

A new clustering algorithm based on a radar scanning strategy with applications to machine learning data

Lin Ma^a, Yi Zhang^b, Víctor Leiva^{c,*}, Shuangzhe Liu^d, Tiefeng Ma^a

^a School of Statistics, Southwestern University of Finance and Economics, China

^b College of Electrical Engineering and Automation, Fuzhou University, China

^c School of Industrial Engineering, Pontificia Universidad Católica de Valparaíso, Chile

^d Faculty of Science and Technology, University of Canberra, Australia

ARTICLE INFO

Keywords:

Adaptive clustering
Artificial intelligence
Greedy algorithm
High dimensionality
Probability density function

ABSTRACT

In this paper, we propose a novel density-based radar scanning clustering algorithm. Its main objective is to quickly discover and accurately extract individual clusters by employing the radar scanning strategy. By using this algorithm, the number of clusters does not need to be specified beforehand. Two techniques are utilized in our proposed method. First, we use a fast mean-shift algorithm with adaptive radius and active subsets to effectively locate the centers, reducing the computational time significantly. Second, we employ the shape of the probability density function of the distribution of distances between a selected point and the other points in the data set. This is performed to determine the critical parameters of the radiuses of the fast mean-shift algorithm and radiuses of clusters. The new algorithm has four merits. It reduces the computational complexity, overcomes problems caused by high dimensionality, is capable of dealing with heterogeneous spherical data sets, and lastly, is robust to noise and outliers. After applying our proposed method to several kinds of synthetic and real-world data sets, the results indicate that the density-based radar scanning algorithm is efficient and accurate.

1. Introduction

In this section, we provide background on related work to the present investigation, define the notations, and describe the structure of the paper.

1.1. Background

Clustering is a frequently used technique in the fields of artificial intelligence, big data analytics, computational statistics, data mining, image processing, machine learning, pattern recognition, and vector quantization (El-Shafeiy, Sallam, Chakraborty, & Abohany, 2021; Jain, Murty, & Flynn, 1999; Martin-Barreiro, Ramirez-Figueroa, Cabezas, Leiva, & Galindo-Villardón, 2021; Thrun & Ultsch, 2021).

Clustering aims to reduce the size of the data set by grouping similar data items together (Cheng, 1995). Therefore, it is a helpful technique for reducing the size of a data set by categorizing a large number of observations into a smaller number of subgroups. When clustering is applied, the observations in each subgroup share similar characteristics (Kile & Uhlen, 2012).

With the soaring demand for clustering in practical applications, several new adaptive and refined algorithms based on classical methods have been proposed. These methods can be categorized into five types: (i) partition-based; (ii) hierarchical; (iii) model-based; (iv) grid-based; and (v) density-based (Frossyniotis, Pertselakis, & Stafylopatis, 2002).

In the partition-based methods, the objective function is minimized by partitioning data sets utilizing a distance measure. Among the partition-based methods, k-means is the most known and popular technique because of its efficiency and conciseness. However, there are some limitations with the k-means algorithm: (i) the number of clusters must be previously specified by users; (ii) the initial values always significantly impact the results due to the local convergence; and (iii) the Euclidean distance-based k-means-type algorithms tend to partition clusters into equal size, so it fails to handle heterogeneous clusters. Fig. 1 shows unsatisfactory situations of the k-means algorithm. To alleviate these situations, diverse improvements have been proposed (Arthur & Vassilvitskii, 2017; Dan & Moore, 2000; Huang et al., 2014; Kazemi & Boostani, 2021).

In hierarchical methods, the data set is organized in the form of a dendrogram or a tree. These methods can deal with different types

* Corresponding author. Víctor Leiva: victor.leiva@pucv.cl; victorleivasanchez@gmail.com; www.victorleiva.cl.

E-mail addresses: malin@mail.swufe.edu.cn (L. Ma), zhangyi@fzu.edu.cn (Y. Zhang), victor.leiva@pucv.cl (V. Leiva), shuangzhe.liu@canberra.edu.au (S. Liu), matiefeng@swufe.edu.cn (T. Ma).

<https://doi.org/10.1016/j.eswa.2021.116143>

Received 22 November 2020; Received in revised form 5 August 2021; Accepted 21 October 2021

Available online 16 November 2021

0957-4174/© 2021 Elsevier Ltd. All rights reserved.

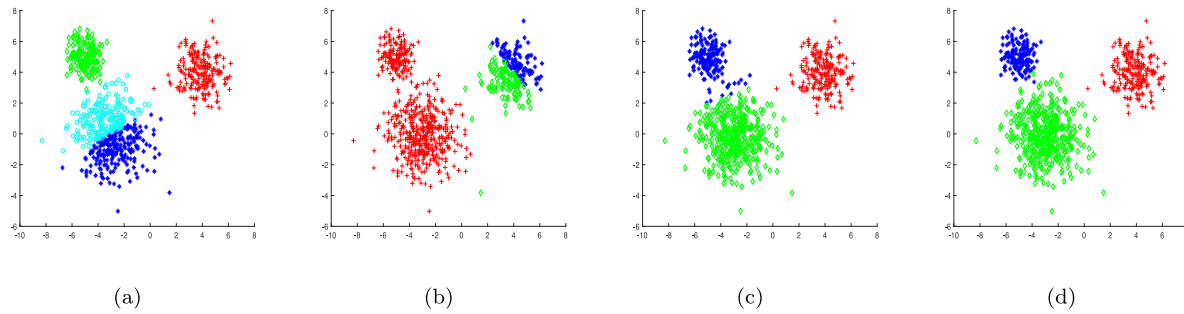


Fig. 1. Limitations of the k-means algorithm with (a) the wrong number of clusters; (b) poor initial points; (c) the right number of clusters and appropriate initial points, but poor assignment for boundaries; and (d) correct clustering conditions.

of clusters when suitable distance measures are applied. Nevertheless, three points need to be addressed when applying hierarchical methods: (i) the number of clusters also needs to be specified by users previously; (ii) the hierarchical process is time-consuming and space-consuming; and (iii) their results are greatly impacted by noise (Guan, Li, He, Zhu, & Chen, 2021; Zhong, Miao, & Franti, 2011).

In model-based methods, the data sets are assumed to follow a multivariate mixture Gaussian statistical distribution. Thus, this model-based clustering approach is also called the Gaussian mixture model (GMM) (Fraley & Raftery, 2002). Generally, the model-based methods may perform better when the clusters in the data sets are no longer homogeneous. Nonetheless, in model-based methods: (i) the number of clusters still needs to be provided by users previously; and (ii) due to the application of the expectation-maximization algorithm, it may fail to find a global optimal for the distribution parameters. Model-based algorithms are presented in Akogul and Eriş (2017) and Andrews (2018).

In grid-based methods, the data sets are segmented into uniform cells, which are then treated as a whole, thereby speeding up the clustering process to a great extent in low dimensions. However, in high dimension cases, the number of cells increases exponentially. Additionally, the length of intervals is hard to determine. In general, it acts as a preprocessor for other methods such as partition-based (Wang & Yao, 2018), mathematical morphology (Wang, 2019), and especially density-based (Chen et al., 2020) techniques.

Compared to the four types of clustering methods above, the main advantage of the density-based methods lies in their universality. Note that the density-based methods can detect arbitrarily shaped clusters and automatically discover the number of clusters whilst identifying noise. For this reason, several kinds of density-based methods and their variants have been proposed recently. Among them, the density-based spatial clustering algorithm with noise (DBSCAN) as well as the density peak clustering (DPC) and mean-shift algorithms are the most considered.

Firstly, the DBSCAN (Ester, Kriegel, Sander, & Xu, 1996) is a connection-oriented method that extends core points to clusters and is robust to noise. Nevertheless, the DBSCAN is parameter-sensitive and time-consuming. Due to its sensitivity, a method named ADBSCAN was proposed to determine neighborhood radius values automatically (Li, Liu, Li, & Gan, 2020). In addition, an algorithm named FEM-DBSCAN divides the feature space into subspaces, on which the DBSCANs with different parameters are applied. The FEM-DBSCAN can handle heterogeneous clusters while being applied to big data sets (Aykroyd, Leiva, & Ruggeri, 2019; Kazemi & Boostani, 2021). These methods may be used to deal with the problem of mismatching between global parameters and local densities. Because of its time complexity, grid methods (Chen et al., 2020) and group strategies (Gholizadeh, Saadatfar, & Hanafi, 2020) are embedded into the DBSCAN algorithm to improve its efficiency.

The DPC is another widely employed density-based algorithm. In its first phase, it uses subtle tips to find cluster centers, whose densities are local maximums and far away from the other local density peaks. In its second phase, the cluster centers are determined by a decision graph. However, the density must be carefully measured due to its significant impact on clustering results. In addition, the number of clusters must be decided manually. For the former problem, a refined measurement of density was constructed by employing multiple characteristics of trajectory data (Yang, Cai, Yang, Zhang, & Zhao, 2019). Similarly, a relative density of objects was established utilizing the k-nearest neighbors instead of using a given cut-off distance (Wang & Yang, 2021). For the latter problem, a fuzzy density peak k-medoids algorithm was proposed to automatically choose the number of clusters by constructing a more explicit decision graph (Liu, He, & Xu, 2017). Moreover, an algorithm that automatically detects clustering centers from the decision graph by integrating comprehensive metrics and distances between potential clustering centers was proposed (Min, Huang, & Sheng, 2020).

On this same line, the mean-shift algorithm is also one of the best well-known density-based methods, which detects the clusters by a hill-climbing strategy (Cheng, 1995). When converging, sample points shifting to the same local density peak are labeled as a cluster. Nonetheless, the bandwidth or radius is estimated in advance, which greatly influences the results. The adaptive method is used to determine an effective bandwidth or radius by employing density heat (Chen et al., 2018), cross-validation, and plugging in bandwidth selectors. Under the assumption that clusters are dense areas surrounded by low-density (sparse) areas, the logic of the valley seeking algorithm is similar to the mean-shift algorithm, which grows a cluster by descending from a density peak to the valleys (Laohakiat & Saing, 2021).

Note that density-based methods refine the existing algorithms. For example, the k-means algorithm based on density canopy (DCk-means) was utilized in Zhang, Zhang, and Zhang (2018) to determine the number of clusters and the position of initial values simultaneously. Moreover, projection-based clustering is one of the best conventional algorithm to cluster (Thrun & Ultsch, 2020). For this reason, density-based methods cooperate with the projection scheme. A projection-based split-and-merge clustering algorithm acts to decide whether two parts should be merged or split and is embedded to diverse classical algorithms (Cheng, Ma, & Liu, 2019).

Although density-based methods are well developed and widely used, they still suffer from some limitations. First, as stated above, the parameters in density measurement are critical so that they should be carefully determined. Furthermore, global parameters may not fit the local densities well, and thereby fail to deal with heterogeneous clusters. Lastly, the computational complexity of most density-based methods is $O(N^2)$ or greater, where N is the number of sample points in the data set.

Based on this detailed bibliographical review, we note there exists an opportunity to improve the density-based clustering algorithms

widely used today in different frameworks for data science and related areas.

The objective of this paper is to propose and derive a novel clustering algorithm by employing density-based radar scanning (DBRS) and the statistical distribution of distances between a selected point and the rest of the points in the data set. This new algorithm is a greedy method that does not produce an optimal solution (Cabezas, Garcia, Martin-Barreiro, Delgado, & Leiva, 2021; Martin-Barreiro, Ramirez-Figueroa, Nieto-Librero, et al., 2021; Ramirez-Figueroa, Martin-Barreiro, Nieto-Librero, Leiva, & Galindo-Villardón, 2021). However, a greedy algorithm solves a problem heuristically, making the locally optimal choice at each stage, and attaining locally optimal solutions approximating a globally optimal solution in a reasonable amount of time.

