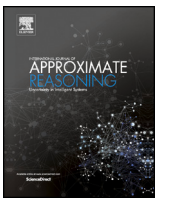
International Journal of Approximate Reasoning 164 (2024) 109075

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

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



GFDC: A granule fusion density-based clustering with evidential reasoning

Mingjie Cai [a,b], Zhishan Wu [a], Qingguo Li [a], Feng Xu [a,*], Jie Zhou [c]

[a] School of Mathematics, Hunan University, Changsha, Hunan, 410082, PR China
[b] Shenzhen Research Institute of Hunan University, Shenzhen, Guangdong, 518000, PR China
[c] School of Computer Science and Software Engineering, Shenzhen University, Shenzhen, Guangdong, 518060, PR China

ARTICLE INFO

Keywords:
Granular computing
Granule fusion
Density-based clustering
Dempster-Shafer theory

ABSTRACT

Density-based clustering algorithms are known for their ability to detect irregular clusters, but they have limitations when it comes to dealing with clusters of varying densities. In this paper, we propose a new clustering algorithm called granule fusion density-based clustering with evidential reasoning (GFDC). The approach introduces the concept of sparse degree, which measures both the local density and global density of samples. The sparse degree of samples reflects the stability of samples. Moreover, a core-granule is composed of the neighborhood granule of a sample, of which the sparse degree is minimum in its neighborhood. Then, the core-granules are generated based on the sparse degree and are insensitive to clusters with varying densities. The core samples, which consist of samples in core-granules, are used to form initial clusters through fusion strategies. Additionally, an assignment method is developed from Dempster-Shafer theory to assign border samples and identify outliers. The experimental results demonstrate the effectiveness of GFDC on extensive synthetic and real-world datasets.

1. Introduction

Clustering is a significant technique for categorizing data samples into clusters, where samples within the same cluster exhibit a high degree of similarity. This unsupervised machine learning method enables the extraction of inherent structural information from data without any prior knowledge. At present, clustering algorithms are rapidly evolving with a more comprehensive knowledge framework and wider application scenarios. In general, existing clustering algorithms can be classified into the following categories: partitioning clustering (e.g., k-means [1], k-means++ [2]), hierarchical clustering (e.g., BIRCH [3]), graph-based clustering (e.g., SC [4]), density-based clustering (e.g., DBSCAN [5], DPC [6]), grid-based clustering (e.g., STING [7]), model-based clustering (e.g., EM [8]), deep clustering (e.g., DEN [9], CCNN [10]), etc. Density-based clustering algorithms have gained significant attention due to their capability to detect clusters with arbitrary shapes.

A novel and effective density-based clustering algorithm (DPC) is proposed by Rodriguez and Laio in 2014 [6], which has been widely used and studied. In DPC, cluster centers have a higher density than their neighbors, and are relatively farther away from other denser samples. The algorithm utilizes a density estimation function to measure the density of samples, and then identifies cluster centers and outliers using a decision graph. Once the cluster centers are determined, the remaining samples are assigned to

* Corresponding author.
E-mail addresses: cmjlong@163.com (M. Cai), zhishanwu@hnu.edu.cn (Z. Wu), liqingguoli@aliyun.com (Q. Li), fengxuphd@163.com (F. Xu), jie_jpu@163.com (J. Zhou).

https://doi.org/10.1016/j.ijar.2023.109075
Received 23 August 2023; Received in revised form 20 October 2023; Accepted 26 October 2023

Available online 31 October 2023

the cluster where the nearest sample with a higher density is located. Although DPC is natural and efficient, there are some problems to be improved. Firstly, the parameter cutoff distance in the density estimation function is difficult to determine, and an improper parameter can affect the generation of a decision graph and the selection of cluster centers. In order to address the difficulty of determining cutoff distance, some scholars improve the density measure with the idea of k-nearest neighbors [11–19]. Furthermore, other scholars question the validity of density metrics, so they introduce the concept of belief in the Dempster-Shafer theory [20] to improve the density measure formula [21–23]. From the experimental results of the above algorithms, it turns out that the difficulty in determining the parameter is indeed reduced by replacing the cutoff distance with the k-nearest neighbors. Secondly, cluster centers cannot be accurately determined by a decision graph. What's more, the step of selecting cluster centers makes the algorithm impossible to achieve full process automation, so the authors of DPC propose a hint that cluster centers can be chosen from the samples with the highest product of density and distance. ADPC-KNN [13] regards the samples with a distance greater than the cutoff distance as initial cluster centers. A scoring formula with density and distance is exploited to identify cluster centers in DPC-DBFN [14], and the samples with the highest score are considered as cluster centers. These algorithms [13–16] can determine cluster centers simply and achieve full process automation. Thirdly, DPC produces unreasonable results when cluster centers are chosen improperly, and the errors are propagated rapidly. As a result, many assignment strategies are proposed [12–16,18,19,21–25]. Among them, some methods [14,19,22] are more beneficial to avoid budding errors and propagating errors rapidly based on initial clusters or other stable structures.

Information granules [26] was proposed by Zadeh, in which the concept of fuzzy granules and information granularity was first introduced and discussed. Information granules are the basic elements of granular computing [27]. The process of generating information granules is called information granulation. Information granulation is a process that data elements are grouped to information granules based on a specific relationship. This means that data elements with similar characteristics can be grouped together, simplifying the data without altering its structure. This property has made information granulation and information granules increasingly popular among scholars. Nowadays, an increasing number of studies demonstrate that excellent results can be achieved through the inclusion of information granules into machine learning. For example, GSVM (granular support vector machine) decomposes a linearly non-separable problem into multiple linearly separable problems by generating information granules and building an SVM in each granule [28]. Ding proposes a new fuzzy support vector machine based on information granulation, which granulates a dataset with FCM (fuzzy c-means) to get multiple granules and builds a classifier only on the mixed granules because mixed granules contain more useful boundary information [29]. Recently, Xia introduces the concept of a granular ball, which is a granule granulated by k-means and represented by a hypersphere structure, and presents a granular classifier framework base on granular balls [30]. In addition, granular balls are utilized to reduce the computation time of distances among centers and accelerate k-means [31]. Obviously, information granulation not only reduces the difficulty of problem processing, but also discovers the most informative samples in data. Moreover, different samples may be suitable for different processing methods, so it is reasonable to aggregate samples with the same character by information granulation and take into account processing strategies separately in terms of information granules. Therefore, granulation is beneficial to improving the efficiency and quality of data mining as well as making classification, clustering and other machine learning methods work better.

Evidential reasoning originated from a framework for analyzing arguments [32], which was a quantitative method used to make decisions based on acquirable evidence [33]. Dempster-Shafer theory, proposed in the 1960s [20], can be considered as an approach of evidential reasoning. Dempster-Shafer theory represents the concepts of uncertainty and unknown by introducing belief function and plausibility function, and combines pieces of evidence into conclusions by implementing a rule of combination. Due to its exceptional ability to handle uncertain information representation and fusion, Dempster-Shafer theory has become an important technological approach and effective expression tool for machine learning nowadays, which has been extensively utilized in various domains such as information fusion [34], classification [35], clustering, ensemble learning [36], image segmentation [37], medical diagnosis [38], etc. In 2004, Denoeux and Masson proposed EVCLUS [39], the first method combining Dempster-Shafer theory and clustering. What's more, they first introduced the concept of credal partition, by which the uncertainty of data can be well judged and represented. Remarkably, the credal partition is a general model of partition and it can generalize crisp partition, fuzzy partition, rough partition, etc. After that, many effective evidential clustering algorithms are researched and proposed [21–23,40–43], not least of which is the improvement of density-based algorithms using Dempster-Shafer theory.

Density-based clustering algorithms utilize a density measure to identify representative elements of a cluster, which consist of one or multiple samples with high density. Hence, the density measure plays a crucial role. And the label propagation strategy is affected by complex data structures. While there are numerous advancements, density-based clustering algorithms encounter challenges when handling complex data structures, clusters with varying densities, and the setting of optimal parameters. Taking into account the benefits of granular computing and evidential reasoning in data representation and information fusion, a granule fusion density-based clustering with evidential reasoning (abbreviated as GFDC) is proposed in this paper, which can process data with arbitrary cluster shapes and large density differences among clusters. First of all, the sparse degree of samples is introduced, which measures both the local and global density information of samples. Subsequently, core-granules are defined based on the sparse degree. The center of each core-granule has the minimum sparse degree and samples in each core-granule are taken as core samples. Then, all core-granules are fused to stable initial clusters. Finally, an improved evidential assignment method is applied to label assignment of border samples and identification of outliers. In detail, the main contributions of GFDC are as follows:

1) By integrating the notion of optimal information granularity and k-nearest neighbors, this paper proposes a novel sparse degree metric to measure both local and global densities of samples.

2) The proposed algorithm granulates samples based on their sparse degrees so that to generated core-granules, not only in high-density regions but also in low-density regions, which allows the proposed algorithm to cope with data where there are significant differences in density among clusters.
3) Based on intersection relationship, density transmission and distance, three fusion strategies are designed to fuse core-granules into initial clusters, which enables the proposed algorithm to detect clusters with arbitrary shapes.
4) An improved evidential assignment method is proposed for assigning the remaining border samples based on initial clusters, which makes the assignment results more reasonable and also can identify outliers.

The rest of this paper is organized as follows: In Section 2, the basic concepts of granular computing, the Dempster-Shafer theory and credal partition are briefly recalled. The details of the proposed algorithm GFDC are introduced in Section 3. Section 4 presents the experimental results of GFDC on extensive synthetic and real-world datasets. The conclusions of the paper are given in Section 5.

2. Preliminaries

The purpose of this section is to briefly review some necessary background information for readers, including the connotation of information granules [27,44] (Section 2.1), some basis of the Dempster-Shafer theory [20,45–47] (Section 2.2) and the concept of credal partition [39] (Section 2.3).

2.1. Information granules

Information granules are collections of elements (samples) gathered together according to indistinguishability, similarity, proximity or functionality [27]. The process of constructing information granules is called *information granulation*, which involves the abstract representation of samples and the extraction and summarization of information for data. In particular, binary relations are the frequently used information granulation strategies, and they include equivalence relations, neighborhood relations, tolerance relations and dominance relations.

Definition 1. [44] Given a universe U and a family of binary relations $\boldsymbol{R}$, the **granular structure** induced by a binary relation $P \in \boldsymbol{R}$ can be represented as a vector $K(P) = \left(N_P(x_1), N_P(x_2), \ldots, N_P(x_n)\right)$, where $x_i \in U$ and $N_P(x_i)$ is the **information granule** generated by a sample x_i concerning P.

An information granule $N_P(x_i)$ is a set of samples containing at least x_i in U. A collection of information granules is a granular structure. For example, in Example 3.2 in the later section, as shown in Fig. 4(a), a generated granular structure, when $k = 5$, is $(\{x_1, x_2, x_3, x_4, x_5\}, \{x_6\}, \{x_7, x_9, x_{10}, x_{11}, x_{12}\}, \{x_8\}, \{x_{13}\})$. In other words, $\{x_1, x_2, x_3, x_4, x_5\}$, $\{x_6\}$, $\{x_7, x_9, x_{10}, x_{11}, x_{12}\}$, $\{x_8\}$ and $\{x_{13}\}$ are information granules in this case. It is worth noting that a granular structure always satisfies $\bigcup_{x_i \in U} N_P(x_i) = U$, in other words, a granular structure can form a cover of the universe. In particular, a granular structure induced by a equivalence relation satisfies $N_P(x_i) \cap N_P(x_j) = \emptyset$, where $i \neq j$, which means that it can even form a partition of the universe.

