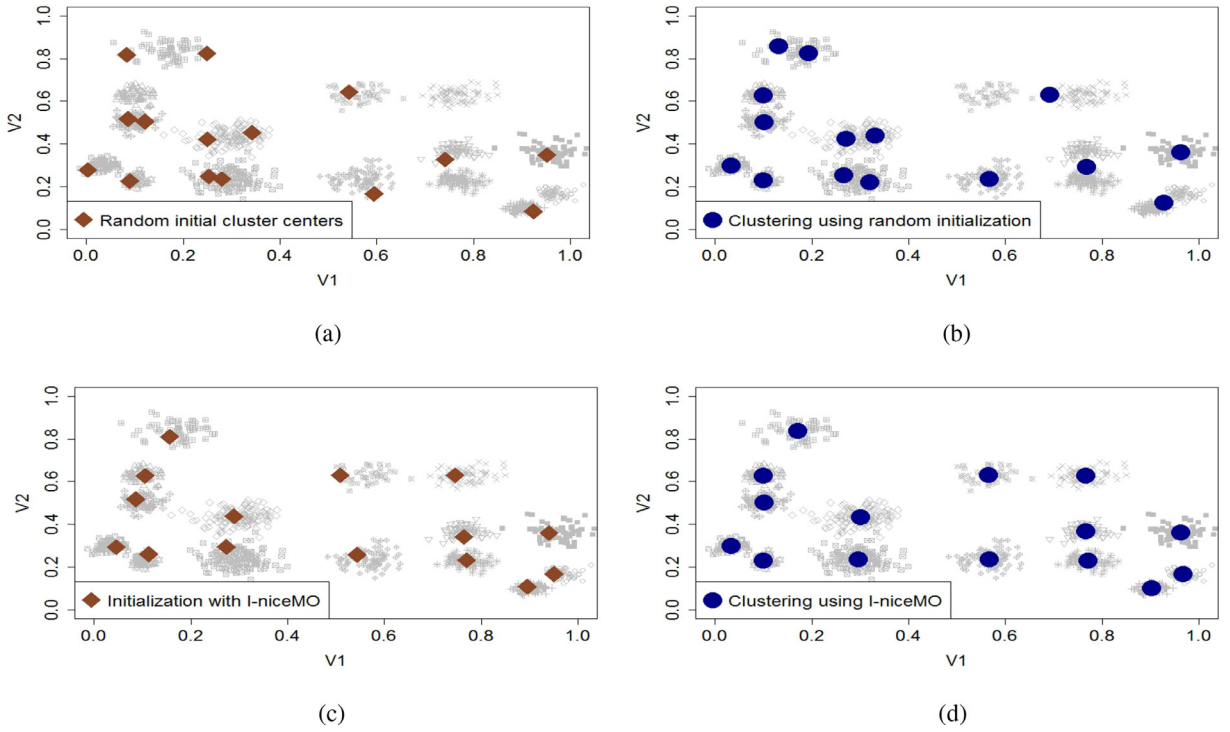


Fig. 13. Comparison of clustering results with randomly selected initial cluster centres and I-niceMO-identified initial cluster centres on dataset DS1 from configuration 11. (a) Randomly selected initial cluster centres. (b) Clustering results using random initial centres. (c) The initial cluster centres identified by I-niceMO. (d) Clustering results using I-niceMO-generated initial centres.