Note that a density represents the aggregation of sample points, that is, how many points per unit area are there. In this way, the relative position of points, a fundamental factor of clustering, is eliminated. In this paper, we propose and derive a novel density-based method using the probability density function (PDF) of the distribution of distances between a selected point and the rest of the points in the data set. Note that this approach reflects not only the density of a single area but also the relative position of points. Thus, the affiliation of a point is decided by the scattering of points around it. In our proposal, the PDF projects the data set into a one-dimensional space with minimum information loss, saving time consumption as the dimension increases. The features of the PDF (the location of the first peak and the first valley rather than a fixed quantile) are helpful data-driven parameters of proper radius, active subsets for fast mean-shift, and radiuses of clusters.

Our DBRS clustering algorithm is analogous to electromagnetic pulses emitted by radars and determines some crucial parameters with the following four advantages:

- (i) It determines the number of clusters in an adaptive way, similar to the DBSCAN algorithm. It quickly locates a center, determines the corresponding radius and then extracts the cluster. This process is done recursively until all the clusters are extracted. Thus, the number of clusters does not need to be specified by the user.
- (ii) As it employs the PDF of the statistical distribution of distances, the radar scanning strategy maps the data set into a one-dimensional space so that it is more applicable to high-dimension situations.
- (iii) It allows the radiuses of clusters to be determined by the distribution of distances rather than a fixed threshold. As a result, the DBRS algorithm can handle heterogeneous data sets well.
- (iv) Last but not least, during the clustering process, the outliers and noise are recognized by the distribution of distances. Hence, according to that, the outliers and noise are always left in the end so that the DBRS algorithm is non-sensitive (robust) to noise nor outliers.

1.2. Notation and organization

Table 1 presents the notations to be used in this paper. After this introduction and definition of notations, the paper is organized as follows. In Section 2, we introduce some preliminaries which facilitate the comprehension of our proposal. Section 3 describes the proposed DBRS algorithm. In Section 4, experimental results are conducted for synthetic data sets in order to evaluate the performance of our algorithm and compare it to some existing methods. Section 5 illustrates the DBRS algorithm with real data sets from a machine learning repository. Finally, we make some concluding remarks on the obtained results and provide ideas for possible future work in Section 6.

2. Preliminaries

In this section, we introduce some preliminaries which facilitate the comprehension of our proposal based on the PDF of the distribution of distances and the mean-shift algorithm for clustering.

2.1. Statistical distribution of distances

Distance is an important factor for clustering. In partition-based clustering methods, the distance is used to determine the affiliation of sample points. In hierarchical clustering methods, it is employed to measure whether two groups should be merged or not. In density-based clustering methods, the distance is utilized to measure the density as well as to move noise (Li et al., 2020). Thus, studying the statistical distribution of distances is meaningful. Assume a spherical data set $D \in \mathbb{R}^q$ follows the GMM if each cluster follows a multivariate Gaussian distribution in the q -dimensional Euclidean space.

First, a condition with only one cluster is studied. The sample point x comes from the distribution $X \sim N_q(\mu, \sigma^2 E_q)$, where the mean μ is a q -dimensional vector and E_q is the q -dimensional identity matrix, with $\sigma^2 E_q$ being the corresponding covariance matrix. For a given arbitrary point a in a q -dimensional space, the random variable $Y = \|X - a\|^2$ follows a non-central χ^2 distribution (Díaz-García & Leiva, 2003) with PDF stated as

$$f_Y(y) = \frac{1}{2\sigma^2} \left(\frac{y}{\delta}\right)^{\frac{q-2}{4}} \exp\left(-\frac{(y+\delta)}{2\sigma^2}\right) I_{\frac{q}{2}-1}\left(\frac{\sqrt{y\delta}}{\sigma^2}\right), \quad y \geq 0, \quad (1)$$

where $\delta = \|\mu - a\|^2$ is the non-centrality parameter, and I_ν is the modified Bessel function of first kind of order ν . The random variable Y represents the squared Euclidean distance between the given point a and other points in D .

The sample points come from K different Gaussian distributions. Without loss of generality, suppose that the GMM, corresponding to the mixture random vector, X' namely, with K different Gaussian distributions, is mixed in terms of $\{X_1, \dots, X_K\}$, where $X_k \sim N_q(\mu_k, \sigma_k^2 E_q)$, for $k \in \{1, \dots, K\}$. Hence, $Y_k = \|X_k - a\|^2$ follows a χ^2 distribution with PDF f_{Y_k} being similar to that expressed in (1), but δ is replaced with $\delta_k = \|\mu_k - a\|^2$ and σ with σ_k . Therefore, $Y' = \|X' - a\|^2$ follows a mixed non-central χ^2 distribution with PDF given by

$$\begin{aligned} f_{Y'}(y) &= \sum_{k=1}^K p_k f_{Y_k}(y) \\ &= \sum_{k=1}^K p_k \frac{1}{2\sigma_k^2} \left(\frac{y}{\delta_k}\right)^{\frac{q-2}{4}} \exp\left(-\frac{(y+\delta_k)}{2\sigma_k^2}\right) I_{\frac{q}{2}-1}\left(\frac{\sqrt{y\delta_k}}{\sigma_k^2}\right), \end{aligned} \quad (2)$$

for $y \geq 0$ and $\sum_{k=1}^K p_k = 1$.

This means the distribution of squared Euclidean distances between a given point a and all of the other points in the data set follows the mixed non-central χ^2 model, which is a superposition of several non-central χ^2 distributions. For example, in Fig. 2(a), the blue dashed line represents the curve of the χ^2 PDF, given in (2), transformed from the squared Euclidean distance to the Euclidean distance. On this multimodal curve, the locations of the first peak and the first valley are important information, which implies the structure of the data set close to the given point a . Therefore, the proposed radar scanning strategy employs the mixed non-central χ^2 distribution by establishing the corresponding PDF, explained in detail in Section 3.

2.2. Mean-shift algorithm

Let $D \in \mathbb{R}^q$ be a finite set embedded in the q -dimensional Euclidean space. Let \mathcal{K} be a flat kernel, that is, the characteristic function of radius λ defined as

$$\mathcal{K}(x) = \begin{cases} 1, & \text{if } \|x\| \leq \lambda; \\ 0, & \text{if } \|x\| \geq \lambda. \end{cases} \quad (3)$$

The sample mean at x is stated as

$$m(x) = \frac{1}{\sum_{x_i \in D} \mathcal{K}(x_i - x)} \sum_{x_i \in D} \mathcal{K}(x_i - x) x_i. \quad (4)$$

Note that the difference $m(x) - x$ is called mean-shift, where $\mathcal{K}(x)$ and $m(x)$ are defined in (3) and (4), respectively. The repeated movement

Table 1

Notations and acronyms used in the present document.

Symbol	Acronym/Notation
$D \in \mathbb{R}^q$	A spherical data set D in the q -dimensional set of real numbers
N	Number of sample points in D
N_D	Number of data sets for the Friedman and F statistics
N_A	Number of algorithm/methods for the F statistic
D_s	A subset of D , which is the current data set in the process
$ D_s $	Size of the current data set D_s
q	Dimension of the sample points
a	A q -dimensional arbitrary point
X and X'	Two q -dimensional random vectors
x_i	Sample point i composed by (x_{i1}, \dots, x_{iq})
$d(x_i, x_j)$	Euclidean distance between x_i and x_j
$X \sim N_q(\mu, \sigma^2 E_q)$	X is q -variate normal distributed with mean μ and covariance matrix $\sigma^2 E_q$
E_q	The q -dimensional identity matrix
$Y = \ X - a\ ^2$	Squared Euclidean distance between an arbitrary a and other points in D
O	Center or local density peak in the original data set
\mathcal{K}	Kernel function
K	Right number of clusters
λ	Radius of the flat kernel
L	Number of clusters obtained by a clustering method
f_Y	PDF of the random variable Y
$f_{D_s, a}$	PDF of distances between the subset D_s and an arbitrary point a
C	Curve of PDF $f_{D_s, a}$
ARI	Adjusted rand index
BIC	Bayesian information criterion
cluster _{k}	A general cluster labeled by k
center _{k}	Center of cluster _{k}
N_k	Number of sample points in cluster _{k}
DBSCAN	Density-based spatial clustering algorithm with noise
DBRS	Density-based radar scanning
DCK-means	k-means based on density canopy
DPC	Density peak clustering
FS	F score
GMM	Gaussian mixture model
LenInt	Length of interval
N/A	Not applicable or not available
PDF	Probability density function
Pinitial	Initial point
RC _{k}	Radius of cluster k
RI	Rand index
Rpeak	First peak radius
RT	Running time
Rvalley	First valley radius
SD	Standard deviation

of data points to the sample means is named the mean-shift algorithm (Fukunaga & Hostetler, 1975). In each iteration of the algorithm, $x_i \rightarrow m(x_i)$ is performed for all $x_i \in D$ simultaneously. The mean-shift algorithm has been proposed as a method for clustering analysis. However, in clustering, there is no need to involve all the sample points in the iteration. Instead, only those representative points are randomly initialized and involved in the blurring iteration process (Cheng, 1995).

The efficiency of the mean-shift algorithm is greatly impacted by the number of representative points and radius λ . In this paper, for spherical clustering, we make three adaptations to the original method for efficiency. First, only one point is arbitrarily chosen at a time. Second, an active subset is selected. Third, the radius λ is determined by the radar scanning strategy.

3. The DBRS algorithm

In this section, we introduce the DBRS algorithm in detail, which runs in four phases. In the first phase, we locate a center O , choose a point a arbitrarily, and then achieve the most proper radius as well as an active subset for mean-shift corresponding to a . Thus, a fast mean-shift process is used to shift the arbitrarily-chosen point to a local density peak denoted as O . In the second phase, we extract the cluster centered at O by utilizing the distribution of distances among O and all the other points. Hence, we get the radius of the cluster whose center is O and extract this cluster. In the third phase, we make a judgment on whether one of the terminal conditions is reached. If this occurs, we

go to the fourth phase. If none of the terminal conditions are reached, we go back to the first phase. Therefore, in the fourth phase, we assign the unsettled points, which should be noise or the fringe points of the clusters formerly extracted. However, there are still details that should be carefully checked. For example, how outliers must be dealt with and how to save computational time. Next, we elaborate the DBRS algorithm.

3.1. Definitions

To describe the DBRS algorithm more coherently, we provide some definitions below.

Definition 1 (PDF of the distances). Suppose that D_s is a non-empty subset of D with $|D_s|$ sample points. There is no harm in supposing that $D_s \equiv \{x_1, \dots, x_{|D_s|}\}$. For a given sample point $a \in D_s$, the distances between a and each of the sample points in D_s , namely $\{d(a, x_1), \dots, d(a, x_{|D_s|})\}$, are used to estimate a PDF, denoted as $f_{D_s, a}$.

For example, all the sample points in Fig. 2(b) constitute D_s , the red diamond is the given sample point a , and $\{d(a, x_1), \dots, d(a, x_{|D_s|})\}$ are the distances between a and all sample points in D_s , which generate the histogram in Fig. 2(a). The histogram is smoothed as the red curve, which is an estimate of $f_{D_s, a}$. This PDF reflects how the data points scatter around a . Note that it shows the relative position between a and its nearest density peak.