Information granularity of an information granule measures the uncertainty of the granular structure and represents the discernibility ability of information in the granule. A smaller information granularity is associated with a stronger discernibility ability. However, it is not necessarily better to have smaller information granularity. Instead, it is important to determine the optimal information granularity based on the specific situation. Since information granulation is a process of abstract expression and extraction of information with clear objectives, corresponding to the degree and perspective of abstract expression as well as the depth and breadth of extracted information, the information granularity must be determined by the practical requirements when performing information granulation.

2.2. Dempster-Shafer theory

Dempster-Shafer theory, also known as the evidence theory or theory of belief functions [20,45] is an important method of uncertain reasoning.

Given a finite and unordered set $\Omega = \{\omega_1, \omega_2, \ldots, \omega_c\}$, which represents the possible values of a variable x, called the *frame of discernment*. Given a mass function is defined as a mapping from 2^Ω to $[0, 1]$ such that

$$\sum_{A \subseteq \Omega} m^\Omega(A) = 1, \tag{2.1}$$

where 2^Ω is the power set of Ω, with $2^\Omega = \{\emptyset, \{\omega_1\}, \{\}\omega_2\}, \ldots, \{\omega_1, \omega_2\}, \ldots, \Omega\}$. The mass function is called the *basic belief assignment* (BBA) [46] and the value of $m^\Omega(A)$ is called the *basic belief masses*, which indicates a degree of belief assigned to the hypothesis that x is associated with A. The subsets A of Ω with $m^\Omega(A) > 0$ are called *focal sets* of m^Ω. It is said to be normal that a BBA such that $m^\Omega(\emptyset) = 0$, but this condition may be relaxed if the *open-world assumption* state that the set Ω might be incomplete can be accepted [47]. That is to say, $m^\Omega(\emptyset)$ denotes a degree of belief assigned to the hypothesis that x might not lie in Ω.

Assume that there are two BBAs m_1 and m_2, which represent distinct items of evidence, to be combined. The standard way of the combination of m_1 and m_2 is to utilize the conjunctive sum operation $\textcircled{\cap}$ defined as

Table 1
An example of crisp partition.

	Cl_1	Cl_2	Cl_3
u_{1p}	1	0	0
u_{2p}	0	0	1
u_{3p}	0	1	0

Table 2
An example of fuzzy partition.

	Cl_1	Cl_2	Cl_3
f_{1p}	0.875	0.08	0.045
f_{2p}	0.183	0.208	0.609
f_{3p}	0.03	0.51	0.46

Table 3
An example of credal partition on $\Omega = \{Cl_1, Cl_2, Cl_3\}$.

	$\varnothing$	$\{Cl_1\}$	$\{Cl_2\}$	$\{Cl_3\}$	$\{Cl_1, Cl_2\}$	$\{Cl_1, Cl_3\}$	$\{Cl_2, Cl_3\}$	Ω
$m_1^\Omega(\cdot)$	0	0.8	0.02	0.01	0.1	0.05	0.02	0
$m_2^\Omega(\cdot)$	0	0.1	0.1	0.5	0.05	0.05	0.1	0.1
$m_3^\Omega(\cdot)$	0	0.01	0.35	0.3	0.02	0.02	0.3	0
$m_4^\Omega(\cdot)$	0	1	0	0	0	0	0	0
$m_5^\Omega(\cdot)$	1	0	0	0	0	0	0	0
$m_6^\Omega(\cdot)$	0	0	0	0	0	0	0	1

$$(m_1 \textcircled{\cap} m_2)(A) \triangleq \sum_{B \cap C = A} m_1(B) m_2(C), \tag{2.2}$$

for all $A \subseteq \Omega$. The *degree of conflict* between m_1 and m_2 is defined as

$$\mathcal{K} \triangleq (m_1 \textcircled{\cap} m_2)(\emptyset) = \sum_{B \cap C = \emptyset} m_1(B) m_2(C), \tag{2.3}$$

which denotes a degree of inconsistency between two information sources. If necessary, the normality condition $m^\Omega(\emptyset)$ can be recovered by dividing each mass $(m_1 \textcircled{\cap} m_2)(A)$ by $1 - \mathcal{K}$. Based on the conjunctive sum operation $\textcircled{\cap}$ and the degree of conflict, the *Dempster's rule of combination* is widely used to combine two BBAs:

$$(m_1 \oplus m_2)(A) \triangleq \frac{(m_1 \textcircled{\cap} m_2)(A)}{1 - \mathcal{K}} = \frac{1}{1 - \mathcal{K}} \sum_{B \cap C = A} m_1(B) m_2(C), \tag{2.4}$$

for all $A \subseteq \Omega$ and $A \neq \emptyset$. It is worth noting that, both combination rules have two excellent properties for combination operators, commutative and associative.

2.3. Credal partition

Let $U = \{x_1, x_2, \ldots, x_n\}$ be a set with n samples, and $\Omega = \{Cl_1, Cl_2, \ldots, Cl_C\}$ of C clusters is a partition of U. For the problem that which cluster each sample belongs to, if complete knowledge is available, then a *crisp partition* can be acquired (as shown in Table 1), which is represented by binary variables u_{ip}. Sample x_i belongs to cluster Cl_p if $u_{ip} = 1$ and sample x_i does not belong to cluster Cl_p if $u_{ip} = 0$. Assume that only partial knowledge is available, a *fuzzy partition* might be acquired (as shown in Table 2), which is represented by fuzzy membership f_{ip}. The probability that sample x_i belongs to cluster Cl_p is 0.8 if $f_{ip} = 0.8$. Furthermore, based on partial knowledge, a *credal partition* might be acquired (as shown in Table 3), which is represented by BBAs on the set Ω, as the following example illustrates.

Example 2.1. There is an example of credal partition on $U = \{x_1, x_2, x_3, x_4, x_5, x_6\}$ with six samples, and $\Omega = \{Cl_1, Cl_2, Cl_3\}$ of three clusters is showed in Table 3, whose element in Table 3 is the value of BBA for each sample. The situation of each sample is illustrated: The cases of samples x_1, x_2 and x_3 are the most common situation in a credal partition based on only partial knowledge. The credal partition not only gives the degree of belief that samples belong to single-clusters Cl_1, Cl_2 and Cl_3, but also gives the degree of belief that samples belong to meta-clusters $\{Cl_1, Cl_2\}$, $\{Cl_1, Cl_3\}$, $\{Cl_2, Cl_3\}$ and Ω. The cluster of sample x_4 is known certainly from $m_4^\Omega(Cl_1) = 1$, whereas the cluster of sample x_6 is absolutely unknown from $m_6^\Omega(\Omega) = 1$. Ultimately, it can be known that sample x_5 does not lie in Ω, which is indicated by $m_5^\Omega(\emptyset) = 1$.

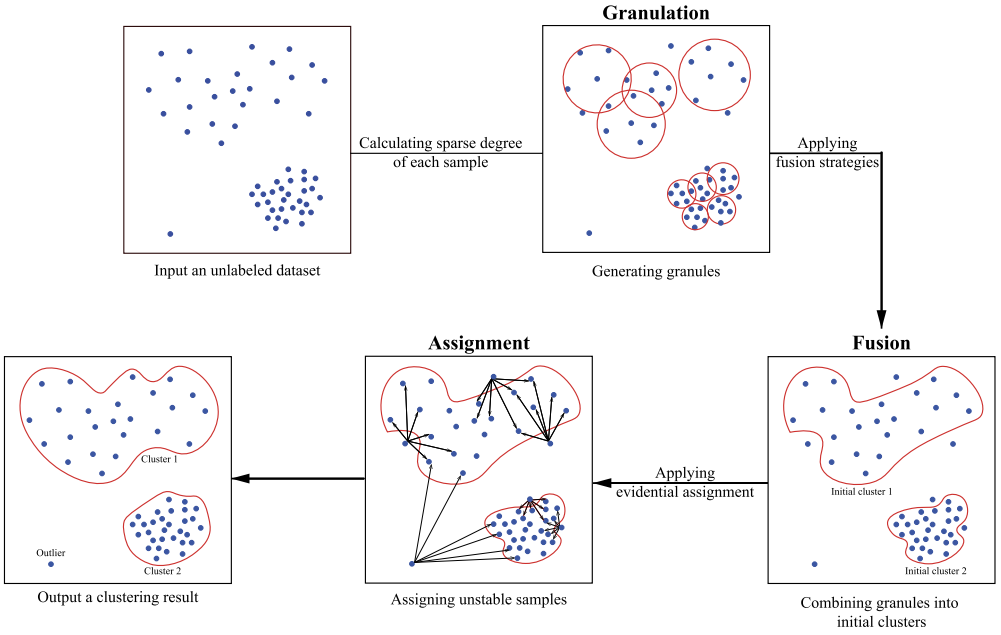


Fig. 1. The framework of the proposed algorithm.

As shown above, a *credal partition* of n samples $U = \{x_1, x_2, \ldots, x_n\}$ is defined as the n-tuple BBAs $M = \{m_1^\Omega, m_2^\Omega, \ldots, m_n^\Omega\}$ [39]. Interestingly, the *credal partition* can be regarded as a general model of partition, and the *crisp partition* and the *fuzzy partition* are two particular cases of the *credal partition*:

- When the *focal sets* of all BBAs m^Ω are singletons of Ω and the whole *basic belief masses* are allocated to a unique singleton of Ω, then *credal partition* M degenerates into *crisp partition*;
- When the *focal sets* of all BBAs m^Ω are singletons of Ω and all BBAs m^Ω are equivalent to probability functions, then *credal partition* M degenerates into *fuzzy partition*.

3. Density-based clustering algorithm with granulation, fusion and evidential assignment

This section aims at expounding the principle, process, and detail of the proposed clustering algorithm named GFDC. As illustrated by the framework of the proposed algorithm in Fig. 1, GFDC consists of three main steps: **(1) Granulation:** calculating the sparse degree of samples and aggregating the samples into core-granules based on the sparse degrees; **(2) Fusion:** fusing core-granules into stable granule-related structures and initial clusters; **(3) Assignment:** assigning remaining border samples.

In the first step, the sparse degree for measuring the densities of samples is introduced by finding the adaptive neighborhood radius and combining the idea of k-nearest neighbors. It is remarkably different from other density measure formulas, because the sparse degree can provide both local and global density information of a sample. After that, multiple core-granule are generated based on the sparse degree of samples, and the center of each core-granule is the peak density sample in the core-granule. In the second step, three fusion strategies are designed and used sequentially to fuse core-granule into different forms of stable granule-related structures and obtain initial clusters. What's more, these fusion strategies are based on intersection relationship, density transmission and distance respectively. In the third step, an improved assignment method on the basis of Dempster-Shafer theory is applied to assign the remaining border samples and identify outliers. The details of each step of GFDC are discussed in Section 3.1 to Section 3.4. At the end of this section, the pseudo-code of the proposed algorithm is demonstrated and the time complexity of GFDC is analyzed in Section 3.5.