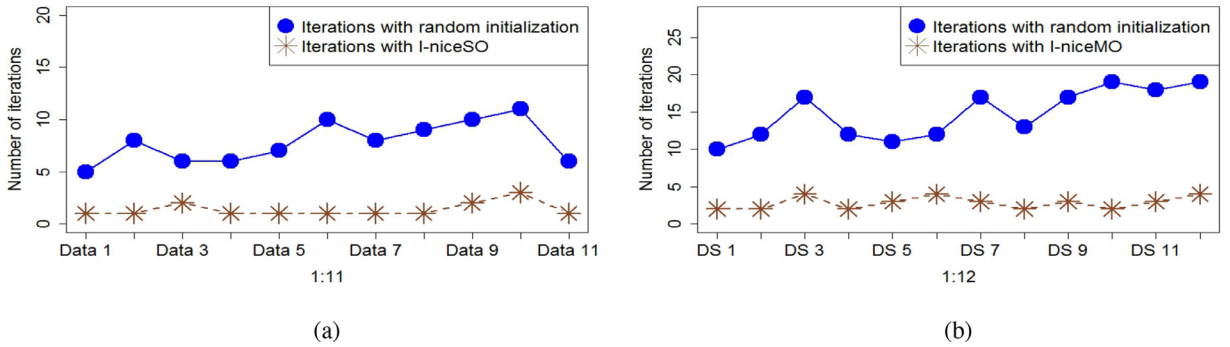


Fig. 14. Comparison of the numbers of iterations with randomly selected initial cluster centres and I-nice-selected initial cluster centres in the *k*-means clustering process. (a) The number of iterations with random initial cluster centres and the I-niceSO algorithm on the 11 datasets from configurations 1–3. (b) The number of iterations with random initial cluster centres and the I-niceMO algorithm on the 12 datasets from configurations 11–12.

results on the two datasets by *k*-means. Many true clusters were incorrectly clustered. In Fig. 12(b), two true clusters on the right were clustered as one cluster because no initial cluster centre was selected for these two true clusters. One true cluster on the left was clustered as two clusters because three initial cluster centres were selected from this true cluster. Only one true cluster was correctly clustered by *k*-means with randomly selected initial cluster centres. In Fig. 13(b), several true clusters were incorrectly clustered because of the random selection of the initial cluster centres. Since this dataset contains 15 true clusters, it is very difficult to correctly cluster it without starting from good initial cluster centres.

Figs. 12(c) and 13(c) show the initial cluster centres automatically identified by I-niceSO and I-niceMO, respectively, from the two datasets. Figs. 12(d) and 13(d) show the clustering results of I-niceSO and I-niceMO, which also use the *k*-means clustering process to cluster these two datasets. According to Figs. 12(c) and 13(c), for each true cluster in the data, one initial cluster centre was identified. All true clusters in these two datasets were correctly clustered by I-niceSO and I-niceMO, as shown in Figs. 12(c) and 13(c). Therefore, the automatically identified initial cluster centres significantly improved the clustering accuracy of the *k*-means clustering process.

Table 9
Comparison of clustering results of I-niceSO, I-niceMO and other methods on real-world datasets.

Datasets	Purity						ARI					
	I-niceSO	True Centre	I-niceMO	Kmeans++	Rand.	MMCA	I-niceSO	True Centre	I-niceMO	Kmeans++	Rand.	MMCA
Appendicitis	0.830	0.830	0.830	0.801	0.801	0.830	0.378	0.378	0.378	0.314	0.314	0.378
Banana	0.566	0.566	0.566	0.566	0.566	–	0.017	0.017	0.017	0.017	0.017	–
Iris	0.893	0.893	0.893	0.893	0.825	0.893	0.730	0.730	0.730	0.730	0.638	0.730
Mammo.	0.685	0.685	0.685	0.685	0.685	0.686	0.136	0.136	0.136	0.136	0.136	0.136
Wine	0.702	0.702	0.702	0.702	0.691	0.702	0.371	0.371	0.371	0.371	0.356	0.371
BTissue	0.433	0.433	**0.452**	0.433	0.433	0.434	0.186	0.186	0.175	0.179	0.186	0.179
Seed	0.895	0.895	0.895	0.895	0.891	0.895	0.716	0.716	0.716	0.716	0.711	0.716
Landsat	–	0.737	0.737	0.594	0.614	–	–	0.529	0.529	0.329	0.462	–
Glass	0.570	0.588	0.588	0.565	0.559	0.537	0.261	0.270	0.255	0.242	0.246	0.263
Ecoli	–	0.818	0.830	0.824	0.812	0.827	–	0.690	0.426	0.356	0.405	0.494
Texture	–	0.660	0.642	0.645	0.606	–	–	0.544	0.508	0.506	0.458	–
Libras	0.438	0.472	**0.488**	0.430	0.457	0.478	0.265	0.320	0.317	0.259	0.283	0.313