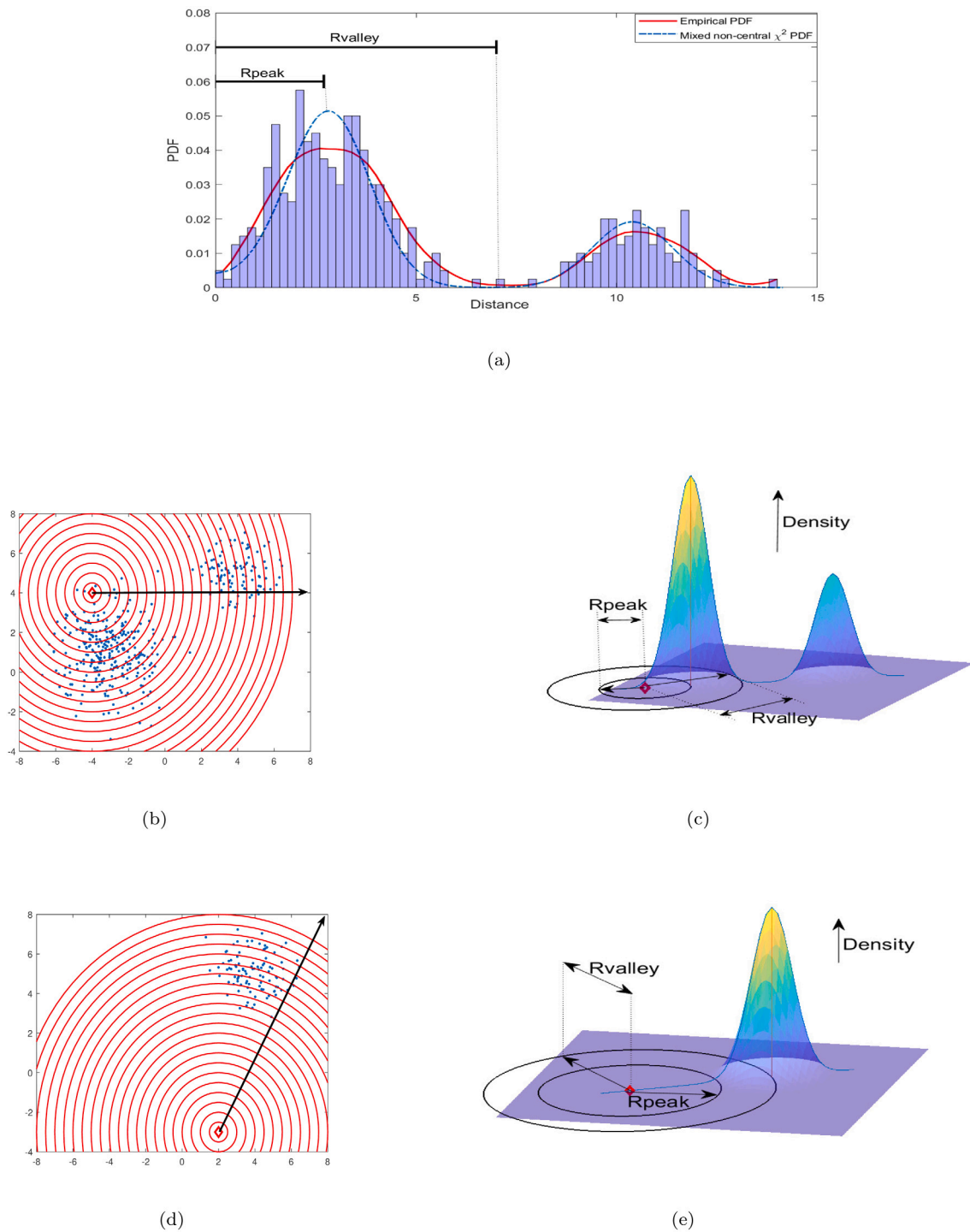


Fig. 2. (a) Formation of the PDF of distances and comparison between the empirical PDF of distances and the χ^2 PDF of distances; (b) the radar map showing the formation of the histogram displayed in (a), which is equivalent to mapping the points in the annulus to the corresponding intervals; (c) density of the data set shown in (b) and the corresponding R_{peak} and R_{valley} ; (d) the radar map of a debris point; and (e) density feature of the debris point shown in (d) and corresponding R_{peak} and R_{valley} .

Definition 2 (First peak radius – R_{peak} –). Suppose that the curve corresponding to the PDF of the distances is C , with its first peak being located at $(distance_{peak1}, frequency_{peak1})$ in C , where the first peak radius is fixed to $distance_{peak1}$; see Fig. 2(a).

Definition 3 (First valley radius – R_{valley} –). Suppose that the curve corresponding to the PDF of the distances is C , with its first valley being

located at $(distance_{valley1}, frequency_{valley1})$ in C , where the first valley radius is fixed to $distance_{valley1}$; see Fig. 2(a).

Definition 4 (Debris point). In a non-empty data set D_s , two kinds of points are defined as debris points: (i) the noise and (ii) the fringe points of clusters, which are leftover by previous extraction processes; see Fig. 2(d, e).

Definition 5 (Relative distance). Considering the heterogeneity of the clusters, the Euclidean distance is unreasonable to measure the closeness between sample points and clusters. Thus, a relative distance is considered between point i and cluster_k , which is defined as

$$d_{ik}^{\text{re}} = \frac{d(\mathbf{x}_i, \text{center}_k)}{\text{RC}_k}, \quad (5)$$

where center_k and RC_k are the center and radius of cluster_k , respectively.

As shown in Figs. 2(b) and 2(c), to display how $f_{D_s, a}$ is formed, we draw a series of concentric annulus. Intuitively, establishing the PDF of distances is equivalent to mapping the sample points in the same annular into the corresponding interval in the ray from the origin. The number of sample points in one interval (frequency) is defined as the values in the corresponding curve. With these frequencies, we can construct the histogram of distances. According to the histogram, the smoothed curve in red reflects the true PDF of the distances, that is, the curve of $f_{D_s, a}$; see Fig. 2(a). Note that the PDF established by the radar scanning strategy accurately reflects the location of Rpeak and Rvalley according to the mixed non-central χ^2 distribution, that is, the PDF curve is reasonable to determine suitable parameters. Debris points have two main characteristics. On the one hand, in our case, the density near the debris points are much less than the density of clusters. On the other hand, the distances between the debris points and other density peaks are large. Reflected in the PDF curve, these two features can be described as Rpeak being approximately equal to Rvalley in Figs. 2(d) and 2(e), respectively.

3.2. Algorithms

Fast mean-shift process. [Algorithm 1] The aim of this process is to identify a local density peak, namely one of the centers. In the beginning, we choose an initial point arbitrarily and then calculate the distances between the initial point and the rest of the points; see Fig. 2(a, b). Based on these distances, the corresponding PDF is established. The efficiency of the original mean-shift algorithm is greatly impacted by the number of representative points and the radius. In this paper, for spherical clustering, we make three adaptations to the original method for the sake of efficiency as indicated below:

- (i) We arbitrarily select one point at a time instead of involving several points. Note that a point will certainly shift to the center no matter where it starts due to the symmetry of the spherical clusters.
- (ii) An active subset is selected. In the original mean-shift process, all the sample points are included; see Fig. 3(c). However, the sample points which further away make no contribution to the hill-climbing process. Thus, in the fast mean-shift process, only the sample points in the active data set are involved in the hill-climbing process, which is superior to the original method in the spherical data set; see Fig. 3(a). The active subset consists of the points whose distances to the initial point are less than Rvalley ; see Fig. 2(a). According to the true distribution, the points located out of Rvalley are unlikely to belong to the present cluster.
- (iii) The radius is determined to be Rpeak . Using Rpeak as the radius, the mean quickly shifts to the nearest local density peak; see Figs. 3(b). Therefore, the computational time is saved significantly and the robustness increases.

Algorithm 1 Fast mean-shift process

Input: data set D_s , length of interval LenInt , initial point Pinitial
Output: A center
1: $|D_s| \leftarrow \text{size}(D_s, 1)$
2: **for** $i = 1; i \leq |D_s|$ **do**
3: $\text{InitialP_to_evP}(i) \leftarrow d(\text{Pinitial}, x_i)$
4: **end for**
5: $\text{n_of_intervals} \leftarrow \max(\text{InitialP_to_evP})/\text{LenInt}$
6: $\text{frequency}(i) \leftarrow \# \text{InitialP_to_evP} \in [i * \text{LenInt}, (i+1) * \text{LenInt}]$
7: $\text{S_frequency} \leftarrow \text{Smooth}(\text{frequency})$
8: $\text{Rpeak} \leftarrow \text{radius of the first peak}$
9: $\text{Rvalley} \leftarrow \text{radius of the first valley}$
10: $\text{Activesubset} \leftarrow \text{points} \in O(\text{Pinitial}, \text{Rvalley})$
11: $\text{center} \leftarrow \text{single point Mean-shift}(\text{Pinitial}, \text{Rpeak}) \in \text{Activesubset}$

Extract process. [Algorithm 2] A center is acquired in the fast mean-shift process. Furthermore, we intend to extract the corresponding cluster in this process. The PDF of the distances is still used in this phase. Once we calculate the distances between the center and all the points in the data set D_s , then from the PDF of distances, we can get Rvalley with respect to a center. Note that Rvalley is an approximate radius of the corresponding cluster, which may also be seen from the scatter diagram and its corresponding PDF of the distances; see Fig. 4(a). The points in the cycle centered at the previously acquired center and with radius equal to Rvalley (see Fig. 4(b)) are extracted and labeled as the same cluster.

Algorithm 2 Extract process

Input: D_s , LenInt , center gained by Algorithm 1
Output: renewed data set D_s , radius of cluster RC , potential cluster.
1: $|D_s| \leftarrow \text{size}(D_s, 1)$
2: **for** $i = 1; i \leq |D_s|$ **do**
3: $\text{center_to_evP}(i) \leftarrow d(\text{center}, x_i)$
4: **end for**
5: $\text{n_of_intervals} \leftarrow \max(\text{center_to_evP})/\text{LenInt}$
6: $\text{frequency}(i) \leftarrow \# \text{center_to_evP} \in [i * \text{LenInt}, (i+1) * \text{LenInt}]$
7: $\text{S_frequency} \leftarrow \text{Smooth}(\text{frequency})$
8: $\text{RC} \leftarrow \text{radius of the first valley}$
9: $\text{cluster} \leftarrow \text{points} \in O(\text{center}, \text{RC})$
10: $D_s \leftarrow \text{points} \notin O(\text{center}, \text{RC})$

Terminal conditions. The main process is recursive. The fast mean-shift and extraction processes are executed sequentially in every loop. In other words, a potential cluster is determined and removed in a loop. Hence, the number of clusters is dynamic with the main process. Moreover, the size of the remaining data set becomes smaller in every loop and so the computational cost is greatly reduced. After looping L times, all the clusters are removed from the data set. Therefore, the number of clusters is automatically set to be L . By this design, the number of clusters does not need to be an input parameter specified by users. When all the clusters are determined (that is, when there are no obvious clusters in the remaining data set), the main process should be terminated. Thus, we set up the following two terminal conditions (if one of them is achieved, the main process should be terminated):
[Condition 1] There are not enough sample points left in the remaining data set D_s (for example, 5% of the input data set D).
[Condition 2] Debris points are chosen as the initial point several times successively (for example, 5 times).