3.1. Sparse degrees of samples

Without loss of generality, assume that the dataset U consists of n samples $\{x_1, x_2, \ldots, x_n\}$ and w attributes, with the attribute values of a sample x_i represented as $\{x_{i1}, x_{i2}, \ldots, x_{iw}\}$.

Considering the shortcomings of the density measure in DPC and its variations discussed above, a measure is introduced, which takes into consideration both local and global densities of samples, by combining the adaptive neighborhood radius and the radius of kth nearest neighbor.

At first, choosing different neighborhood radii, each sample can form the corresponding neighborhood:

$$\delta(x_i) = \{x_j \mid x_j \in U, d(x_i, x_j) \le \delta\},$$

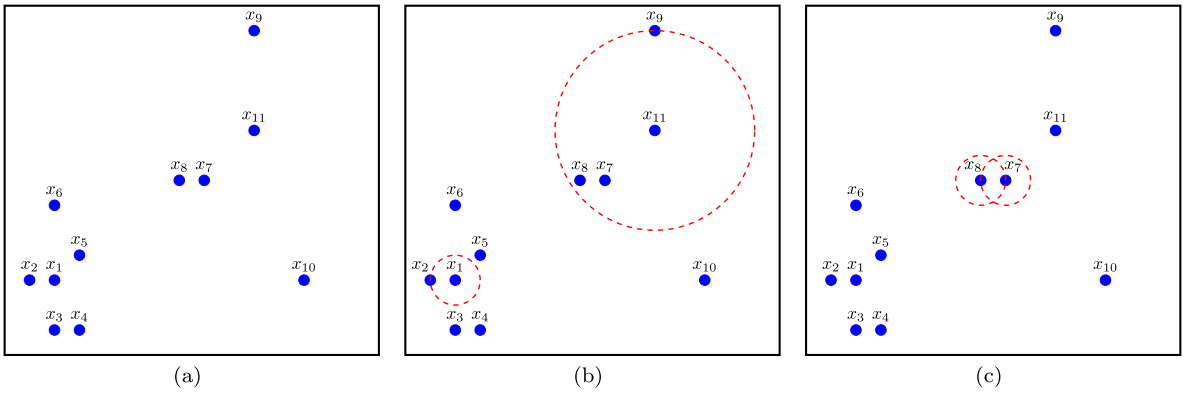


Fig. 2. Illustration for the density-limit-granules of samples.

where $d(x_i, x_j) = d_{ij} = ||x_i - x_j||_2$ denotes the Euclidean distance between samples x_i and x_j. The neighborhood radius δ is from 0 to $d(x'_i, x'_j)$, where x'_i and x'_j have a maximum distance in U. In fact, the meaningful neighborhood radius of a sample x_i should be the distance between x_i and any other sample in U:

$$\delta(x_i, d_{ij}) = \{x_z \mid x_z \in U, d(x_i, x_z) \leq d_{ij}\}. \tag{3.1}$$

In order to measure the density in the neighborhood of a sample, inspired by literature [48], an understandable formula can be used to express the relative density of a sample in its neighborhood with a radius:

$$\rho^*(x_i, d_{ij}) = \frac{|\delta(x_i, d_{ij})|}{(d_{ij})^w}, \tag{3.2}$$

where $|\cdot|$ denotes the cardinality of a set. The numerator of Eq. (3.2) indicates the number of samples in the w-dimensional hypersphere with sample x_i as the center and d_{ij} as the radius. The denominator is the wth power of the radius, where the intent of the power is to eliminate differences in dimensional space and make the relative density $\rho^*(x_i, d_{ij})$ can represent the size of samples per unit area.

Actually, different neighborhood radii can be regarded as corresponding to different information granularities. It is necessary to find a suitable information granularity to reflect the global density of samples in U. We consider that the neighborhood radius, which makes the relative density of a sample highest, is the optimal information granularity, and it can reflect the global density of the sample:

$$r^*(x_i) = \arg\max_{d_{ij}} \frac{|\delta(x_i, d_{ij})|}{(d_{ij})^w}, \tag{3.3}$$

for all $x_j \in U$. Further, according to the neighborhood $\delta(x_i, r^*(x_i)) = \{x_z \mid x_z \in U, d(x_i, x_z) \leq r^*(x_i)\}$, we define a circle, sphere or hypersphere, where sample x_i is the center and $r^*(x_i)$ is the radius, as the **density-limit-granule**. It can be known that a density-limit-granule has the highest relative density among all information granules centered on x_i, in other words, a density-limit-granule with $r^*(x_i)$ can reflect the density limitation of x_i. For the density-limit-granules of different samples, the higher the relative density of a density-limit-granule, the more samples have in the density-limit-granule or the smaller neighborhood radius has. Therefore, without considering the number of samples tentatively, if the neighborhood radius $r^*(x_i)$ of the density-limit-granule of sample x_i is smaller, it indicates that the relative density of the density-limit-granule of x_i is likely to be higher, namely, x_i is likely to be a sample with a higher density.

Example 3.1. In Fig. 2, there is a dataset containing 11 samples. As shown in Fig. 2(a), it can be found that sample x_1 is denser, whereas sample x_{11} is sparser. After calculating the neighborhood radius of the density-limit-granule of x_1 and x_{11} respectively, the density-limit-granules are plotted in Fig. 2(b). It can be visualized that sample x_1 has a density-limit-granule with a smaller neighborhood radius, i.e., sample x_1 is more likely to be a high-density sample. Sample x_{11} has a density-limit-granule with a larger neighborhood radius, i.e., sample x_{11} is less likely to be a low-density sample. This means that, to some extent, $r^*(x_i)$ may be able to measure the density of a sample. Moreover, $r^*(x_i)$ measures the global density because it is obtained by taking into account the relationships between a sample and all other samples.

However, as shown in Fig. 2(c), both samples x_7 and x_8 are sparser in terms of the whole dataset, but they are too close to each other such that the neighborhood radii of their density-limit-granules both are the distance between them, i.e., correspondingly $r^*(x_7)$ and $r^*(x_8)$ are small. That is to say, according to the above discussion about the relationships between the neighborhood radius of the density-limit-granule and the density, x_7 and x_8 are most likely to be mistaken as high-density samples.

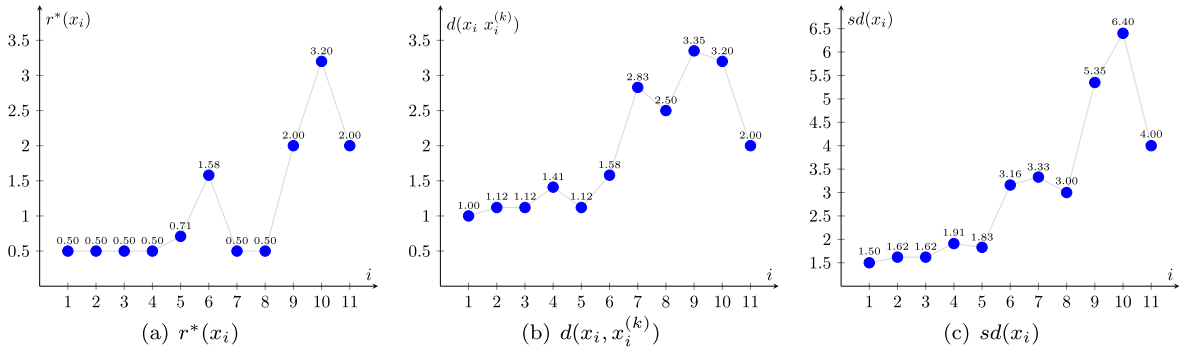


Fig. 3. The sparse degrees of samples in Example 3.1.

It can be believed that this problem stems from the lack of considering the number of samples within the density-limit-granule according to the above discussion. Nevertheless, since the number of samples within the density-limit-granule is correlated with the neighborhood radius, to avoid the impact of correlation, another density measurement is introduced. The radius corresponding to the kth nearest neighbor, which takes into account the number of samples and is able to measure the local density, is combined with the neighborhood radius of the density-limit-granule in this paper. Further, a new formula is proposed to measure both local and global densities of samples as below.

Definition 2 *(sparse degree).* The sparse degree of sample x_i, denoted by $sd(x_i)$, is defined by

$$sd(x_i) = r^*(x_i) + d\left(x_i, x_i^{(k)}\right), \tag{3.4}$$

where $x_i^{(k)}$ indicates the kth nearest neighbor of sample x_i and $d\left(x_i, x_i^{(k)}\right)$ means the distance between x_i and its kth nearest neighbor. Generally, the value of k is taken to be $\lceil log_2(n) \rceil$, where $\lceil \cdot \rceil$ denotes rounding up for real numbers.

Obviously, both $d\left(x_i, x_i^{(k)}\right)$ and $r^*(x_i)$ change in the same direction, which means that, given the value of k, the denser the sample, the smaller $d\left(x_i, x_i^{(k)}\right)$ is, and the sparser the sample, the larger $d\left(x_i, x_i^{(k)}\right)$ is. Therefore, it is reasonable to combine $d\left(x_i, x_i^{(k)}\right)$ and $r^*(x_i)$ to form a new and effective density measure.

To visualize the effectiveness of sparse degree, the following explanation is given based on Example 3.1. In the dataset shown in Fig. 2(a), taking $k = \lceil log_2(11) \rceil = 3$, $r^*(x_i)$, $d(x_i, x_i^{(k)})$ and sparse degrees $sd(x_i)$ of 11 samples are calculated and recorded in Fig. 3. From Fig. 3, it can be seen that sample x_1 with the highest density has the lowest sparse degree, $sd(x_1) = 1.50$, and sample x_{10} with the lowest density has the highest sparsity, $sd(x_{10}) = 6.40$. In addition, $sd(x_7) = 3.33$ and $sd(x_8) = 3.00$ reflect the density of samples x_7 and x_8 very well. This suggests that the sparse degree can indeed consider both the local and global densities of samples and give a suitable measure of density.

3.2. Granulation of samples

Definition 3 *(core-granule).* Given a positive integer of k, according to the sparse degree, a set of samples can be called a **core-granule** G in this paper, if two requirements are satisfied: $\exists x_i \in U$, such that
(1) $G = \{x_j \mid x_j \in U, d(x_i, x_j) \le d\left(x_i, x_i^{(k-1)}\right)\}$ and $|G| = k$;
(2) $sd(x_i) = \min\limits_{x_j \in G} sd(x_j)$.

Thus, there are k samples in a core-granule G and the center of G is with the lowest sparse degree within G. Furthermore, all core-granules and samples outside any core-granule can constitute a cover of the dataset, so the above process of finding core-granules can be known as **granulation**. In the later part of this paper, core-granules of the data are noted as $\{G_1, G_2, \ldots, G_g\}$ and a solid line circle is used in the illustration to represent a core-granule in the two-dimensional plane. A simple example of granulation is given below for the convenience of understanding.