Fig. 14 shows the numbers of iterations of the *k*-means clustering process in producing the clustering results for the synthetic datasets with randomly selected initial cluster centres and automatically identified initial cluster centres. Fig. 14(a) shows a comparison of the performance of I-niceSO and the average performance of the *k*-means algorithm with random initial cluster centres. Fig. 14(b) shows a comparison of the performance of I-niceMO and the average performance of the *k*-means algorithm with random initial cluster centres. The *k*-means clustering process required few iterations to converge when the automatically identified initial clusters were specified. Given the randomly selected initial cluster centres, the *k*-means clustering process required many more iterations to converge, and the clustering process fluctuated depending on the random selection of the initial cluster centres. The clustering results of the I-nice algorithms are comparatively stable and consistent.

Table 7 shows the clustering results of the 11 synthetic datasets generated with configurations 1–3. These two-dimensional datasets are comparatively easy to cluster because they only contain a few clusters. The clustering results were measured in terms of purity and ARI. The I-niceSO column indicates that the clustering results were obtained by using initial cluster centres automatically selected by I-nice. The True Centres column indicates that the clustering results were obtained by specifying the true cluster centres as initial cluster centres in the *k*-means clustering process. The two clustering results are the same, which indicates that I-niceSO automatically identified initial cluster centres that are very close to the true cluster centres. Column Random shows the average clustering results of the *k*-means algorithm with random initial cluster centres, whose performance was much lower. The proposed I-niceSO algorithm significantly outperformed the randomly selected initial cluster centres in terms of both purity and ARI. The Kmeans++ and MMCA columns indicate that the clustering results were obtained by using the *k*-means++- and MMCA-selected initial cluster centres, respectively. The MMCA algorithm performs well in selecting the initial cluster centres on datasets with a few hundred objects. The I-niceSO method also outperformed *k*-means++ on four datasets and the MMCA methods on one dataset in terms of both purity and ARI.

Table 8 shows the clustering results of 12 synthetic datasets generated with configurations 11 and 12. These datasets are difficult to cluster because they contain many objects with many clusters. The clustering results were measured in terms of purity and ARI. The I-niceMO column indicates that the clustering results were obtained by using initial cluster centres automatically selected by I-niceMO. The True Centres column indicates that the clustering results were obtained by specifying the true cluster centres as initial cluster centres in the *k*-means clustering process. The two clustering results are close, which indicates that I-nice automatically identified initial cluster centres close to the true cluster centres. Column Random shows the average clustering results of the *k*-means algorithm with the random initial cluster centres. The notation " - " in the MMCA column indicates that the MMCA algorithm did not select the initial cluster centres on the datasets with more than five thousand objects within several hours. The I-niceMO algorithm achieved markedly better purity and ARI values than the *k*-means++ method, the MMCA algorithm and randomly selected initial cluster centres. We also observe that the *k*-means with the random initial cluster centres produced poor clustering results on data with many clusters.

We also used the I-niceSO and I-niceMO algorithms to cluster the 12 real-world datasets and compared the clustering accuracies with the initial cluster centres computed with the same class labels in the data and the initial cluster centres automatically identified by I-nice-SO and I-niceMO. Table 9 shows the comparison results, which are measured in terms of purity and ARI, which measures the correspondence between the true class labels and the cluster IDs. The notation " - " in the I-niceSO column indicates that I-niceSO did not find the correct number of clusters. We posit that I-niceSO and I-niceMO obtained higher values of purity and ARI than randomly selected initial cluster centres, the *k*-means++ method and the MMCA algorithm. For the datasets with few classes, I-niceSO and I-niceMO produced results with the same clustering accuracy as those produced by the *k*-means algorithm using the true class centres as the initial cluster centres. For the datasets with more classes, I-niceMO produced even better results than *k*-means when using the true class centres as the initial cluster centres. Therefore, identification of the correct number of clusters and the correct initial cluster centres by both I-niceSO and I-niceMO can improve the clustering accuracy.

Table 10
Comparison of running times for identifying the number of clusters and the initial cluster centres and generating clustering results with the I-nice method and selecting initial cluster centres and generating clustering results with MMCA on a few synthetic and real-world datasets.