Adjust process. When one of the terminal conditions is achieved, the main steps of the algorithm are completed, but two problems remain. First, debris points may be leftover. Second, if two clusters overlap with each other, for the sake of accuracy, the affiliations of the points in the boundary area need to be carefully analyzed (Parmar et al., 2019). These two kinds of points should be reassigned to the existing clusters. The rule for reassignment should not simply rely on Euclidean distances

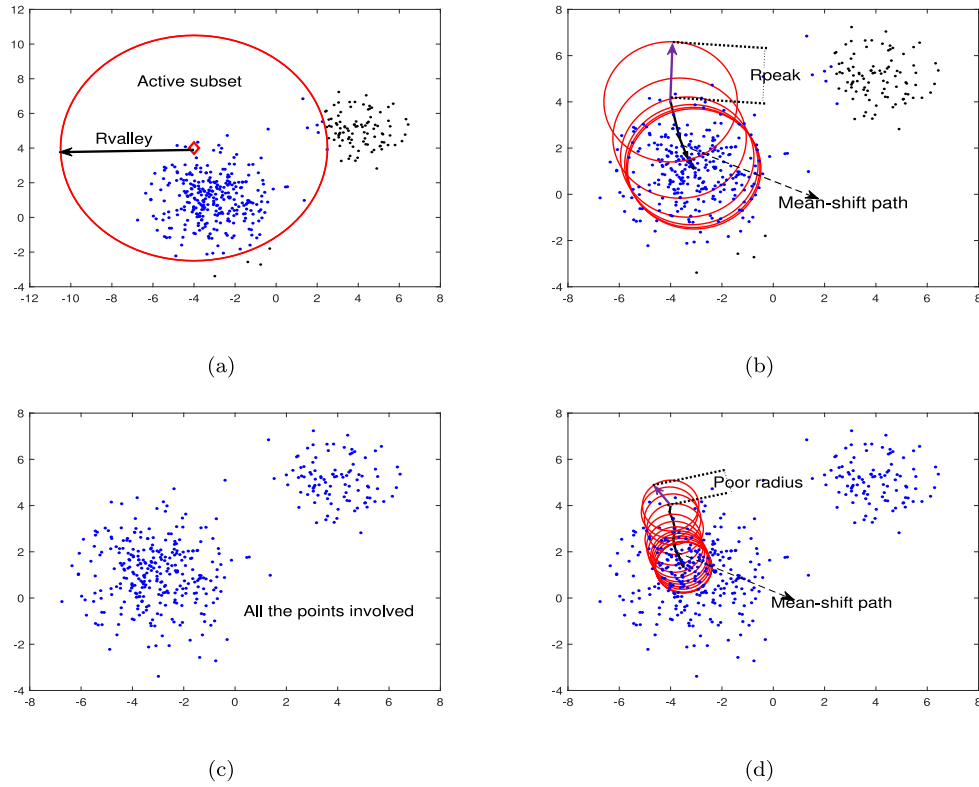


Fig. 3. (a) Active subset gained by the radar scanning technique, whose radius is R_{valley} shown in the PDF of distances; (b) fast mean-shift process with only one starting point and proper radius; (c) data set for original mean-shift with all point involved; and (d) original mean-shift with one starting point but poor radius.

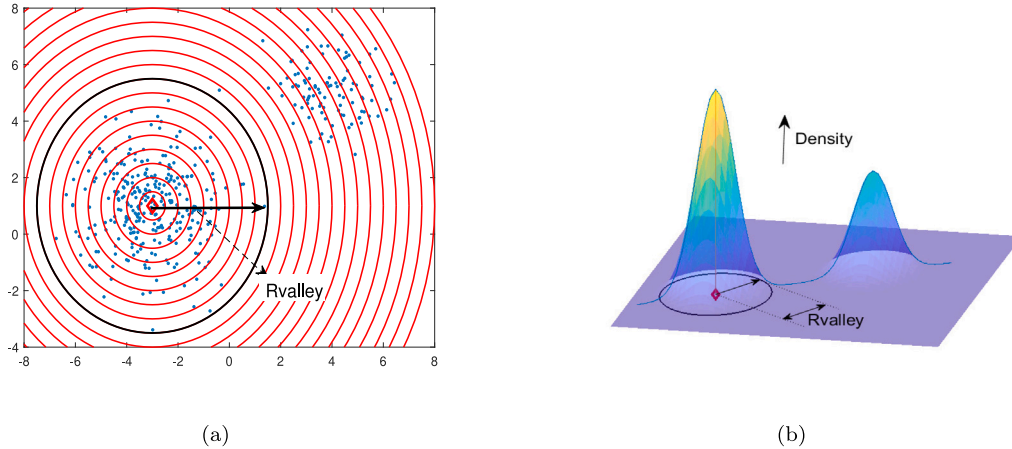


Fig. 4. (a) Radar map originated in one of the centers and radius for extracting the corresponding cluster; (b) density of the data shown in (a) and R_{valley} used for extraction.

between the unsettled sample points and the centers. Considering the heterogeneity of the clusters, we use the proposed relative distance as a measure. For the sample point x_i , if d_{ik}^{re} is the minimum among $\{d_{i1}^{re}, \dots, d_{iL}^{re}\}$, then the sample point should be assigned to $cluster_k$.

3.3. Main function and complexity analysis

We have provided all the modules of the proposed algorithm in the above subsection. In this subsection, we illustrate the algorithm from a global perspective. The main function is shown in Algorithm

3. Furthermore, we present the computational complexity analysis of the DBRS algorithm.

The computational complexity of the DBRS algorithm is comprised of the following three parts. The first part calculates the interval length for the PDF of the distances. We state that it is proportional to the span of the data set so that the time complexity is $O(N)$, where, as mentioned, N is the number of sample points in the data set. The second part is the fast mean-shift process with the complexity being $O(r + tr + sr^*)$, where the number of points in the current subset D_s is r . We have r distances to establish the PDF, and the complexity is r . In addition, t is associated with the corresponding PDF curve of the

Algorithm 3 Main function of the algorithm

Input: data set D
Output: labels of data points

```

1: LenInt  $\leftarrow$  usually proportional to the range of  $D$  (set it as a global variable)
2:  $D_s \leftarrow D$ ;  $k = 1$ ;
3: count = 0 (counting the times of outliers detected)
4: while  $\text{size}(D_s) > 0.05 * \text{size}(D)$  and count  $< 5$  do
5:   Pinitial  $\leftarrow$  a randomly-chosen sample point in  $D_s$ 
6:   if Pinitial is a debris point then
7:     count = count + 1; continue;
8:   else
9:     centerk  $\leftarrow$  fast mean-shift( $D_s$ , Pinitial) (Algorithm 1)
10:   end if
11:   [clusterk, RCk,  $D_s$ ]  $\leftarrow$  Extract( $D_s$ , centerk) (Algorithm 2)
12:   labels(clusterk)  $\leftarrow k$ ;
13: end while
14: labels(debris points)  $\leftarrow$  Adapt(debris points, centerk, RCk) -Eq. (5)-

```

distances which has been smoothed t times (here we always choose $t = 3$). Furthermore, s represents for the times of shifting step of the mean, and r^* is the number of sample points in the active subset. The third part is the extraction process with time complexity $O(r + sr + r)$, where the first two items are the same as the second part, and the third r stands for extracting and deciding which point should be extracted in the first loop, with $r = |D| = N$. In the process, one cluster is handled as the second and third parts are executed once. After the first cluster is extracted, the size of the remaining data set r is less than N and continues decreasing with every loop. Therefore, the time complexity of the circulatory process while handling all the clusters is $O(cLN)$, where, as mentioned, L stands for the number of clusters found by the DBRS algorithm and c is a small number. In addition, the time complexity of the adjustment process is $O(Lr^{**})$, with r^{**} being the number of sample points left over, which is negligible. Therefore, the total time complexity of the DBRS algorithm is $O(cLN)$.

4. Numerical experiments with synthetic data

This section contains four parts. First, we generate several synthetic data sets, which are different in size, number of clusters, and dimension of the samples. Some of these data sets include heterogeneous clusters. Second, we introduce different clustering methods, considering Dck-means (Zhang et al., 2018), X-means (Dan & Moore, 2000), GMM and k-means++ (Arthur & Vassilvitskii, 2017) as comparison. Moreover, the parameter setting for these algorithms is illustrated. Third, we compare the performance of the DBRS algorithm with the four mentioned algorithms on spherical clusters. Last, we add different ratios of noise to both data sets with homogeneous and heterogeneous clusters to test the ability of anti-noise.

4.1. Synthetic data sets

To verify that the DBRS algorithm is capable of handling heterogeneous clusters and determining the right number of clusters, we generate different data sets which vary in size, number of clusters, and dimension of sample points. These data sets may include heterogeneous clusters. Table 2 reports the features of the synthetic data sets. To increase the generality of the model, the number of sample points in data sets 1, 2, 5, 6 are generated from Gaussian distributions, where the size in Table 2 is the expected number of sample points. To provide more coherent information on the data sets, we use data sets 3 and 6 as examples.

Data set 3. It is composed of six bivariate Gaussian distributions. The components are:

$$\begin{aligned}
 X_i &\sim N_2 \left(\begin{bmatrix} 4 \\ 5 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right), \quad i \in \{1, \dots, 100\}; \\
 X_i &\sim N_2 \left(\begin{bmatrix} -3 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 & 0.2 \\ 0.2 & 2 \end{bmatrix} \right), \quad i \in \{101, \dots, 400\}; \\
 X_i &\sim N_2 \left(\begin{bmatrix} -5 \\ 8 \end{bmatrix}, \begin{bmatrix} 0.6 & 0.2 \\ 0.2 & 0.6 \end{bmatrix} \right), \quad i \in \{401, \dots, 800\}; \\
 X_i &\sim N_2 \left(\begin{bmatrix} 11 \\ 11 \end{bmatrix}, \begin{bmatrix} 3 & 0.2 \\ 0.2 & 3 \end{bmatrix} \right), \quad i \in \{801, \dots, 1800\}; \\
 X_i &\sim N_2 \left(\begin{bmatrix} 6 \\ -1 \end{bmatrix}, \begin{bmatrix} 0.6 & 0.2 \\ 0.2 & 1 \end{bmatrix} \right), \quad i \in \{1801, \dots, 2300\}; \\
 X_i &\sim N_2 \left(\begin{bmatrix} -10 \\ 3 \end{bmatrix}, \begin{bmatrix} 0.6 & 0 \\ 0 & 0.3 \end{bmatrix} \right), \quad i \in \{2301, \dots, 2600\}.
 \end{aligned}$$

Data set 5. It is composed of 15 homogeneous clusters. The numbers of sample points in each cluster, N_k namely, are generated from $N_k \sim N(200, 100)$, with covariance equal to 0.5, for $k \in \{1, \dots, 15\}$, and means being arbitrarily located in the square area $[(-20, -20), (20, 20)]$.

4.2. Comparative algorithms and parameter setting

The Dck-means algorithm is an improvement on initial points for the k-means algorithm. It can determine the number of clusters automatically. We compare it with the DBRS algorithm directly. The X-means algorithm needs two parameters: k_{\min} , the minimum of clusters expected, and k_{\max} , the maximum of clusters expected. Once these two parameters are given, the X-means algorithm stops when the value of the Bayesian information criterion (BIC) does not increase further or k_{\max} is reached. It ensures the fairness of comparison between the DBRS clustering and X-means algorithms to set k_{\min} equal to two and k_{\max} equal to the right number of clusters. Similarly, when using the GMM, the number of clusters must be specified beforehand. Thus, adjustments must be made when comparing the DBRS algorithm and the GMM. It is common to determine the number of clusters by the BIC. Hence, we change the number of clusters from two and set it to be slightly greater than the right number of clusters. From these results, we pick the cluster result with the smallest BIC value for the GMM to compare with other methods.

Note that the k-means++ algorithm also needs the number of clusters to be an input. Then, we design the experiments as follows. Assuming that K is the right number of clusters, we set the number of clusters to be inputted in the k-means++ algorithm equals to $K_1, K_1 + 1, \dots, K, K + 1, \dots, K_2 - 1, K_2$, where K_1 is an integer less than K , and K_2 is an integer greater than K . Hence, we compare the overall results of the k-means++ and DBRS algorithms from a holistic perspective.

4.3. Results on synthetic data sets and comparison

The commonly accepted evaluation criteria for clustering are the adjusted rand index (ARI), F score (FS), rand index (RI), and running time (RT). We apply these criteria to evaluate the results among the five clustering methods compared here.

The comparison of the DBRS algorithm to the Dck-means, GMM and X-means algorithms is presented in Table 3. The experiments are repeated 20 times for all data sets and then average values and 95% confidence intervals of the criteria are reported. The results show that the DBRS algorithm outperforms the other three methods for most synthetic data sets.

To show that the performance of the compared algorithms are significantly different, we use the Friedman and F tests (Laohakiat & Saing, 2021). The null hypothesis is that the performances of all algorithms are not different. The Friedman statistic is calculated by

$$\tilde{\chi}^2 = \frac{12N_D}{N_A(N_A + 1)} \left(\sum_{j=1}^{N_A} R_j^2 - \frac{N_A(N_A + 1)^2}{4} \right), \quad (6)$$

Table 2
Synthetic data sets.