Example 3.2. The dataset consists of 13 samples shown in Fig. 4, and k is taken to be 5 and the sparse degrees of samples are calculated firstly. According to Definition 3, it can be found that only samples x_3 and x_9 can generate core-granules because the 4-nearest neighbors of the other samples contain samples with lower sparse degrees than centers in Table 4. For instance, the 4-nearest neighbors of sample x_7 are samples x_9, x_6, x_{10} and x_{11}, but the sparse degrees of x_9, x_{10} and x_{11} are lower than that of x_7, so x_7 can not generate a core-granule. As shown in Fig. 4(a) and Fig. 4(b), it is similar to the above when k takes the value of 6. The sparse degrees of the 5-nearest neighbors of sample x_3 are higher than that of x_3 and the sparse degrees of the 5-nearest neighbors of sample x_9 are higher than that of x_9. While k is taken to be 7, the 6-nearest neighbors of sample x_3 contain samples x_1, x_5, x_2, x_4, x_6, and

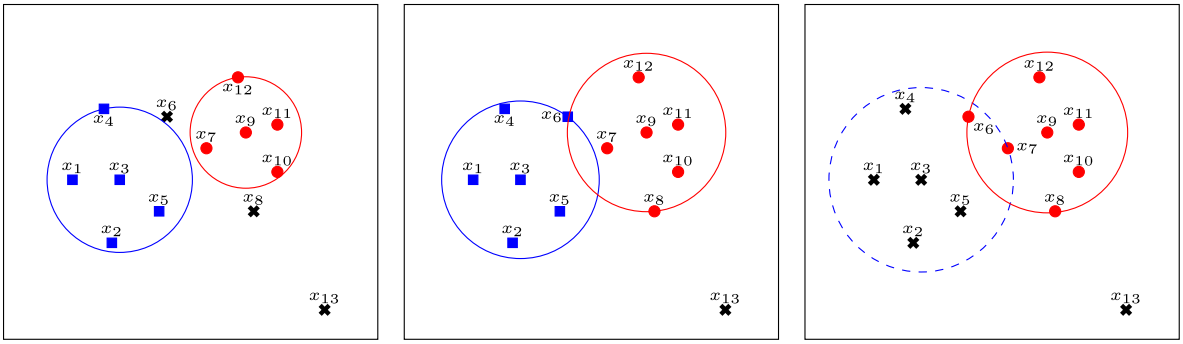


Fig. 4. Illustration of granulation.

Table 4
The sparse degrees of samples in Example 3.2.

Dataset		$k=5$		$k=6$		$k=7$	
x_i	(x_{i1}, x_{i2})	4-nearest neighbors	$sd(x_i)$	5-nearest neighbors	$sd(x_i)$	6-nearest neighbors	$sd(x_i)$
x_1	(4.00, 4.40)	x_3,x_2,x_4,x_5	2.04	x_3,x_2,x_4,x_5,x_6	2.35	x_3,x_2,x_4,x_5,x_6,x_7	2.88
x_2	(4.50, 3.60)	x_5,x_3,x_1,x_7	2.51	x_5,x_3,x_1,x_7,x_4	2.55	x_5,x_3,x_1,x_7,x_4,x_6	2.65
x_3	(4.60, 4.40)	x_1,x_5,x_2,x_4	1.64	x_1,x_5,x_2,x_4,x_6	1.81	x_1,x_5,x_2,x_4,x_6,x_7	2.35
x_4	(4.40, 5.30)	x_6,x_3,x_1,x_7	2.46	x_6,x_3,x_1,x_7,x_5	2.69	x_6,x_3,x_1,x_7,x_5,x_2	2.73
x_5	(5.10, 4.00)	x_3,x_2,x_7,x_1	1.92	x_3,x_2,x_7,x_1,x_8	1.93	x_3,x_2,x_7,x_1,x_8,x_6	2.20
x_6	(5.20, 5.20)	x_7,x_4,x_3,x_9	2.06	x_7,x_4,x_3,x_9,x_{12}	2.23	x_7,x_4,x_3,x_9,x_{12},x_5	2.43
x_7	(5.70, 4.80)	x_9,x_6,x_{10},x_{11}	1.98	x_9,x_6,x_{10},x_{11},x_{12}	2.00	x_9,x_6,x_{10},x_{11},x_{12},x_8	2.00
x_8	(6.30, 4.00)	x_{10},x_7,x_9,x_{11}	1.78	x_{10},x_7,x_9,x_{11},x_5	2.12	x_{10},x_7,x_9,x_{11},x_5,x_{13}	2.21
x_9	(6.20, 5.00)	x_{11},x_7,x_{10},x_{12}	1.42	x_{11},x_7,x_{10},x_{12},x_8	1.43	x_{11},x_7,x_{10},x_{12},x_8,x_6	1.90
x_{10}	(6.60, 4.50)	x_8,x_{11},x_9,x_7	1.94	x_8, x_{11},x_9,x_7,x_{12}	2.21	x_8, x_{11},x_9,x_7,x_{12},x_6	2.22
x_{11}	(6.60, 5.10)	x_9,x_{10},x_{12},x_7	1.55	x_9,x_{10},x_{12},x_7,x_8	1.82	x_9,x_{10},x_{12},x_7,x_8,x_6	2.27
x_{12}	(6.10, 5.70)	x_9, x_{11},x_7,x_6	2.08	x_9,x_{11},x_7,x_6,x_{10}	2.49	x_9, x_{11},x_7,x_6,x_{10},x_8	2.53
x_{13}	(7.20, 2.75)	x_8,x_{10},x_{11},x_5	5.62	x_8,x_{10},x_{11},x_5,x_9	5.70	x_8,x_{10},x_{11},x_5,x_9,x_7	5.99

x_7, where the sparse degrees of samples x_5 and x_7 are lower than that of sample x_3, so it is impossible to generate a core-granule with x_3 as the center. At last, only one core-granule with sample x_9 as the center can be generated, which is shown in Fig. 4(c).

With the above granulation method, a group of core-granules $\{G_1, G_2, \ldots, G_g\}$ with stable internal structure and clear density gradient can be obtained. For instance, the result of calculating sparse degrees of samples and granulation on the Jain dataset are shown in Fig. 5(a) and Fig. 5(b). The Jain dataset consists of two bowl-shaped clusters and there is a significant difference in the density of the two clusters. Observe Fig. 5(a) first, the color of a sample indicates its sparse degree. A sample with warm colors such as red, orange, yellow, etc., means that it is a high-density sample with a low sparse degree. And a sample with cold colors such as pink, purple, blue, etc., means that it is a low-density sample with a high sparse degree. After granulating the samples according to their sparse degrees, the black circles in Fig. 5(b) represent core-granules. It can be seen from the figure that the granules are generated not only in the high-density regions but also in the low-density regions, that is, samples with clear density gradients in the low-density regions can form core-granules too. Consequently, it indicates that the granulation method coupled with the sparse degrees of samples can well deal with the kind of datasets with large density differences among clusters. Also, this method can avoid a situation where excessive attention is paid to high-density clusters at the expense of low-density clusters.

With the concept of the core-granule, each center of core-granules represents a density peak within the core-granules, indicating the density in its neighborhood. In contrast, DPC determines the density peak for the entire dataset, which affects the assignment of all other samples and directly influences the final clustering result. Consequently, there is a higher risk of error propagation and the cost of incorrectly selecting density peak samples is increased. In our granulation method, unlike DPC, the density peak samples do not directly and individually impact the structure of the final clusters. This weakens the influence of the density peak samples and contributes to generating a better clustering result.

3.3. Initial clusters obtained by core-granule fusion

In the following, core-granules are fused according to the relationships among them, which leads to other forms of stable granule-related structures and initial clusters. Three fusion strategies for core-granules, based on intersection relationship, density transformation and distance respectively, are performed sequentially, which are beneficial to handling datasets with clusters of various shapes. These three fusion strategies are carried out sequentially, and possibly in a loop if special circumstances are encountered. The details of fusion strategies are described below.

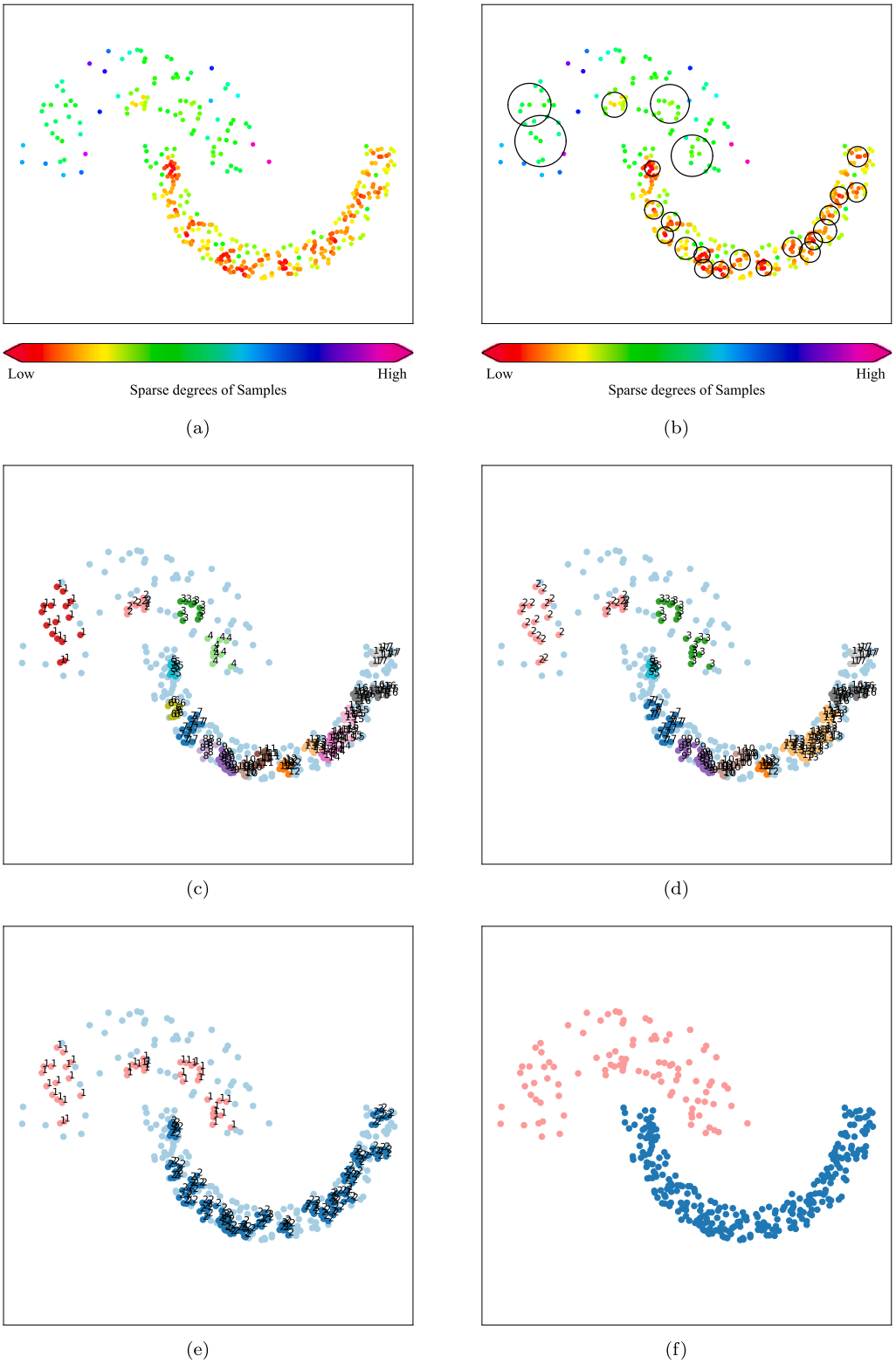


Fig. 5. The clustering process of the Jain dataset: (a) Sparse degrees of samples; (b) Granulation of samples; (c) Fusion based on intersection relationship; (d)Fusion based on density transmission; (e) Fusion based on distance; (f) Final clustering result. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

I. Fusion based on intersection relationship.

It is taken for granted that the samples within two core-granules are very similar if the two core-granules intersect. So, it is natural that fuse the core-granules based on their intersection relationship first.