Datasets	Instances	Clusters	Features	Running Time (Hour:Minute:Second)	
				I-nice	MMCA
Appendicitis	106	2	7	00:00:18	00:02:25
Iris	150	3	4	00:00:30	00:03:24
Data11	270	4	2	00:00:53	00:07:51
Data5	500	5	2	00:00:27	00:26:48
Glass	214	6	10	00:00:20	00:22:37
Data9	700	7	2	00:00:57	00:55:15
DS1	1200	15	2	00:02:54	04:21:41
DS2	1300	17	2	00:04:05	05:21:30

In Table 10, we compare the running times of two algorithms of similar type, namely, I-nice and MMCA, on a few synthetic and real-world datasets. I-nice is proposed for identifying the number of clusters and the initial cluster centres, which are used to generate the clustering results, whereas MMCA is applied to select the initial cluster centres for the specified number of clusters. Then, the initial cluster centres are used to generate the clustering results. The results show that the MMCA algorithm is very slow in selecting the initial cluster centres from large datasets.

7. Conclusions and future work

In this paper, we have presented two parameter-free clustering algorithms, namely, I-niceSO and I-niceMO, which can automatically find the number of clusters in data and identify the initial cluster centres. A set of observation points are allocated in the data space, and the high-dimensional input data are transformed to a set of distance distributions. Gamma mixture models are used to model the distance data, and the EM algorithm is used to solve the GMMs. Akaike information criterion AICc is used to select the best-fitted model with the most components. The number of components of the selected GMM is taken as the number of clusters in the data. The I-niceSO algorithm explores the number of clusters and the initial cluster centres from the selected final model, while I-niceMO uses all best-fitted models to identify the initial cluster centres and the number of clusters in the *k*-means algorithm to improve the clustering accuracy and efficiency.

We have presented experimental results on both synthetic and real-world datasets and shown that the proposed algorithms significantly outperformed two widely used methods, namely, Elbow and Silhouette, in finding the correct number of clusters in data. The clustering results obtained using the proposed method were better than the clustering results obtained using the *k*-means++ method, MMCA and randomly selected initial cluster centres. We have also shown that the accuracy of clustering using I-niceSO-identified initial cluster centres was similar to the accuracy of clustering using true class centres. In addition, the I-niceMO algorithm improved the clustering accuracy in the *k*-means clustering process.

Since the number of clusters and the set of initial cluster centres are automatically found, the I-nice clustering algorithms are easy to use because no parameter is required in advance. Furthermore, the use of the automatically identified initial cluster centres reduces the number of iterations in the *k*-means clustering process. Therefore, the proposed algorithms are effective and efficient for cluster analysis.

The main weakness of the I-niceSO algorithm is that it is sensitive to observation points. This problem may be overcome by an effective observation point selection method. The main weakness of I-niceMO is that it is not effective on imbalanced datasets that contain majority and minority clusters.

The current research on selecting initial cluster centres from imbalanced datasets can be improved. Our future research will focus on extending the I-nice approach to a semi-supervised clustering model using distance distributions of multiple effective observation points.

Acknowledgements

This paper was supported by National Natural Science Foundation of China under Grant no. 61473194 and Shenzhen-Hong Kong Technology Cooperation Foundation under Grant no. SGLH20161209101100926.

References

[1] J. Alcalafdez, A. Fernandez, J. Luengo, J. Derrac, S. Garcia, KEEL data-mining software tool: data set repository, integration of algorithms and experimental analysis framework, Soft Comput. 17 (2011) 255–287.
[2] H. Akaike, Information theory and an extension of the maximum likelihood principle, in: B.N. Petrov, Csaki (Eds.), Proceeding of the Second International Symposium on Information Theory, Academia Kiado, Budapest, 1973, pp. 267–281.
[3] O. Arbelaitz, I. Gurrutxag, J. Muguerza, J.M. Perez, I. Perona, An extensive comparative study of cluster validity indices, Pattern Recognit. 46 (2013) 243–256.