Data set	Size (N)	Dimension (q)	Number of clusters (K)	Homogeneous	Overlapping
1	300	2	3	Yes	No
2	1500	2	3	Yes	Yes
3	2600	2	6	No	No
4	1400	2	6	No	Yes
5	3000	2	15	Yes	No
6	4500	2	15	No	N/A
7	2000	3	4	Yes	Yes
8	1200	3	4	No	No
9	15000	10	5	Yes	No
10	12000	10	5	No	N/A
11	8000	100	3	No	No
12	15000	100	5	No	N/A

where R_j is the average rank of the algorithms, N_D is the number of data sets, and N_A is the number of algorithms to be compared. Then, we use the F statistic (Demiar & Schuurmans, 2006) stated as

$$F = \frac{(N_D - 1)\tilde{\chi}^2}{N_D(N_A - 1) - \tilde{\chi}^2}. \quad (7)$$

Note that F given in (7) follows a Fisher distribution with $N_A - 1 = 3$ and $(N_D - 1)(N_A - 1) = 3 \times 11 = 33$ degrees of freedom, that is, $F \sim F(3, 11)$. According to expressions given in (6) and (7), we have $\tilde{\chi}^2 = 18.4500$ and $F = 11.5641$. Hence, the corresponding p -value computed from the $F(3, 11)$ distribution is less than 0.001, so that the null hypothesis is rejected at 1% of significance. Therefore, the performances of the compared algorithms are significantly different, with the DBRS algorithm outperforming statistically the other three algorithms.

Table 4 reports the results in the colored scatter diagrams. As mentioned, the X-means algorithm and GMM always underestimate the number of clusters. Even though the logic of these two methods is different, both of them rely on the BIC. However, sometimes the BIC does not fit the clustering process well. In the X-means algorithm, the BIC is based on the assumption that both divided parts follow multivariate Gaussian distributions. Nevertheless, even when the number of clusters is moderate, at the beginning of the algorithm, the distributions of both parts are GMM. Therefore, it always stops early when the number of clusters is slightly large. Furthermore, we can verify it in the colored scatter diagrams of Table 4. This is the reason why the X-means algorithm is always the fastest method. The same problem occurs with the GMM. For the Dck-means algorithm, the radius of density canopy is fixed to the average pairwise distances of all sample points in the data set. Therefore, it loses some flexibility and faces some problems when the clusters are heterogeneous.

We compare the k-means++ and DBRS algorithms, whose results are provided in Fig. 5 for data set 1, as an example, whereas all data sets are shown in the figures displayed in the Appendix. The initial points chosen are as scattered as possible which, to some degree, can relieve the local convergence. However, it still cannot overcome the distance-based drawbacks previously illustrated. The DBRS algorithm uses the shape information in the data set and so the influence of assigning samples to clusters simply by distance is weakened to a great extent. In turn, the accuracy of the DBRS algorithm is considerable and often higher than the k-means++ algorithm, even when the right number of clusters (K) is given in the k-means++ algorithm. In addition, the running time of the k-means++ algorithm is approximately proportional to K . Consequently, when number of clusters given by user is large, the k-means++ algorithm is not as efficient as the DBRS algorithm.

4.4. Performance under noise

To show that the DBRS algorithm is non-sensitive to noise, we increase the noise of data sets 1 and 3, comprising homogeneous and

heterogeneous clusters, respectively. Then, we apply the four clustering methods mentioned above, which can automatically determine the number of clusters to the data sets with different proportions of noise; see Figs. 6 and 7. As the noise ratio increases, the DBRS algorithm maintains good performance. However, the introduction of noise causes fluctuations in the algorithms, so we also show the standard deviation (SD) of the ARI, FS, RI, and RT from 20 repeated experiments. The translucent bands in Figs. 6 and 7 represent the SD. The lower bounds of the bands equal the means minus SDs and the upper bounds of the bands equal the means plus SDs. From the widths of the bands, note that the proposed method is robust, especially on data set 3, which has more data points. That is because the location of Rpeak and Rvalley can be determined with a higher accuracy due to a finer PDF of distances. Nevertheless, the RT fluctuations are denser than the other three algorithms. That is because Pinitial is randomly selected and so if a debris point is selected as Pinitial, the time on establishing the PDF of distances concerning to this debris point is wasted.

We also demonstrate that the DBRS algorithm is non-sensitive to noise by the colored scatter diagrams displayed in Figs. 8 and 9. We present intermediate states in the scatter diagrams of the proposed method; see Figs. 8(a) and 9(a). The centers of the circles are cluster centers obtained by the fast mean-shift process, and the radius of the circles are the radius of clusters obtained by the extract process. The numbers in the circles show the order of extraction. Thus, while noise exists, their densities are relatively low compared to the density of the main body of clusters so that the boundary can be determined accurately using the PDF of distances. In data set 1, except for the GMM, all algorithms obtain good results. In data set 3, the DBRS algorithm is much more reasonable than the other methods. According to the above statement and the scatter diagrams, we conclude that the proposed method is non-sensitive to noise.

We note that, in Fig. 9(a), the noise is found as clusters by the DBRS algorithm. However, this is tolerable for the following reasons. First, when using ARI, FS, and RI to evaluate the effect of clustering algorithms, the noise is labeled as zero in the benchmark. Nevertheless, all four algorithms label all the points, including noise, as integers from one to the number of clusters. Moreover, the noise is only a small fraction in the whole data set and so the assignment of noise causes a slight drop in the evaluation criteria. This is the reason why the values of the evaluation criteria are consistently high in Figs. 6 and 7 even the noise is recognized as clusters.

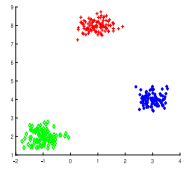
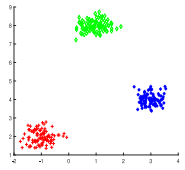
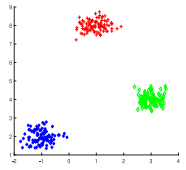
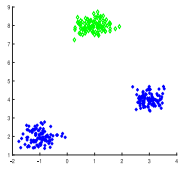
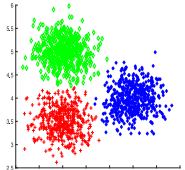
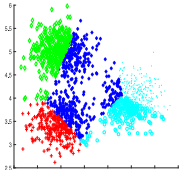
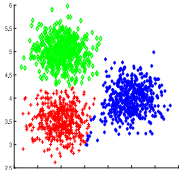
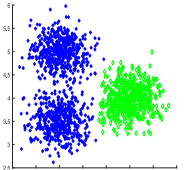
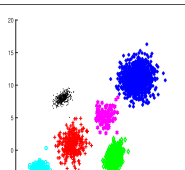
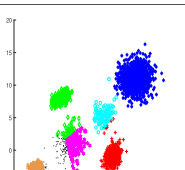
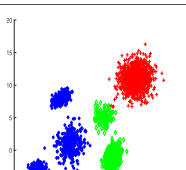
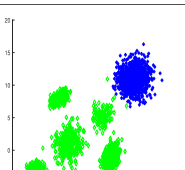
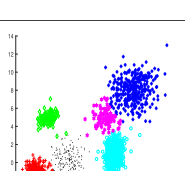
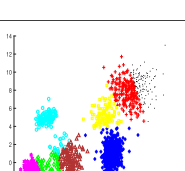
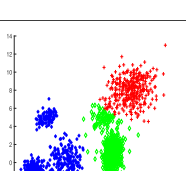
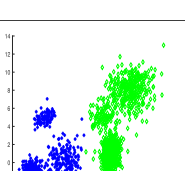
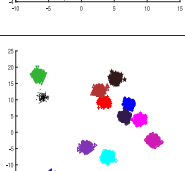
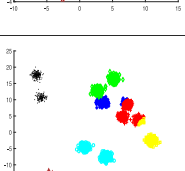
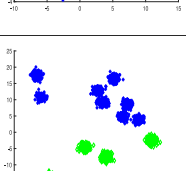
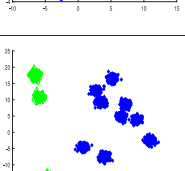
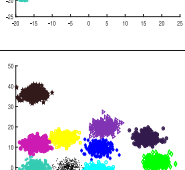
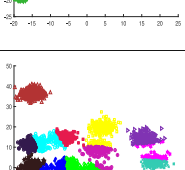
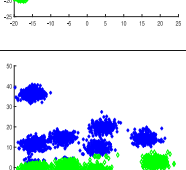
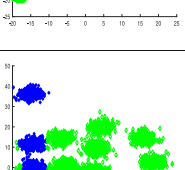
Table 3

Average values and 95% confidence intervals of the indicated criterion for the listed algorithm and synthetic data set.