Assume that any two core-granules are intersecting if they satisfy the condition: $|G_i \cap G_j| \geqslant 1$. On the basis of the intersection relationship of core-granules, all pairs of intersecting core-granules are fused and a set of **granule-clusters** (GCs) $\{GC_1, GC_2, \ldots, GC_{g^*}\}$ is generated, where g^* indicates the number of granule-clusters.

II. Fusion based on density transmission.

In order to fuse granule-clusters based on their characteristics of density, the idea of density transmission in the assignment strategy of DPC is improved and incorporated into the fusion strategy. This fusion strategy based on density transmission can make the shape of data distribution has little effect on the clustering results by transmitting low-density granule-clusters to high-density granule-clusters.

Since the idea of density transmission is derived, the definitions concerning distance and density are essential. Hence, the distance between granule-clusters is firstly defined as follows:

$$d^*(GC_a, GC_b) = \min\left\{d_{ij} \mid x_i \in GC_a, x_j \in GC_b\right\}, \tag{3.5}$$

where $a, b = 1, \ldots, g^*$ and $a \neq b$. Namely, the distance between the nearest pair of samples located in the different granule-clusters is taken as the distance between the two granule-clusters.

In order to measure the density of granule-clusters, the sparse degrees of samples are used to define the sparse degrees of granule-clusters:

$$sd^*(GC_a) = \frac{1}{|GC_a|} \sum_{x_i \in GC_a} sd(x_i). \tag{3.6}$$

The sparse degree $sd^*(\cdot)$ of a granule-cluster is expressed by the average sparse degree of samples.

With idea of density transmission, two granule-clusters will be fused if one granule-cluster satisfies the requirements of being the nearest neighbor of the other and having a lower sparse degree than the other. Exactly, granule-cluster GC_a can be fused into granule-cluster GC_b, if they satisfy two requirements:

(1) $d^*(GC_a, GC_b) = \min\{d^*(GC_a, GC_y) \mid GC_y \in \{GC_1, GC_2, \ldots, GC_{g^*}\} \wedge GC_y \neq GC_a\}$,

(2) $sd^*(GC_a) \geq sd^*(GC_b)$.

In other words, the closest granule-cluster GC_b to granule-cluster GC_a is found at first, and then if the sparse degree of GC_b is lower than or equal to that of GC_a, GC_a fuse into GC_b.

The above fusion strategy fuses granule-clusters with a high sparse degree into granule-clusters with a low sparse degree, resulting in the transmission of low density to high density. Such transmission is favorable to address the problem of detecting clusters with arbitrary shapes. Based on density transmission, a set of granule-clusters (GCs) $\{GC_1, GC_2, \ldots, GC_{g^*}\}$ are transformed into a set of **granule-flocks** (GFs) $\{GF_1, GF_2, \ldots, GF_{g'}\}$, where g' indicates the number of granule-flocks.

III. Fusion based on distance.

Like hierarchical agglomerative clustering, a set of granule-clusters or granule-flocks represents the initial clusters of a dataset at a certain granularity. For the purpose of being able to match practical requirements, given the number of clusters c ($c < n$), a set of initial clusters at a specified granularity can be formed by utilizing the following fusion strategy.

Unlike the previous two fusion strategies, the fusion strategy base on distance is dynamic. Namely, during each iteration, the operation of fusing a pair of granule-flocks will impact the operation in the next iteration. Details of the fusion strategy based on distance are as follows.

Similar to granule-clusters, the distance between two granule-flocks is defined as the distance between the pair of nearest samples located in the different granule-flocks:

$$d^*(GF_a, GF_b) = \min\left\{d_{ij} \mid x_i \in GF_a, x_j \in GF_b\right\}, \tag{3.7}$$

where $a, b = 1, \ldots, g'$ and $a \neq b$.

Given the number of clusters c, three relationships among c, the number of granule-clusters g^* and the number of granule-flocks g' are considered: $c = g'$ (or $c = g^*$), $c < g'$ and $c > g'$.

- For the case of $c = g'$ (or $c = g^*$), the GFs (or the GCs) are taken directly as the initial clusters, and then the remaining samples are assigned using the assignment method proposed in Section 3.4 to obtain a final clustering result.
- In the case of $c < g'$, the fusion strategy based on distance is applied: a pair of GFs with the smallest distance is fused in each iteration, and the number of GFs and the distance between GFs are updated after each iteration until the current number of GFs to c. At last, the initial clusters $\{Cl_1, Cl_2, \ldots, Cl_c\}$ are produced.
- To address the case of $c > g'$, each GF is considered as a new small sub-dataset, and then repeat the processes including calculating the sparse degrees of samples, granulation, fusion based on intersection relationship and fusion based on density transmission on all sub-datasets respectively. In particular, following the fusion based on density transmission, every single sample without any new GF in a sub-dataset is considered as a new GF. After that, the new GFs generated from all sub-datasets are put together to form a new set of GFs. By judging the relationships between the number of the new GFs and c, the corresponding solution is

selected according to the above discussion. What should be noted is that the iterations in this fusion strategy can definitely end up eventually, as described in the following remark.

Remark 1. For the case of $c > g'$, the iterations will be definitely stopped because sub-datasets become smaller during each round of iteration, i.e., the number of samples within each GF becomes smaller, and the limit case is that each GF contains only one sample. At this time, g' is equal to the number of samples within the core-granules and, in general, it will be greater than c. However, if a more extreme case occurs, where the dataset generates only one core-granule with k samples, i.e., $g' = k$, as well as $k < c$, the iterations will not be stopped. To cope with such a case, it is preferable to take k as equal to c in advance. Therefore, for a given c, the value of k in the proposed algorithm is taken to be $\max\{c, \lceil log_2(n) \rceil\}$ to avoid getting into a dead loop. Finally, the iterations must be stopped and the initial clusters $\{Cl_1, Cl_2, \ldots, Cl_c\}$ can be achieved.

Illustrative graphs of fusion based on intersection relationship, density transmission and distance on the Jain dataset are exemplified in Fig. 5(c) to Fig. 5(e). Above all, following the fusion strategy based on the intersection relationship, the intersecting core-granules are fused and 17 GCs are formed. As shown in Fig. 5(c), different colors and serial numbers are used to distinguish the different GCs where samples are located. Secondly, observing Fig. 5(c) and Fig. 5(d), 17 GCs are fused into 10 GFs by the fusion strategy based on density transmission. For a brief analysis, since the closest granule-cluster of GC_1 is GC_2 and $sd^*(GC_2)$ is lower than $sd^*(GC_1)$, the low-density GC_1 can be fused into the high-density GC_2; However, since the closest granule-clusters of GC_{12} is GC_{11} but $sd^*(GC_{11})$ is higher than $sd^*(GC_{12})$, the high-density GC_{12} cannot be fused into low-density GC_{11}. In the end, according to the case where the number of GFs is greater than the true number of clusters, 10 GFs are fused into 2 initial clusters using the fusion strategy based on distance and the result is displayed in Fig. 5(e). The samples marked with the number 1 indicate that they belong to the initial cluster Cl_1 and the samples marked with the number 2 indicate that they belong to the initial cluster Cl_2. The marked samples are core samples, while the others are border samples. From the above, it can be seen that GFDC can generate reasonable initial clusters and detect clusters with arbitrary shapes.

3.4. Evidential assignment of border samples

After the initial clusters obtained, samples within the initial clusters are defined as core samples which are with definite label information, and samples outside the initial clusters are defined as border samples (assume there are q border samples) which need to be further assigned. In order to assign these border samples as well as identify outliers, the EK-NN rule [49] based on the Dempster-Shafer theory is used to design a method for evidential assignment of border samples and create a credal partition of U.

Firstly, assume that the frame of discernment is $\Omega = \{Cl_1, Cl_2, \ldots, Cl_c\}$ and for all core samples in the initial clusters, i.e., $x_i \in Cl_u$, $u = 1, 2, \ldots, c$, the mass function m_i^{Ω} is defined as:

$$m_i^{\Omega}(Cl_u) = \begin{cases} 1, & \text{if } x_i \in Cl_u, \\ 0, & \text{if } x_i \notin Cl_u, \end{cases} \tag{3.8a}$$

$$m_i^{\Omega}(A) = 0, \tag{3.8b}$$

$$m_i^{\Omega}(\Omega) = 1 - \sum_u m_i^{\Omega}(Cl_u) - \sum_A m_i^{\Omega}(A) = 0, \tag{3.8c}$$

where $A \in 2^{\Omega} \backslash \{Cl_1, Cl_2, \ldots, Cl_c, \Omega\}$. The value of $m_i^{\Omega}(\cdot)$ represents the degree of belief that core sample x_i belongs to a single-cluster Cl_u, a meta-cluster A or Ω. It is worth noting that, according to the EK-NN rule and the purpose of this paper, the degree of belief that a sample belongs to a meta-cluster is not considered. Therefore, there is $m^{\Omega}(A) = 0$ for all samples.

Secondly, a stable set S with all core samples within the initial clusters is initialized. Next, sample x_i with the lowest sparse degree from the border samples outside S is considered. Then, pieces of evidence are calculated, which regard the label of x_i provided by k nearest neighbors of x_i within S. So, for an border sample x_i with the lowest sparse degree outside S, the corresponding mass function is defined as follows:

$$\begin{cases} m_{ij}^{\Omega}(Cl_u) = \exp(-(d_{ij} + sd(x_j))) \cdot m_j^{\Omega}(Cl_u), \\ m_{ij}^{\Omega}(\Omega) = 1 - \sum_u m_{ij}^{\Omega}(Cl_u), \end{cases} \tag{3.9}$$

for all $x_j \in N_k^S(x_i)$, where $N_k^S(x_i)$ denotes the set comprising k nearest neighbors of sample x_i in S. Here it is easy to know that k is less than or equal to $|S|$ because there are at least k samples in a set of core-granules according to the definition of the core-granule. The $m_{ij}^{\Omega}(Cl_u)$ denotes a piece of evidence that x_i belongs to Cl_u provided by x_j, and $m_{ij}^{\Omega}(\Omega)$ denotes a piece of evidence that x_i is an outlier provided by x_j. Unlike algorithms using the EK-NN rule [49], such as CBP-EKNN [22], when designing the mass function for border samples, not only the distance factor is taken into account but also the density factor by adding the sparse degrees of the nearest neighbors.