[4] D. Arthur, S. Vassilvitskii, K-means++: the advantages of careful seeding, in: Proceeding of the Symposium on Discrete Algorithms (SODA), Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.
[5] S. Atapattu, C. Tellambura, H. Jiang, A mixture gamma distribution to model the SNR of wireless channels, IEEE Trans. Wireless Commun. 10 (12) (2011) 4193–4203.
[6] J.D. Banfield, A.E. Raftery, Model-based gaussian and non-gaussian clustering, Biometrics 49 (3) (1993) 803–821.
[7] J.C. Bezdek, N.R. Pal, Some new indexes of cluster validity, Syst. Man Cybern. 28 (3) (1998) 301–315.
[8] P.S. Bradley, U.M. Fayyad, Refining initial points for *K*-means clustering, in: Proceedings of the Fifteenth International Conference on Machine Learning, Morgan Kaufmann, San Francisco, 1998, pp. 91–99.
[9] L. Breiman, Bias, variance, and arcing classifiers, Technical report-460, Statistics Department, University of California, Berkeley, CA, USA, 1996.
[10] K.P. Burnham, D.R. Anderson, Model Selection and Multimodel Inference : A Practical Information-Theoretic Approach, Springer-Verlag New York, Inc., New York, USA, 2002.
[11] K.P. Burnham, D.R. Anderson, K.P. Huyvaert, AIC model selection and multimodel inference in behavioral ecology: some background, observations, and comparisons, Behav. Ecol. Sociobiol. (Print) 65 (1) (2011) 23–35.
[12] H. Chang, D. Yeung, Robust path-based spectral clustering, Pattern Recognit. 41 (1) (2008) 191–203.
[13] R.C. De Amorim, C. Hennig, Recovering the number of clusters in data sets with noise features using feature rescaling factors, Inf. Sci. (Ny) 324 (2015) 126–145.
[14] S. Deelers, S. Auwatanamongkol, Enhancing *k*-means algorithm with initial cluster centers derived from data partitioning along the data axis with the highest variance, Int. J. Phys. Math. Sci. 1 (11) (2007) 518–523.
[15] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. Royal Stat. Soc. Ser. B (Stat. Methodol.) 39 (1) (1977) 1–38.
[16] J.E. Dennis, R.B. Schnabel, Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Prentice-Hall, Englewood Cliffs, NJ, 1983.
[17] M.A.T. Figueiredo, A.K. Jain, Unsupervised learning of finite mixture models, IEEE Trans. Pattern Anal. Mach. Intell. 24 (3) (2002) 381–396.
[18] C. Fraley, A.E. Raftery, Model-based clustering, discriminant analysis, and density estimation, J. Am. Stat. Assoc. 97 (458) (2002) 611–631.
[19] C. Fraley, A.E. Raftery, Bayesian regularization for normal mixture estimation and model-based clustering, J. Classif. 24 (2) (2007) 155–181.
[20] A. Gionis, H. Mannila, P. Tsaparas, Clustering aggregation, ACM Trans. Knowl. Discov. Data 1 (1) (2007) 1–30.
[21] C. Hennig, T.F. Liao, How to find an appropriate clustering for mixed type variables with application to socio-economic stratification, J. Royal Stat. Soc. Ser. C (Appl. Stat.) 62 (3) (2013) 309–369.
[22] X. Hu, L. Xu, Automatic cluster number determination via BYY harmony learning, in: Advances in Neural Networks, Springer, Berlin Heidelberg, 2004, pp. 828–833. ISNN 2004, LNCS 3173.
[23] L. Hubert, P. Arabie, Comparing partitions, J. Classif. 2 (1) (1985) 193–218.
[24] C.M. Hurvich, C. Tsai, Regression and time series model selection in small samples, Biometrika 76 (2) (1989) 297–307.
[25] A.K. Jain, R.C. Dubes, Algorithms for Clustering Data, Prentice Hall, Englewood Cliffs, NJ,, 1998.
[26] R. Jain, A. Koronios, Innovation in the cluster validating techniques, Fuzzy Optim. Decis. Mak. 7 (3) (2008) 233–241.
[27] D.J. Ketchen, C.L. Shook, The application of cluster analysis in strategic management research: an analysis and critique, Strateg. Manag. J. 17 (6) (1996) 441–458.