Data set	RI [confidence interval]				ARI [confidence interval]			
	DBRS	DCK-means	X-means	GMM	DBRS	DCK-means	X-means	GMM
1	1 [1, 1]	0.9947 [0.9911, 0.9983]	1 [1, 1]	0.7770 [0.7770, 0.7770]	1 [1, 1]	0.9876 [0.9791, 0.9961]	1 [1,1]	0.5698 [0.5698, 0.5698]
2	0.9750 [0.9697, 0.9803]	0.8959 [0.8909, 0.9009]	0.9693 [0.9676, 0.9710]	0.7653 [0.7641, 0.7665]	0.9422 [0.9294, 0.9550]	0.7469 [0.7337, 0.7601]	0.9308 [0.9269, 0.9347]	0.5478 [0.5457, 0.5499]
3	0.9987 [0.9986, 0.9988]	0.9646 [0.9615, 0.9677]	0.8618 [0.8438, 0.8798]	0.7172 [0.7149, 0.7195]	0.9964 [0.9961, 0.9967]	0.9059 [0.8982, 0.9136]	0.6941 [0.6617, 0.7265]	0.4449 [0.4416, 0.4482]
4	0.9663 [0.9614, 0.9712]	0.8961 [0.8918, 0.9004]	0.8330 [0.8196, 0.8464]	0.6682 [0.6606, 0.6758]	0.904 [0.8905, 0.9175]	0.6886 [0.6734, 0.7038]	0.6201 [0.5991, 0.6411]	0.3690 [0.3600, 0.3780]
5	0.9976 [0.9969, 0.9984]	0.9165 [0.9108, 0.9222]	0.5861 [0.5662, 0.6060]	0.5075 [0.4931, 0.5219]	0.9817 [0.9757, 0.9877]	0.5781 [0.5590, 0.5972]	0.1524 [0.1406, 0.1642]	0.1106 [0.1040, 0.1172]
6	0.9689 [0.9568, 0.9810]	0.9329 [0.9302, 0.9356]	0.5545 [0.5515, 0.5575]	0.4917 [0.4707, 0.5127]	0.8432 [0.7995, 0.8869]	0.5595 [0.5459, 0.5731]	0.1221 [0.1206, 0.1236]	0.1008 [0.0943, 0.1073]
7	1 [1, 1]	0.9763 [0.9747, 0.9779]	0.9562 [0.9425, 0.9699]	0.7436 [0.7373, 0.7499]	1 [1, 1]	0.9344 [0.9298, 0.9390]	0.8997 [0.8684, 0.9310]	0.4913 [0.4830, 0.4995]
8	0.9549 [0.9351, 0.9747]	0.9327 [0.9299, 0.9355]	0.8767 [0.8488, 0.9046]	0.7448 [0.7421, 0.7475]	0.8944 [0.8530, 0.9358]	0.8115 [0.8031, 0.8199]	0.7469 [0.6947, 0.7991]	0.4954 [0.4913, 0.4995]
9	0.9500 [0.9440, 0.9560]	0.9437 [0.9369, 0.9505]	0.9253 [0.9101, 0.9405]	0.6742 [0.6603, 0.6881]	0.8682 [0.8544, 0.8820]	0.8251 [0.8016, 0.8486]	0.8204 [0.7872, 0.8536]	0.3859 [0.3698, 0.4020]
10	0.9356 [0.9227, 0.9485]	N/A N/A	0.9134 [0.9020, .9248]	0.6627 [0.6517, 0.6737]	0.8402 [0.8133, 0.8671]	N/A N/A	0.7741 [0.7482, 0.7997]	0.3543 [0.3430, 0.3656]
11	0.9781 [0.9648, 0.9914]	N/A N/A	1 [1, 1]	0.6120 [0.5901, 0.6339]	0.9465 [0.9156, 0.9774]	N/A N/A	1 [1, 1]	0.3089 [0.2850, 0.3328]
12	0.7249 [0.6647, 0.7851]	N/A N/A	0.8640 [0.8465, 0.8815]	0.7799 [0.7695, 0.7903]	0.5248 [0.4253, 0.6243]	N/A N/A	0.6776 [0.6448, 0.7104]	0.5110 [0.4887, 0.5333]
Data set	FS [confidence interval]				RT [confidence interval]			
	DBRS	DCK-means	X-means	GMM	DBRS	DCK-means	X-means	GMM
1	1 [1, 1]	0.9914 [0.9855, 0.9973]	1 [1, 1]	0.7481 [0.7481, 0.7481]	0.3844 [0.3594, 0.4090]	0.1492 [0.1334, 0.1650]	0.0227 [0.0148, 0.0306]	0.3156 [0.3031, 0.3281]
2	0.9604 [0.9512, 0.9696]	0.8165 [0.8061, 0.8269]	0.9539 [0.9513, 0.9565]	0.7363 [0.7351, 0.7375]	2.2539 [2.1651, 2.3427]	2.5219 [2.4897, 2.5541]	0.0695 [0.0649, 0.0741]	2.0117 [1.9668, 2.0566]
3	0.9973 [0.9971, 0.9975]	0.9294 [0.9238, 0.9350]	0.7847 [0.7636, 0.8058]	0.6251 [0.6232, 0.6270]	4.3469 [4.2008, 4.4930]	8.382 [8.2281, 8.5359]	0.1125 [0.1061, 0.1189]	1.1062 [1.0254, 1.1870]
4	0.9258 [0.9155, 0.9361]	0.7542 [0.7414, 0.7670]	0.7258 [0.7123, 0.7393]	0.5650 [0.5596, 0.5704]	2.4891 [2.3836, 2.5946]	1.7945 [1.7703, 1.8187]	0.0547 [0.0516, 0.0578]	1.2602 [1.2008, 1.319]
5	0.9830 [0.9774, 0.9886]	0.6172 [0.6003, 0.6341]	0.2530 [0.2431, 0.2629]	0.2182 [0.2128, 0.2236]	9.9812 [9.6851, 10.277]	26.3242 [24.919, 27.728]	0.1289 [0.1206, 0.1372]	1.2742 [1.2022, 1.3462]
6	0.8572 [0.8180, 0.8964]	0.5946 [0.5823, 0.6069]	0.2279 [0.2266, 0.2292]	0.2102 [0.2048, 0.2156]	16.193 [15.692, 16.694]	131.5789 [126.69, 136.46]	0.3148 [0.2877, 0.3419]	4.3945 [4.2030, 4.5860]
7	1 [1, 1]	0.9498 [0.9463, 0.9533]	0.9298 [0.9078, 0.9518]	0.6615 [0.6567, 0.6663]	2.7563 [2.6934, 2.8192]	6.9938 [6.9605, 7.0271]	0.0891 [0.0839, 0.0943]	1.5219 [1.4424, 1.6014]
8	0.9252 [0.8973, 0.9531]	0.8541 [0.8473, 0.8609]	0.8313 [0.7980, 0.8646]	0.6709 [0.6685, 0.6733]	2.0648 [1.8627, 2.2669]	1.8141 [1.7847, 1.8435]	0.0578 [0.0522, 0.0634]	1.5008 [1.4049, 1.5967]
9	0.9008 [0.8909, 0.9107]	0.8594 [0.8397, 0.8791]	0.8688 [0.8453, 0.8923]	0.5802 [0.5707, 0.5897]	21.568 [20.907, 22.229]	1202.2 [1151.6, 1252.7]	1.6766 [1.6020, 1.7512]	11.677 [11.161, 12.191]
10	0.8819 [0.8631, 0.9007]	N/A N/A	0.8283 [0.8095, 0.8471]	0.5421 [0.5351, 0.5491]	20.273 [19.517, 21.029]	N/A N/A	2.8180 [2.6688, 2.9672]	19.527 [18.652, 20.401]
11	0.9599 [0.9370, 0.9828]	N/A N/A	1 [1,1]	0.5154 [0.5005, 0.5303]	37.620 [33.972, 41.268]	N/A N/A	2.0812 [1.8627, 2.2997]	32.280 [30.692, 33.867]
12	0.7896 [0.7490, 0.8302]	N/A N/A	0.7604 [0.7376, 0.7832]	0.6541 [0.6380, 0.6702]	9.5422 [9.1645, 9.9199]	N/A N/A	3.2852 [3.1271, 3.4433]	63.043 [60.542, 65.545]

The symbol N/A in the table is used when we do not get the result.

Table 4
Graphical results of the indicated algorithm and data set.

Data set	DBRS	DCK-means	X-means	GMM
1				
2				
3				
4				
5				
6				

4.5. Discussion on very different densities in the data

In Sections 4.3 and 4.4, we presented the effectiveness of the DBRS algorithm in different types of data sets. In this subsection, we generate a series of data sets similar to dat_6 considered in Cheng et al. (2019) to discuss the performance of the DBRS algorithm on data sets composed of clusters with very different densities. There are four clusters in the data sets: $cluster_1$ with center at (0,0); $cluster_2$ with center at (0,4); $cluster_3$ with center at (5,0); and $cluster_4$ with center at (5,4); all of them based on covariance matrices equal to $0.5E_2$.

The sizes of the clusters are significantly different, which leads to greater differences in densities. We set the size of $cluster_1$ as a baseline at $N_1 = 100$. The size of other clusters in different data sets (13-16) are as follows.

Data set 13. The sizes of $cluster_1$, $cluster_2$, $cluster_3$, and $cluster_4$ are stated as $N_1 = N_1$, $N_2 = 5N_1$, $N_3 = 10N_1$, and $N_4 = 20N_1$, respectively.

Data set 14. The sizes of $cluster_1$, $cluster_2$, $cluster_3$, and $cluster_4$ are stated as $N_1 = N_1$, $N_2 = 10N_1$, $N_3 = 20N_1$, and $N_4 = 40N_1$, respectively.

Data set 15. The sizes of $cluster_1$, $cluster_2$, $cluster_3$, and $cluster_4$ are stated as $N_1 = N_1$, $N_2 = 20N_1$, $N_3 = 40N_1$, and $N_4 = 80N_1$, respectively.

Data set 16. The sizes of $cluster_1$, $cluster_2$, $cluster_3$, and $cluster_4$ are stated as $N_1 = N_1$, $N_2 = 30N_1$, $N_3 = 60N_1$, and $N_4 = 120N_1$, respectively.

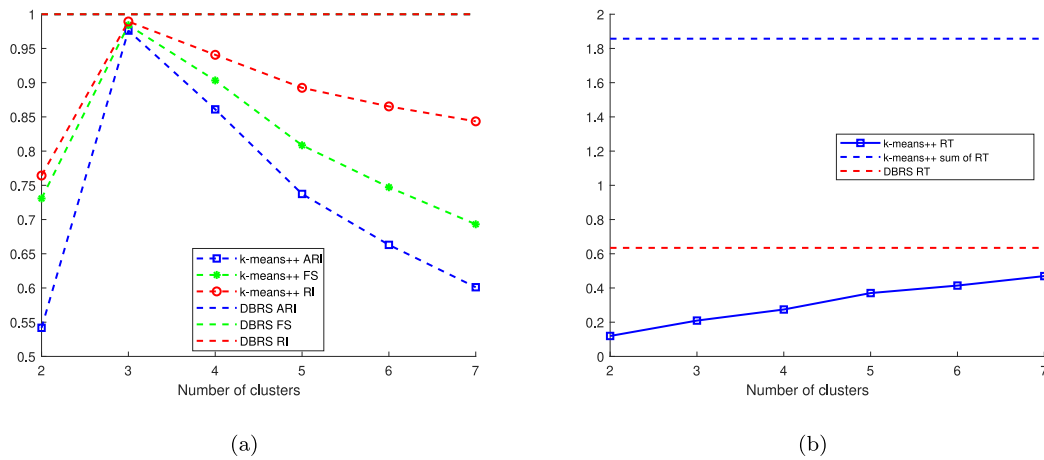


Fig. 5. Values of the (a) indicated criterion and (b) running times for the listed algorithm with data set 1.

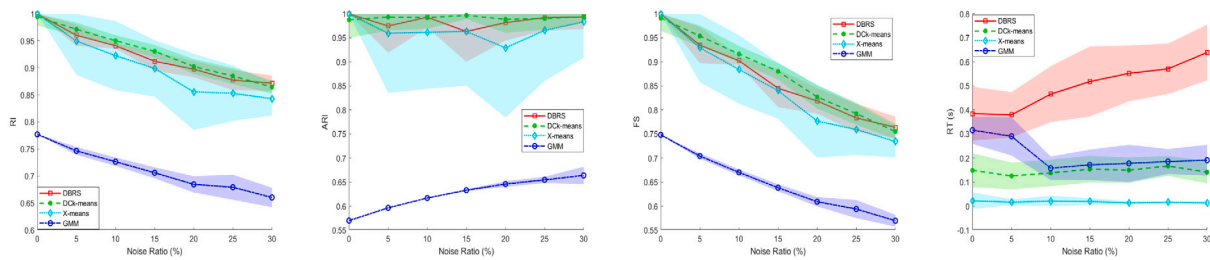


Fig. 6. Noise test on data set 1 (homogeneous clusters) for the indicated criterion and algorithm.

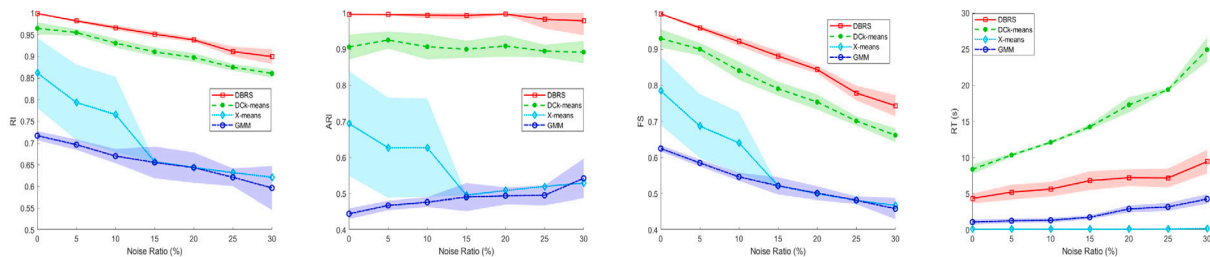


Fig. 7. Noise test on data set 3 (heterogeneous clusters) for the indicated criterion and algorithm.

When establishing the PDF of distances, we smooth the corresponding histogram. In Algorithms 1 and 2, LenInt is used, which is set to be a proportion to the range of D in Algorithm 3. When the densities of clusters are not very different, LenInt has a wide effective range, which ensures robust and reasonable clustering results (so we do not set LenInt as an input parameter in the above parts). Here, “reasonable” means “consistent with human perception” rather than a high ARI, FS, and RI. This is because whether or not the sparser clusters are grouped correctly and do not significantly impact the criteria on a data set with very different densities. After testing with several

values, we get the effective range of the proportion in different cases as follows. Data set 13: [1/50, 1/360]; data set 14: [1/60, 1/250]; data set 15: [1/120, 1/230]. We are unable to find a robust and effective proportion on data set 16. The range of effective proportion shrinks with the increasing differences in densities. This is because we establish the PDF of distances by a smoothed histogram. When the densities of the clusters are very different, the radius of denser clusters are always greater than their exact value because of the smoothing procedures. Consequently, fractions of the sparser cluster are absorbed by denser clusters and the sparser cluster is chopped; see Fig. 10.