From Eq. (3.9), it can be seen that if x_j is far from x_i and x_j is a sparse sample, i.e., the distance between x_i and x_j is long and the sparse degree of x_j is high, the label of x_j will provide very little information regarding the label of x_i. In contrast, if x_j is close to x_i and x_j is a dense sample, i.e., the distance between x_i and neighbor x_j is short and the sparse degree of x_j is low, there will be a large degree of belief that x_i and x_j belong to the same cluster.

Thirdly, with the Dempster's rule of combination (Eq. (2.4)), these k pieces of evidence can be combined depending on

$$m_i^{\Omega}(\cdot) = \bigoplus_{x_j \in N_k^S(x_i)} m_{ij}^{\Omega}(\cdot). \tag{3.10}$$

After all basic belief masses of sample x_i are calculated, add the processed sample x_i to S and consider the next border sample with the lowest sparse degree outside S to repeat the above process until $S = U$.

Finally, after obtaining the credal partition of U, labels can be assigned to samples based on

$$x_i \in \arg\max_{Cl_u} m_i^{\Omega}(Cl_u). \tag{3.11}$$

This means that a sample is assigned to the cluster which makes the value of mass largest. Particularly, when it is desired to identify outliers, a threshold τ can be set such that if $m_i^{\Omega}(\Omega) > \tau$, sample x_i is considered as an outlier. Generally, τ can take the value from 0.99 to 1.

The result of the evidential assignment of the border samples for the Jain dataset is shown in Fig. 5(f), which is without a threshold for outliers. An example of evidential assignment of border samples with a threshold for outliers is illustrated in Section 4.1.

3.5. Analysis of algorithm

According to the exposition in Section 3.1 to Section 3.4, GFDC is summarized as Algorithm 1 and Algorithm 2. The details of the proposed algorithm for fusing core-granules into initial clusters are shown in the former, where the application of three fusion strategies for generating stable granule-related structures and initial clusters is carefully elaborated. The latter describes the complete algorithm.

In the following, the time complexity of GFDC is analyzed in terms of Algorithm 2 as follows:

- Computing the sparse degree in step 1: $O(wn(n-1)) + O(wn^2) + O(n(n-1)) + O(n^2)$, where w denotes the number of attributes, and n denotes the number of samples in a dataset. It includes calculating the matrix of distance, calculating the relative density, finding the maximum and sorting.
- The process of fusion in step 3: $O(g^*(g^*-1)/2 \cdot (n^2/(g^*)^2 - 1)) + O(g^*) + O(g^* - 1) + O((g' - c) \cdot g'(g'-1)/2 \cdot (n^2/(g')^2 - 1))$, where g^* denotes the number of granule-clusters, g' denotes the number of granule-flocks and c is the number of clusters. It includes calculating the distance between each pair of granule-clusters, calculating the sparse degree of each granule-cluster, finding the minimum, and calculating the distance between each pair of granule-flocks in iteration. Moreover, it should be noted that it is impossible to predict the time complexity for the case of $g' < c$.
- Creating the credal partition of the dataset and determining the labels in step 4 to step 11: $O(qn^2/2) + O(q \cdot (log_2(n))^2 \cdot (c+1)) + O(qc)$, where q represents the number of samples outside initial clusters. It includes calculating the masses and combining them, as well as finding the maximum.

In summary, the total time complexity of GFDC is approximately $O(qn^2)$.

4. Experimental results

In this section, to verify the validity of the proposed algorithm, some experiments are conducted on datasets, including 15 synthetic datasets[1] (as listed in Table 6) and 7 real-world datasets[2] (as listed in Table 13), which are commonly used to test the performance of a clustering algorithm. To be more convincing, the performance of the proposed algorithm is compared with several classical or novel and state-of-the-art clustering algorithms, including DBSCAN [5], RNN-DBSCAN [50], DPC [6], DPC-KNN [11], IM-DPC [51] and ECM [40]. All of these are density-based clustering algorithms except for ECM which is an evidential clustering algorithm. Experiments are performed on a computer with Intel(R) Xeon(R) E5-2650 v4 @ 2.20 GHz CPU with 64 GB RAM.

In the experiments, an internal index and three external indexes are made use of to quantify the comparative results:

- *Silhouette Coefficient* (SiC) [52]: SiC is an effective internal index for measuring clustering results if the ground truth is unknown. Both mean distances between each sample and all other samples in the same cluster and the next nearest cluster are considered to reflect the quality of the clustering result. It takes a value between -1 and 1, and the higher the SiC, the better the clustering results.
- *Adjusted Rank Index* (ARI) [53]: ARI is proposed to solve the problem that *Rank Index* (RI) cannot guarantee that its values are close to zero for random label assignments. Compared with RI, ARI has higher discrimination. ARI takes values in the range of $[-1, 1]$ and larger values mean that clustering results match the real situations.

[1] Synthetic datasets derived from the website: https://github.com/milaan9/Clustering-Datasets/tree/master/02.%20Synthetic.

[2] Real-world datasets derived from the website: http://archive.ics.uci.edu/ml/datasets.php.

Algorithm 1: Initial clusters obtained by core-granule fusion.

Input: The core-granules $\{G_1, G_2, \ldots, G_g\}$ and the number of clusters c.
Output: The initial clusters $\{Cl_1, Cl_2, \ldots, Cl_c\}$

for *each core-granule* G_i **do**
  for *each core-granule* G_j *and* $G_i \neq G_j$ **do**
    if $|G_i \cap G_j| \geqslant 1$ **then**
      Fuse G_i and G_j;
    end
  end
end
Renumber and get a set of granule-clusters $\{GC_1, GC_2, \ldots, GC_{g^*}\}$;
if $g^* = c$ **then**
  return *The initial clusters* $\{Cl_1, Cl_2, \ldots, Cl_c\} \leftarrow \{GC_1, GC_2, \ldots, GC_{g^*}\}$;
else
  Compute distance $d^*(GC_a, GC_b)$ between each pair of GCs according to Eq. (3.5);
  Compute sparse degree $sd^*(GC_a)$ of each GC according to Eq. (3.6);
  for *each granule-cluster* GC_a **do**
    Find the nearest granule-cluster GC_b of GC_a;
    if $sd^*(GC_a) \geq sd^*(GC_b)$ **then**
      Fuse GC_a into GC_b;
    end
  end
  Renumber and get a set of granule-flocks $\{GF_1, GF_2, \ldots, GF_{g'}\}$;
  if $g' = c$ **then**
    return *The initial clusters* $\{Cl_1, Cl_2, \ldots, Cl_c\} \leftarrow \{GF_1, GF_2, \ldots, GF_{g'}\}$;
  else if $g' > c$ **then**
    while $g' \neq c$ **do**
      Compute distance $d^*(GF_a, GF_b)$ between each pair of GFs according to Eq. (3.7);
      Find the pair of GFs which has minimal distance and fuse them;
      Update the GFs and $g' \leftarrow g' - 1$;
    end
    return *The initial clusters* $\{Cl_1, Cl_2, \ldots, Cl_c\} \leftarrow \{GF_1, GF_2, \ldots, GF_{g'}\}$;
  else
    Consider each GF as a new small sub-dataset;
    for *each sub-dataset* **do**
      Compute sparse degrees of samples and generate core-granules via steps 1 and 2 from Algorithm 2;
      Generate GCs and GFs via step 1 to step 20;
    end
    Aggregate all GFs generated from all sub-datasets into a new set of granule-flocks;
    Execute step 21 to step 38 based on the new set of granule-flocks.
  end
end

Algorithm 2: GFDC: A granule fusion density-based clustering with evidential reasoning.

Input: Dataset U, the number of cluster c and the threshold τ (if necessary).
Output: A clustering result and outliers (if threshold τ is given).

Compute sparse degree $sd(x_i)$ of each sample according to Eq. (3.4);
Generate the core-granules $\{G_1, G_2, \ldots, G_g\}$ based on sparse degrees of samples;
Fuse core-granules into initial clusters $\{Cl_1, Cl_2, \ldots, Cl_c\}$ via Algorithm 1;
Constitute a set S with samples in the initial clusters and compute the masses $m_{\cdot}^{\Omega}(Cl_u)$ of core samples according to Eq. (3.8a) to Eq. (3.8c);
while $|S| \neq |U|$ **do**
  Consider the unassigned sample x_i with lowest sparse degree in $U \backslash S$, find its k nearest neighbors in S and compute the masses $m_{ij}^{\Omega}(Cl_u)$ and $m_{ij}^{\Omega}(\Omega)$ according to Eq. (3.9);
  Combine the masses $m_i^{\Omega}(\cdot)$ according to Eq. (3.10);
  $S \leftarrow S \cup x_i$;
end
Assign each sample to the corresponding cluster according to Eq. (3.11) or treat it as an outlier.
return *A clustering result and outliers (if threshold* τ *is given).*

- *Adjusted Mutual Information* (AMI) [54]: *Mutual Information* (MI) is used to evaluate the similarity between the real labels and the clustering labels, whose common form is *Normalized Mutual Information* (NMI) and AMI. In particular, like ARI, AMI can identify the case of random labels. The value of AMI ranges from $[-1, 1]$, and the larger the value, the better the clustering result matches the real situation.
- *Clustering Purity* (CP) [55]: Purity is equivalent to the accuracy of the clustering result. It is equal to the number of correctly assigned samples divided by the total number of samples. Since the true class corresponding to each cluster of the clustering result is unknown, the maximum value in each case is taken. It takes a value between 0 and 1, and the larger the purity, the better the clustering results.

Table 5
Comparison of properties of algorithms. (Note that n is the number of samples, c is the number of clusters, p is the number of iterations, q is the number of samples outside initial clusters, Y represents that the algorithm is equipped with the function, and N is the opposite.)

Algorithm	Publication year	Cluster center determination	Density difference handling	Arbitrary shapes handling	Outliers identification	#adjustable parameter	Time complexity
DBSCAN	1996	N	weak	good	Y	2	$O(n^2)$
ECM	2008	Y	very weak	very weak	Y	3	$O(c^2pn)$
DPC	2014	Y	weak	weak	Y	1	$O(n^2)$
DPC-KNN	2016	Y	good	weak	Y	1	$O(n^2)$
RNN-DBSCAN	2018	N	weak	good	Y	1	$O(n^2)$
IM-DPC	2023	Y	good	good	Y	1	$O(n^2)$
GFDC		N	good	excellent	Y	2	$O(qn^2)$

Furthermore, the parameters of compared algorithms in the experiments are set as follows.