[28] S.S. Khan, A. Ahmad, Cluster center initialization algorithm for *k*-means clustering, Pattern Recognit. Lett. 25 (11) (2004) 1293–1302.
[29] J. Lee, S. Olafsson, A meta-learning approach for determining the number of clusters with consideration of nearest neighbors, Inf. Sci. (Ny) 232 (2013) 208–224.
[30] M. Lichman, UCI machine learning repository, School of Information and Computer Sciences, University of California, Irvine, 2013 (Master's thesis). http://archive.ics.uci.edu/ml.
[31] C.D. Manning, P. Raghavan, H. Schutze, Introduction to Information Retrieval, Cambridge University Press, New York, 2008.
[32] G.W. Milligan, M.C. Cooper, An examination of procedure for determining the number of clusters in a datasets, Psychometrika 50 (2) (1985) 159–179.
[33] S. Race, C.D. Meyer, K. Valakuzhy, Determining the number of clusters via iterative consensus clustering, in: Proceedings of the SIAM International Conference on Data Mining, 2014, pp. 94–102.
[34] A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks, Science 344 (6191) (2014) 1492–1496.
[35] P.J. Rousseeuw, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, J. Comput. Appl. Math. 20 (1) (1987) 53–65.
[36] M.P. Sbhatia, D. Khurana, Analysis of initial centers for *k*-means clustering algorithm, Int. J. Comput. Appl. 71 (5) (2013) 9–12.
[37] G. Schwarz, Estimating the dimension of a model, Ann. Stat. 6 (2) (1978) 461–464.
[38] J. Shen, S.I. Chang, E.S. Lee, Y. Deng, S.J. Brown, Determination of cluster number in clustering microarray data, Appl. Math. Comput. 169 (2) (2005) 1172–1185.
[39] N. Sugiura, Further analysis of the data by Akaike's information criterion and the finite corrections, Commun. Stat. Theory Methods 7 (1) (1978) 13–26.
[40] R.L. Thorndike, Who belongs in the family, Psychometrika 18 (4) (1953) 267–276.
[41] R. Tibshirani, G. Walther, T. Hastie, Estimating the number of clusters in a data set via the gap statistic, J. Royal Stat. Soc. Ser. B (Stat. Methodol.) 63 (2) (2001) 411–423.
[42] C.W. Tsai, W.L. Chen, M.C. Chinag, A modified multiobjectives EA-based clustering algorithm with automatic determination of the number of clusters, in: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, 2012, pp. 14–17. October.
[43] G. Vegas-Sáchez-Ferrero, J. Seabra, O. Rodriguez-Leor, A. Serrano-Vida, S. Aja-Fernández, C. Palencia, M. Martín-Fernández, J. Sanches, Gamma mixture classifier for plaque detection in intravascular ultrasonic images, IEEE Trans. Ultrason. Ferroelectr Freq. Control 61 (1) (2014) 44–61.
[44] G. Vegas-Sánchez-Ferrero, M. Martín-Fernández, J.M. Sanches, A gamma mixture model for IVUS imaging, in: Proceedings of the Multi-Modality Atherosclerosis Imaging and Diagnosis, 2014, pp. 155–171.
[45] L. Wang, C. Leckie, K. Ramamohanarao, J. Bezdek, Automatically determining the number of clusters in unlabeled data sets, IEEE Trans. Knowl. Data Eng. 21 (3) (2009) 335–350.
[46] A.R. Webb, Gamma mixture models for target recognition, Pattern Recognit. 33 (12) (2000) 2045–2054.
[47] Y. Ye, J.Z. Huang, X. Chen, S. Zhou, G. Williams, X. Xu, Neighborhood density method for selecting initial cluster centers in *K*-means clustering, in: Proceedings of the Tenth Pacific-Asia Conference, PAKDD, Springer-Verlag, Berlin Heidelberg, 2006, pp. 189–198. Singapore, LNAI.
[48] R.R. Yager, D. Filev, Approximate clustering via the mountain method, IEEE Trans. Syst. Man Cybern. 24 (8) (1994) 1279–1284.
[49] M. Yang, K. Wu, A modified mountain clustering algorithm, Pattern Anal. Appl. 8 (1) (2005) 125–138.
[50] C.T. Zahn, Graph-theoretical methods for detecting and describing gestalt clusters, IEEE Trans. Comput. 20 (1) (1971) 68–86.