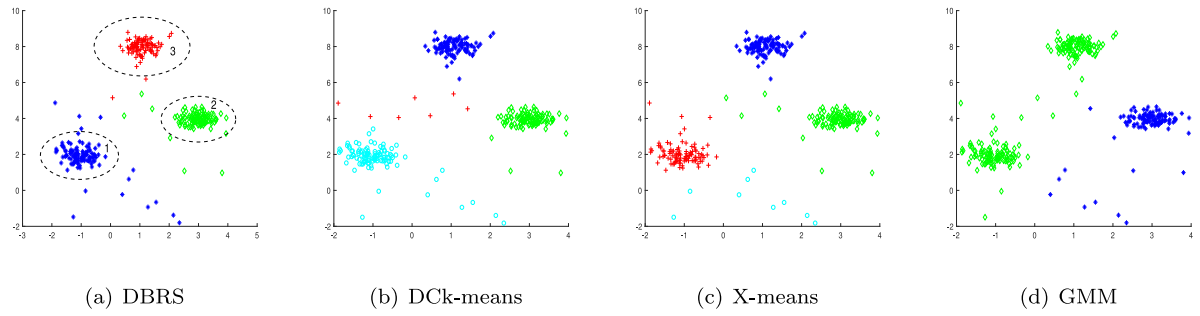


Fig. 8. Results on data set 1 (homogeneous clusters) with 10% noise ratio for the indicated algorithm.

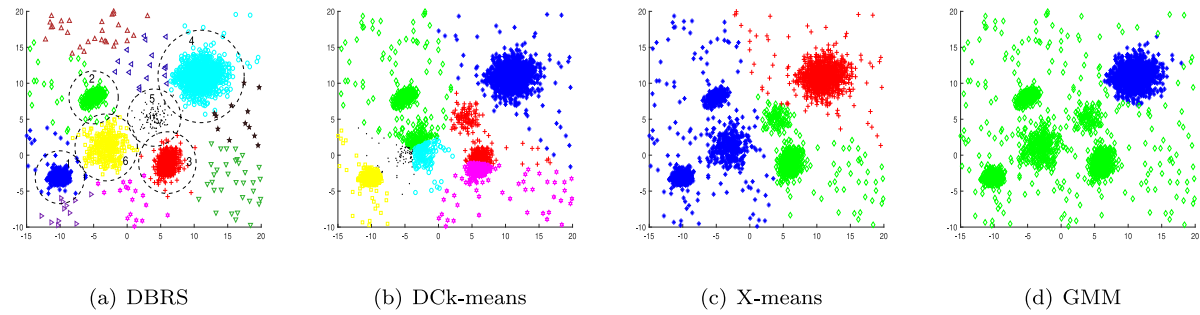


Fig. 9. Results on data set 3 (heterogeneous clusters) with 10% noise ratio for the indicated algorithm.

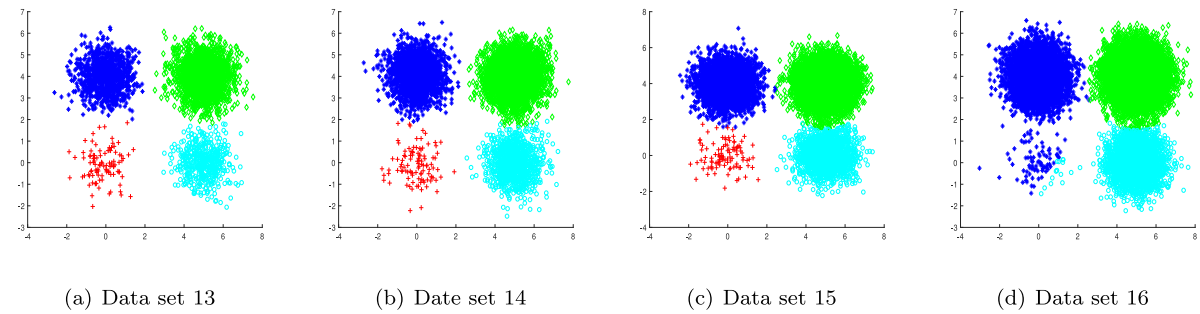


Fig. 10. Clustering results of different densities (all $\text{LenInt}=1/150 \times [\text{range of } D]$) for the indicated data set.

In summary, our proposed method can be used to deal with density differences of about 100 times in the data set. When the density difference is more prominent, we should seek other solutions, such as the granularity method.

5. Illustration with real-world data

In this section, we use four real data sets from the UCI machine learning repository (<https://archive.ics.uci.edu>, accessed on 01 November 2021) of the University of California Irving (UCI) to illustrate the DBRS algorithm and its potential applications. The data sets are different from each other in terms of the number of sample points,

Table 5

Features of the indicated UCI data set.

Data set	Sample points	Attributes	Clusters
Wine	178	13	3
Seed	210	7	3
Abalone	4177	7	29
Iris	150	4	3

attributes and clusters. Some features of these data sets are shown in Table 5.

Table 6
Values of the indicated criterion and algorithm using the listed UCI data set.

Data set	RI				ARI			
	DBRS	DCK-means	X-means	GMM	DBRS	DCK-means	X-means	GMM
Wine	0.7281	0.7166	0.6552	0.7173	0.2746	0.3151	0.1880	0.4906
Seed	0.8437	0.8259	0.8253	0.3301	0.5507	0.5811	0.6106	0.0000
Abalone	0.8126	0.8253	0.6768	0.1043	0.0558	0.0472	0.0424	0.0000
Iris	0.7829	0.8859	0.8497	0.8312	0.5575	0.7455	0.6622	0.6235
Data set	FS				RT			
	DBRS	DCK-means	X-means	GMM	DBRS	DCK-means	X-means	GMM
Wine	0.4737	0.3950	0.4275	0.6739	1.6117	0.1023	0.0313	0.125
Seed	0.7008	0.7007	0.7000	0.7746	0.3383	0.0711	0.0328	0.118
Abalone	0.1600	0.1400	0.1895	0.1154	22.8422	139.9086	0.8805	2.2008
Iris	0.7308	0.8317	0.7750	0.7610	0.1781	0.0375	0.0234	0.0914

We present the clustering results of the DBRS, DCK-means, X-means algorithms and the GMM in Table 6. Comparing the RI, the DBRS algorithm performs better in the wine and seed data sets, whereas the DCK-means algorithm performs better in the other two data sets. However, the performance of the X-means algorithm and GMM is not satisfactory due to the early stop. For the FS, the DBRS algorithm performs well in the seed data set. For the ARI, the DBRS algorithm performs well in the abalone data set. The X-means algorithm and GMM perform better in UCI data sets than in the synthetic data sets because the number of clusters in the wine, seed, and iris data sets is equal to three, and so the influence of early stop is weaker.

6. Concluding remarks and future research

In this paper, we proposed a novel density-based radar scanning clustering algorithm, which can discover and extract individual clusters efficiently and is robust to noise. We used the radar scanning strategy to project all the points to a one-dimensional space, ensuring satisfactory results on high-dimension data sets. We may get crucial information from the probability density functions of distances, which were employed to deduce suitable radiuses and active subsets to accelerate the original mean-shift process. The radiuses of potential clusters are also driven out from the probability density functions of distances, which are more flexible to handle heterogeneous clusters. In general, our findings are reported as follows:

- (i) From the experimental results, the synthetic data sets all indicated that the density-based radar scanning clustering algorithm is effective, accurate and non-sensitive to noise.
- (ii) Compared to other methods, the performance of the new proposed algorithm in the real data sets was not as high as its performance in the synthetic data sets, because the attributes in the UCI data sets are strongly correlated with each other.
- (iii) Similarly to the case of synthetic data, when using real data, the results showed that the density-based radar scanning clustering algorithm is effective, accurate and non-sensitive to noise.
- (iv) The new clustering algorithm can only handle data sets with slight correlation, namely approximately spherical; otherwise we need the clusters to be well-separated.

The aspects indicated below are open problems for the new density-based radar scanning clustering algorithm and they will be considered in future research:

- (i) Improvements should be made to deal with strongly correlated data sets.
- (ii) Since we only apply the density-based radar scanning clustering algorithm to spherical data sets, we intend to extend this method to non-spherical data sets.

Credit authorship contribution statement

Lin Ma: Data curation, investigation, formal analysis, methodology, writing – original draft, writing – review & editing. **Yi Zhang:** Data curation, guidance on applications, data generation and writing. **Victor Leiva:** Investigation, formal analysis, methodology, writing – review & editing. **Shuangzhe Liu:** Investigation, formal analysis, methodology, writing – review & editing. **Tiefeng Ma:** Investigation, formal analysis, methodology, writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to thank the editors and reviewers for their constructive comments on an earlier version of this manuscript which led to an improved presentation. This work was supported partially by the National Natural Science Foundation of China, grant number 51777035 (Y. Zhang); and by FONDECYT, grant number 1200525, from the National Agency for Research and Development (ANID) of the Chilean government under the Ministry of Science, Technology, Knowledge and Innovation (V.Leiva).

Appendix

Comparison of the k-means++ and DBRS algorithms for all data sets; see Figs. 11 and 12.

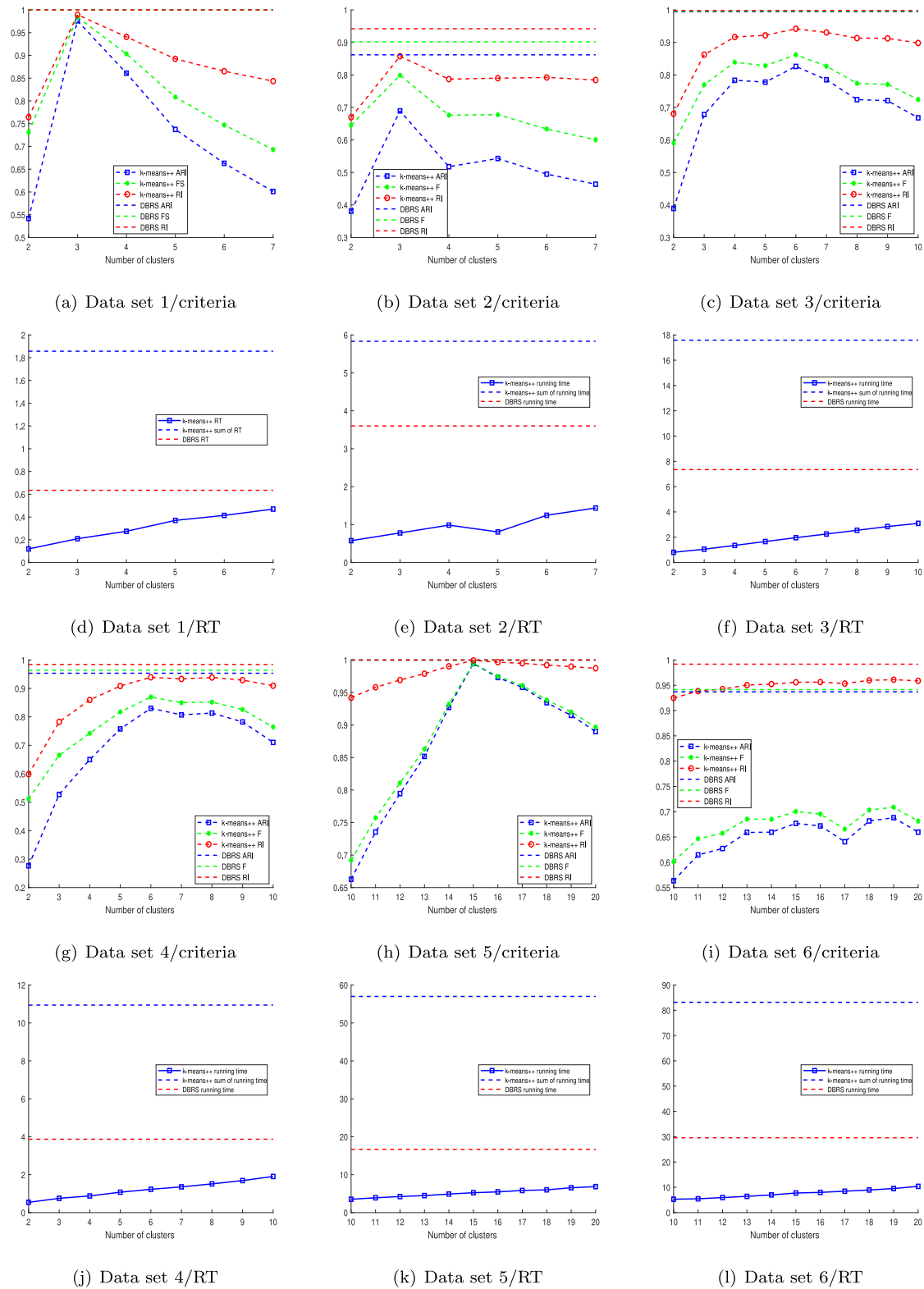


Fig. 11. Values of the indicated criterion (a-c,g-i) and running times (RT) (d-f,j-l) for the listed algorithm and data set.