- For two parameters ϵ and $minpts$ in DBSCAN, the grid search method is used to find the optimal parameters for each dataset, where ϵ is searched in the range of 0.01 to the average distance of any pair of samples in the dataset, and $minpts$ is searched from 5 to 10 with a step size of 1, with a step size of 1.
- In RNN-DBSCAN, only one parameter k, searched from 1 to 100 with a step size of 1 [50], is set to the value that makes the clustering result optimal for each dataset.
- In DPC, there is a parameter cutoff distance d_c, and it can be defined as $d_c = D_{\lceil |D|\times p \rceil}$, where $D = \{D_1, D_2, \dots\}$ is a set of the distances between any pair of samples in the dataset and the distances are in ascending order [56]. And p is a user-defined parameter, which is found the best value from 0.01 to 0.90, with a step size of 0.01, for each dataset in our experiments. In addition, the first C samples with the largest γ are used as cluster centers automatically, where $\gamma = \delta \times \rho$, δ and ρ indicate the distance and density of samples respectively, and C is the true number of clusters.
- In DPC-KNN, the number of nearest neighbors is computed as p percent of the number of samples, and the parameter p is selected from $[0.1\%, 0.2\%, 0.5\%, 1\%, 2\%, 6\%]$ [11]. Also, the method of selecting cluster centers in DPC-KNN is the same as in DPC.
- IM-DPC is the Improved DPC algorithm in [51], the number of clusters in the algorithm is set to the true number of clusters and parameter k is selected from 1 to 30 with a step size of 1.
- For ECM, the number of clusters is set to the true number of clusters and the initial centers are given randomly. Focal elements of size less or equal to 2 and Ω are considered and the pignistic probability is used to transform a credal partition into a hard partition. Only the parameter α in ECM is searched from 1 to 3 with a step size of 0.5, parameters β and δ take the default values of 2 and 10 [40].

When adjusting the parameters of compared algorithms, a set of parameters that makes the value of ARI of a clustering result highest is considered as the optimal parameters and the corresponding clustering result is the optimal result of the algorithm. Meanwhile, for DBSCAN and RNN-DBSCAN, those results, where the number of detected clusters is the same as or closest to the true number of clusters and where there is as little noise as possible, are more likely to be considered as the optimal results.

4.1. Illustrative example

In this subsection, two illustrative examples are used to demonstrate the ability of GFDC for clustering and identifying outliers.

Example 4.1. The Aggregation dataset is considered in this example, consisting of 788 samples and 7 clusters, and the ground truth is shown in Fig. 6(a). This example shows the clustering result on the Aggregation dataset obtained by GFDC and analyzed it. Furthermore, two outliers, marked with red squares, are manually added to the dataset to test whether GFDC can identify them accurately.

Analyzing the clustering result without considering outliers first, in Table 9, the comparison results of $ARI_{DBSCAN} = 0.9065$, $ARI_{RNN-DBSCAN} = 0.9949$, $ARI_{DPC} = 0.9942$, $ARI_{DPC-KNN} = 0.9978$, $ARI_{IM-DPC} = 0.9920$, $ARI_{ECM} = 0.6312$ as well as $ARI_{GFDC} = 0.9949$ indicate that GFDC is good in this case. Here, in the result of the proposed method, samples 205 and 580 are misclassified samples. Sample 205 originally belongs to Cl_3, but due to $m^{\Omega}_{205}(Cl_2) = 0.0271$ and $m^{\Omega}_{205}(Cl_3) = 0.0131$, the sample is misclassified into Cl_2, and sample 580 originally belongs to Cl_5, but due to $m^{\Omega}_{580}(Cl_4) = 0.1024$ and $m^{\Omega}_{580}(Cl_5) = 0.0130$, the sample is misclassified into Cl_4. From Fig. 6, it can be seen that, for most clustering algorithms, samples like samples 205 and 508 which are at the junction of two clusters are very difficult to be classified correctly. In addition, the comparison of clustering results of other algorithms on the Aggregation dataset is shown in Fig. 9.

Analyzing the ability of GFDC to identify outliers next, there are two outliers located in the middle and upper left corner of the Aggregation dataset. In Fig. 6(b), it can be seen that, as $m^{\Omega}_{789}(\Omega) = 0.9985$ is close to 1 and $m^{\Omega}_{790}(\Omega) = 0.9959$ is close to 1, samples 789 and 790 can be identified as outliers. While considering the samples located at the boundaries of other clusters, there are $m^{\Omega}_{695}(\Omega) = 0.8982$, $m^{\Omega}_{728}(\Omega) = 0.8716$ and $m^{\Omega}_{755}(\Omega) = 0.6494$. Although some of the boundary samples have large values of $m^{\Omega}(\Omega)$, they are not very close to 1, so they are not identified as outliers.

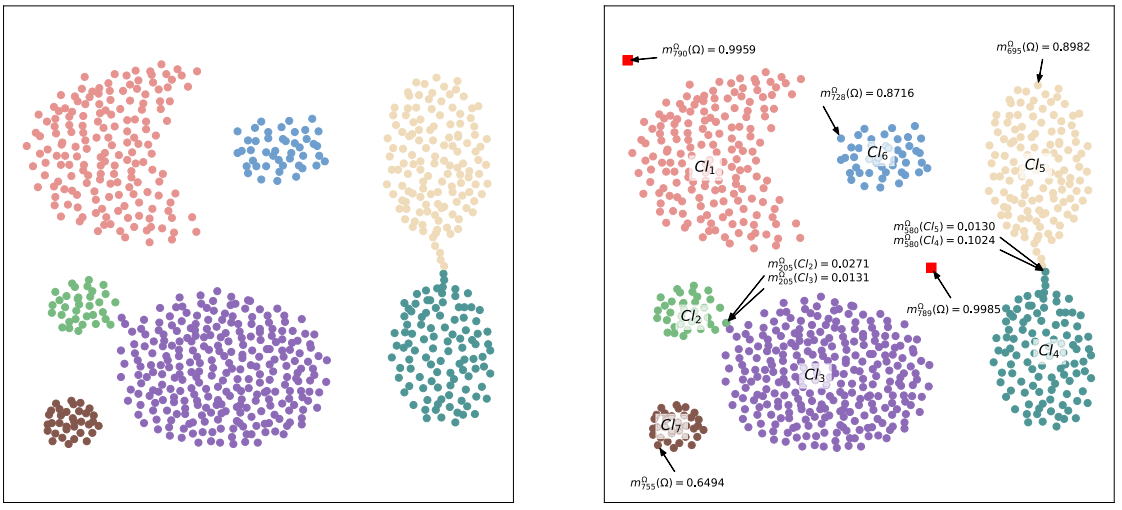


Fig. 6. Illustration of identifying outliers: (a) the ground truth of the Aggregation dataset; (b) the clustering result obtained by GFDC.

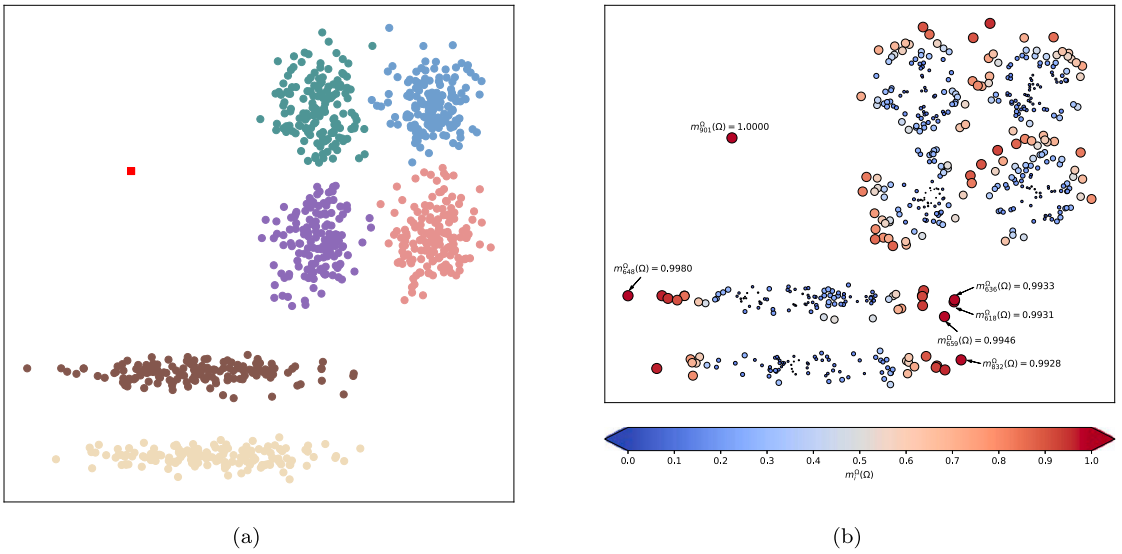


Fig. 7. Illustration of identifying outliers: (a) the ground truth of the Longsquare dataset; (b) the value of $m^{\Omega}_{x_i}(\Omega)$ obtained by GFDC.

Example 4.2. The Longsquare dataset is considered in this example, consisting of 900 samples and 6 clusters, and the ground truth is shown in Fig. 7(a). Furthermore, an outlier, marked with a red square in Fig. 7(a), is manually added to the dataset to test whether GFDC can identify it accurately.

In Fig. 7(b), the color and size of samples indicate the value of $m^{\Omega}_{x_i}(\Omega)$. The closer the color to red or the larger the size represents the larger the $m^{\Omega}_{x_i}(\Omega)$ of a sample, which corresponds to the greater probability of it being an outlier. It can be seen that the marked sample satisfies $m^{\Omega}_{901}(\Omega) = 1.0000$ and will be considered as an outlier, which is consistent with reality. Apart from the added outlier, there are five samples with the value of $m^{\Omega}_{x_i}(\Omega)$ between $[0.99, 1]$, which are located at the boundary of each cluster and far from the main structure of each cluster, so they are also considered as outliers. In addition, the value of $m^{\Omega}_{x_i}(\Omega)$ of samples at the boundaries of clusters is relatively large, which is reasonable.

In sum, it is thus clear that GFDC can effectively identify outliers in the examples. Based on the analysis of the above cases and recommendation of [23], the threshold τ is generally taken in the range of $[0.99, 1]$, which can be determined by the user according to the practice.

Table 6
Synthetic datasets.

Dataset	#sample	#dimension	#cluster	Dataset	#sample	#dimension	#cluster
2d-10c	2990	2	9	DS-577	577	2	3
2spiral	1000	2	2	DS-850	850	2	5
Aggregation	788	2	7	Jain	373	2	2
Banana-Ori	4811	2	2	Lsun	400	2	3
Cure-t0-2000n-2D	2000	2	3	Smile3	1000	2	4
Dartboard1	1000	2	4	Triangle2	1000	2	4
Donut2	1000	2	2	Zelnik3	266	2	3
Donut3	999	2	3				

Table 7
Parameters of clustering algorithms on synthetic datasets.