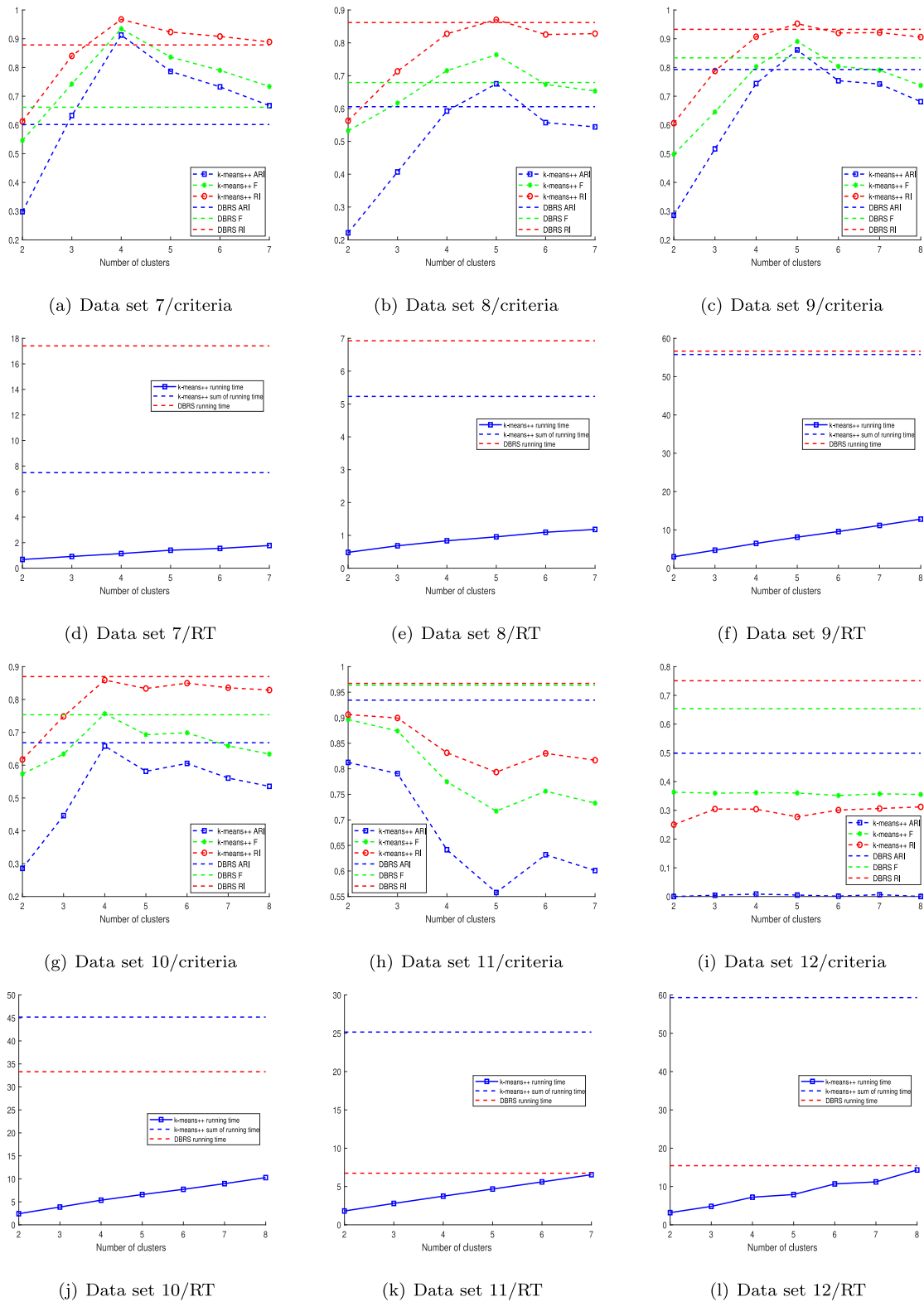


Fig. 12. Values of the indicated criterion (a-c,g-i) and running times (RT) (d-f,j-l) for the listed algorithm and data set.

References

- Akogul, S., & Eriş, M. (2017). An approach for determining the number of clusters in a model-based cluster analysis. *Entropy*, 19, 452–466.
- Andrews, J. (2018). Addressing overfitting and underfitting in gaussian model-based clustering. *Computational Statistics and Data Analysis*, 127, 160–171.
- Arthur, D., & Vassilvitskii, S. (2017). k-means++: the advantages of careful seeding. In *Proceedings of the nineteenth annual ACM-SIAM symposium on discrete algorithms*. Vol. 8 (pp. 1027–1035).
- Aykroyd, R. G., Leiva, V., & Ruggeri, F. (2019). Recent developments of control charts, identification of big data sources and future trends of current research. *Technological Forecasting and Social Change*, 144, 221–232.
- Cabezas, X., Garcia, S., Martin-Barreiro, C., Delgado, E., & Leiva, V. (2021). A two-stage location problem with order solved using a Lagrangian algorithm and stochastic programming for a potential use in COVID-19 vaccination based on sensor-related data. *Sensors*, 21, 5352.
- Chen, Y., Tang, S., Pei, S., Wang, C., Du, J., & Xiong, N. (2018). Dheat: A density heat-based algorithm for clustering with effective radius. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 48, 649–660.
- Chen, Y., Zhou, L., Bouguila, N., Wang, C., Chen, Y., & Du, J. (2020). Block-dbscan: Fast clustering for large scale data. *Pattern Recognition*, 109, Article 107624.
- Cheng, Y. (1995). Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17, 790–799.
- Cheng, M., Ma, T., & Liu, Y. (2019). A projection-based split-and-merge clustering algorithm. *Expert Systems with Applications*, 116, 121–130.
- Dan, P., & Moore, A. (2000). Extending k-means with efficient estimation of the number of clusters. In *proceedings of the seventeenth international conference on machine learning*. (pp. 727–734).
- Demiar, J., & Schuurmans, D. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7, 1–30.
- Díaz-García, J., & Leiva, V. (2003). Doubly non-central t and F distributions obtained under singular and non-singular elliptic distributions. *Communication in Statistics: Theory and Methods*, 32, 11–32.
- El-Shafey, E., Sallam, K. M., Chakraborty, R. K., & Abohany, A. A. (2021). A clustering based swarm intelligence optimization technique for the internet of medical things. *Expert Systems with Applications*, 173, Article 114648.
- Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of second international conference on knowledge discovery and data mining*. Vol. 96 (pp. 226–231).
- Fräley, C., & Raftery, A. E. (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97, 611–631.
- Frossyniotis, D., Pertselakis, M., & Stafylopatis, A. (2002). A multi-clustering fusion algorithm. In *Proceedings of the second hellenic conference on AI: methods and applications of artificial intelligence*. No. 2308 (pp. 225–236). Berlin: Springer.
- Fukunaga, K., & Hostetler, L. (1975). The estimation of the gradient of a density function. *IEEE Transactions on Information Theory*, 21, 32–40.
- Gholizadeh, N., Saadatfar, H., & Hanafi, N. (2020). K-dbscan: An improved dbscan algorithm for big data. *The Journal of Supercomputing*, 77, 6214–6235.
- Guan, J., Li, S., He, X., Zhu, J., & Chen, J. (2021). Fast hierarchical clustering of local density peaks via an association degree transfer method. *Neurocomputing*, 455, 401–408.
- Huang, X., Ye, Y., Guo, H., Cai, Y., Zhang, H., & Li, Y. (2014). Dskmeans: A new kmeans-type approach to discriminative subspace clustering. *Knowledge-Based Systems*, 70, 293–300.
- Jain, A., Murty, M., & Flynn, P. (1999). Data clustering: A review. *ACM Computing Surveys*, 31, 264–323.
- Kazemi, U., & Boostani, R. (2021). FEM-DBSCAN: AN efficient density-based clustering approach. *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, 45, 979–992.
- Kile, H., & Uhlen, K. (2012). Data reduction via clustering and averaging for contingency and reliability analysis. *International Journal of Electrical Power & Energy Systems*, 43, 1435–1442.
- Laohakiat, S., & Saing, V. (2021). An incremental density-based clustering framework using fuzzy local clustering. *Information Sciences*, 547, 404–426.
- Li, H., Liu, X., Li, T., & Gan, R. (2020). A novel density-based clustering algorithm using nearest neighbor graph. *Pattern Recognition*, 102, 1–12.
- Liu, C., He, X., & Xu, Q. (2017). A fuzzy density peak optimization initial centers selection for k-medoids clustering algorithm. In *International conference on computer engineering and networks*. Vol. 1 (pp. 49–55).
- Martin-Barreiro, C., Ramirez-Figueroa, J. A., Cabezas, X., Leiva, V., & Galindo-Villardón, M. P. (2021). Disjoint and functional principal component analysis for infected cases and deaths due to covid-19 in South American countries with sensor-related data. *Sensors*, 21(4094).
- Martin-Barreiro, C., Ramirez-Figueroa, J. A., Nieto-Librero, A. B., Leiva, V., Martin-Casado, A., & Galindo-Villardón, M. P. (2021). A new algorithm for computing disjoint orthogonal components in the three-way tucker model. *Mathematics*, 9(203).
- Min, X., Huang, Y., & Sheng, Y. (2020). Automatic determination of clustering centers for clustering by fast search and find of density peaks. *Mathematical Problems in Engineering*, 2020, 1–11.
- Parmar, M., Wang, D., Zhang, X., Tan, A., Miao, C., Jiang, J., et al. (2019). Redpc: A residual error-based density peak clustering algorithm. *Neurocomputing*, 348, 82–96.
- Ramirez-Figueroa, J. A., Martin-Barreiro, C., Nieto-Librero, A. B., Leiva, V., & Galindo-Villardón, M. P. (2021). A new principal component analysis by particle swarm optimization with an environmental application for data science. *Stochastic Environmental Research and Risk Assessment*, 35, 1969–1984.
- Thrun, M. C., & Ultsch, A. (2020). Using projection-based clustering to find distance- and density-based clusters in high-dimensional data. *Journal of Classification*, 38, 54–65.
- Thrun, M. C., & Ultsch, A. (2021). Swarm intelligence for self-organized clustering. *Artificial Intelligence*, 290, Article 103237.
- Wang, Z. (2019). A new clustering method based on morphological operations. *Expert Systems with Applications*, 145, Article 113102.
- Wang, Y., & Yang, Y. (2021). Relative density-based clustering algorithm for identifying diverse density clusters effectively. *Neural Computing and Applications*, 33, 10141–10157.
- Wang, P., & Yao, Y. (2018). Ce3: A three-way clustering method based on mathematical morphology. *Knowledge-Based Systems*, 155, 54–65.
- Yang, Y., Cai, J., Yang, H., Zhang, J., & Zhao, X. (2019). TAD: A trajectory clustering algorithm based on spatial-temporal density analysis. *Expert Systems with Applications*, 139, 1–16.
- Zhang, G., Zhang, C., & Zhang, H. (2018). Improved k-means algorithm based on density canopy. *Knowledge-Based Systems*, 145, 289–297.
- Zhong, C., Miao, D., & Franti, P. (2011). Minimum spanning tree based split-and-merge: A hierarchical clustering method. *Information Sciences*, 181, 3397–3410.