Dataset	DBSCAN				RNN-DBSCAN			DPC	DPC-KNN	IM-DPC	ECM
	eps	*minpts*	#detected cluster	#detected noise	*k*	#detected cluster	#detected noise	*p*	*p*	*k*	α
2d-10c	6.6684	5	8	0	14	13	104	0.62	0.001	2	1.5
2spiral	0.6116	5	2	0	2	2	0	0.31	0.002	13	3.0
Aggregation	1.8089	10	6	2	13	7	0	0.45	0.010	12	3.0
Banana - Ori	0.0494	5	2	0	27	2	36	0.16	0.010	5	2.5
Cure-t0-2000n-2D	0.1130	5	3	0	26	3	0	0.56	0.002	2	3.0
Dartboard1	0.0476	5	4	0	2	4	0	0.65	0.060	1	1.0
Donut2	0.0100	5	0	1000	9	2	17	0.30	0.002	15	1.0
Donut3	0.0181	5	3	1	9	3	26	0.33	0.060	1	2.5
DS-577	0.2594	8	3	93	7	3	17	0.02	0.005	8	2.0
DS-850	0.3006	5	5	19	17	5	19	0.03	0.005	5	3.0
Jain	3.2486	8	2	1	17	2	2	0.31	0.020	14	1.0
Lsun	0.5134	5	3	0	15	3	11	0.33	0.020	5	3.0
Smile3	0.0653	5	4	0	27	4	11	0.58	0.005	8	1.0
Triangle2	1.5225	8	4	61	14	4	30	0.06	0.002	4	3.0
Zelnik3	0.0338	5	3	0	12	3	0	0.67	0.020	3	2.5

4.2. Experiments on synthetic datasets and results analysis

The characteristics of all synthetic datasets used in experiments, including the number of samples, dimensions and clusters, are shown in Table 6. For the six compared clustering algorithms, the optimal parameters of each synthetic dataset are recorded in Table 7, where the parameters of the number of clusters that are necessary for some compared algorithms are set to the true number of clusters. Furthermore, the number of clusters and noises detected by DBSCAN and RNN-DBSCAN are shown in Table 7. The visual clustering results of seven algorithms on some datasets are shown in Fig. 8 to Fig. 12, where subfigures (a) to subfigures (h) represent the ground truth, the clustering results of DBSCAN, RNN-DBSCAN, DPC, DPC-KNN, IM-DPC, ECM and GFDC respectively. It should be noted that the red star markers represent the centers of the clusters detected by DPC, DPC-KNN, IM-DPC and ECM, and the black cross markers represent the noises identified by DBSCAN and RNN-DBSCAN. Besides, there are no thresholds set for identifying outliers in GFDC in synthetic datasets, because none of the datasets in the experiments contains outliers. Moreover, the evaluation of clustering results, including SiC, ARI, AMI and CP, of the six compared algorithms and the proposed algorithm on fifteen datasets are listed in Tables 8 to 11. Note that the bolded values in each row represent the best performance, and the symbol "-" indicates that the corresponding algorithm cannot produce valid clustering results. The running times of multiple algorithms are shown in Table 12. In each table, the values of ECM are the average results after ten experiments.

The analysis of the visual clustering results for some type of datasets is presented below.

Datasets with spiral clusters: As shown in Fig. 8, GFDC can detect correct structures of clusters without finding centers for datasets such as the 2spiral dataset, where there are no cluster centers in a geometric sense.

Datasets with multi-shaped clusters: The 2d-10c, Aggregation, DS-577, DS-850, Lsun and Triangle2 datasets are those that contain multiple shape clusters, and as can be seen from Table 8 to Table 11 and Fig. 9, GFDC performs well in most of this type of datasets. The performance of GFDC on the datasets 2d-10c, Aggregation and DS-850 are not the best, probably because the clusters are so close to each other that individual samples are misclassified, but in general GFDC outperforms most of the compared algorithms.

Datasets with arc-shaped clusters: In Fig. 10, the clustering result generated by GFDC of the Banana-Ori dataset is correct. In addition, from Table 8 to Table 11, GFDC performs perfectly on the Zelnik3 dataset, which is composed of arc-shaped clusters.

Datasets with cyclic-shaped clusters: The Dartboard1, Donut2, Donut3 (shown in Fig. 11) and Smile3 datasets are typical examples with cyclic-shaped clusters. GFDC always does best in these datasets, surpassing other compared algorithms.

Datasets with large differences in density among clusters: It can be seen in Fig. 12 that, the Jain dataset has two bowl-shaped clusters with different density. What's more, the Cure-t0-2000n-2D dataset is also the same type of dataset. Observing Table 8

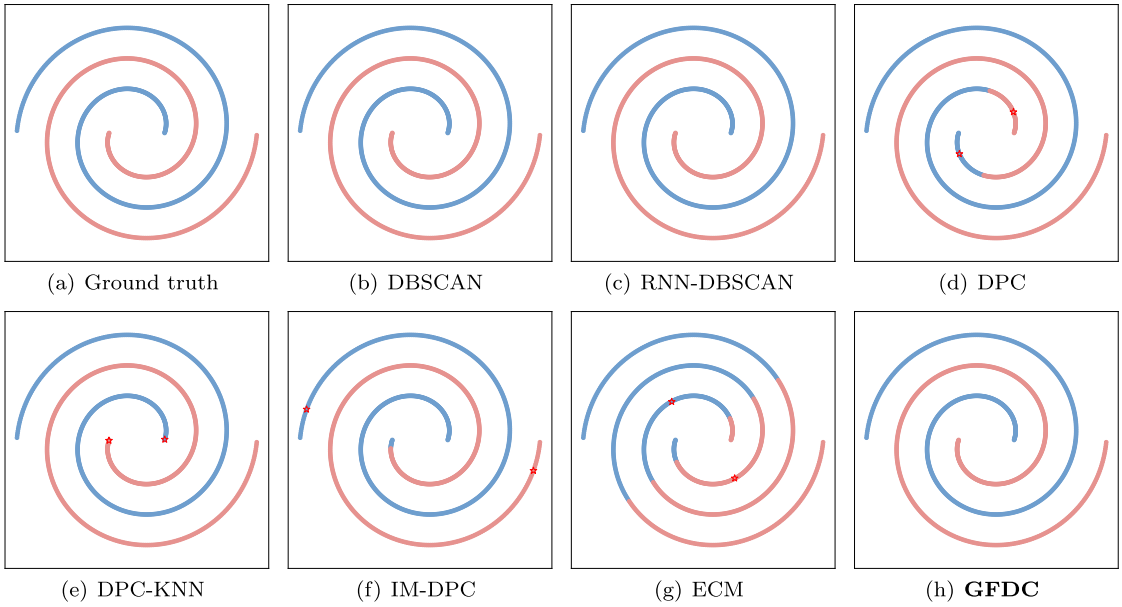


Fig. 8. 2spiral.

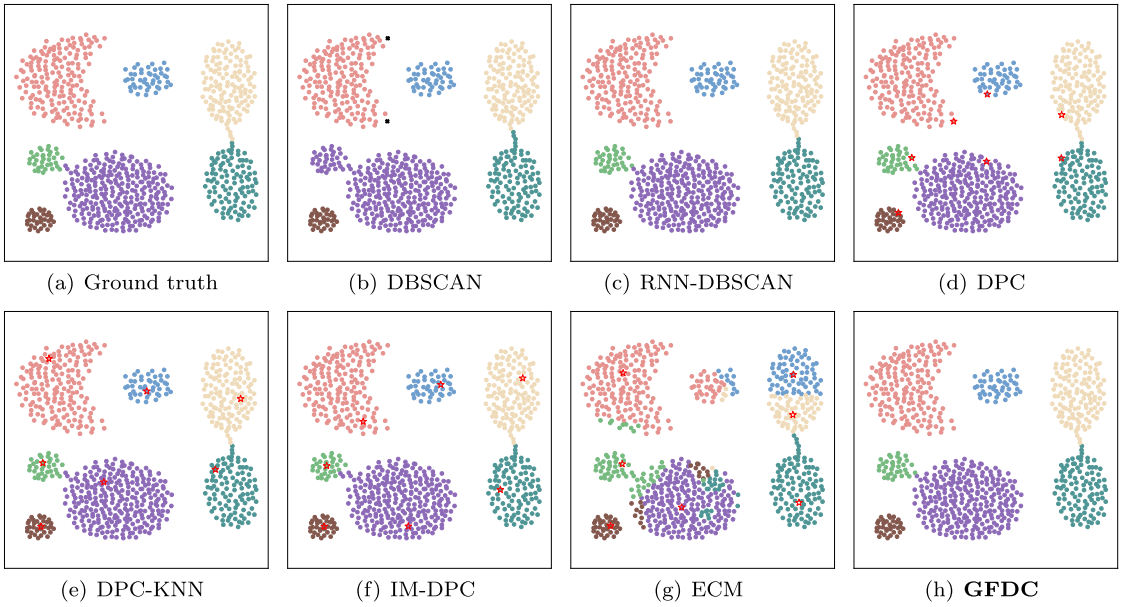


Fig. 9. Aggregation.

to Table 11, it is clear that only GFDC can handle this type of dataset accurately, without focusing too much on high-density clusters.

In general, from the illustration of some clustering results in Fig. 8 to Fig. 12, it is evident that GFDC can handle those datasets with clusters of arbitrary shapes well and also can cope well with those datasets where there are large differences in density among clusters. What's more, from the comparison results in Table 8 to Table 11, it can be said that GFDC outperforms the other six compared algorithms on some of the datasets and achieves as good results as them on some of the datasets. In conclusion, the performance of the proposed algorithm on synthetic datasets is excellent, the properties of each compared algorithm and GFDC are listed in Table 5.

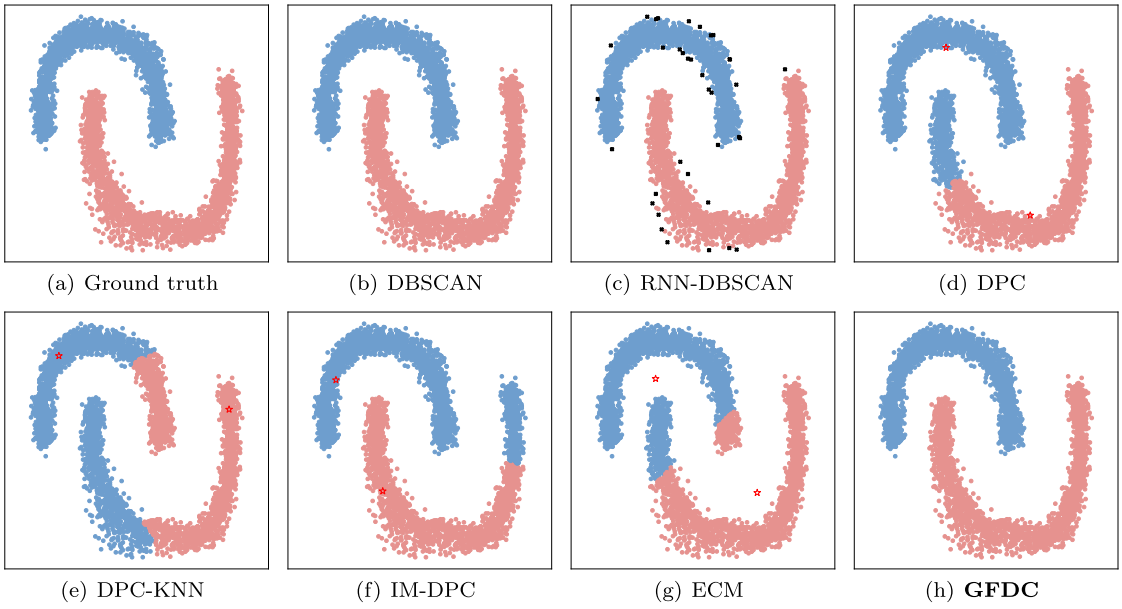


Fig. 10. Banana-Ori.

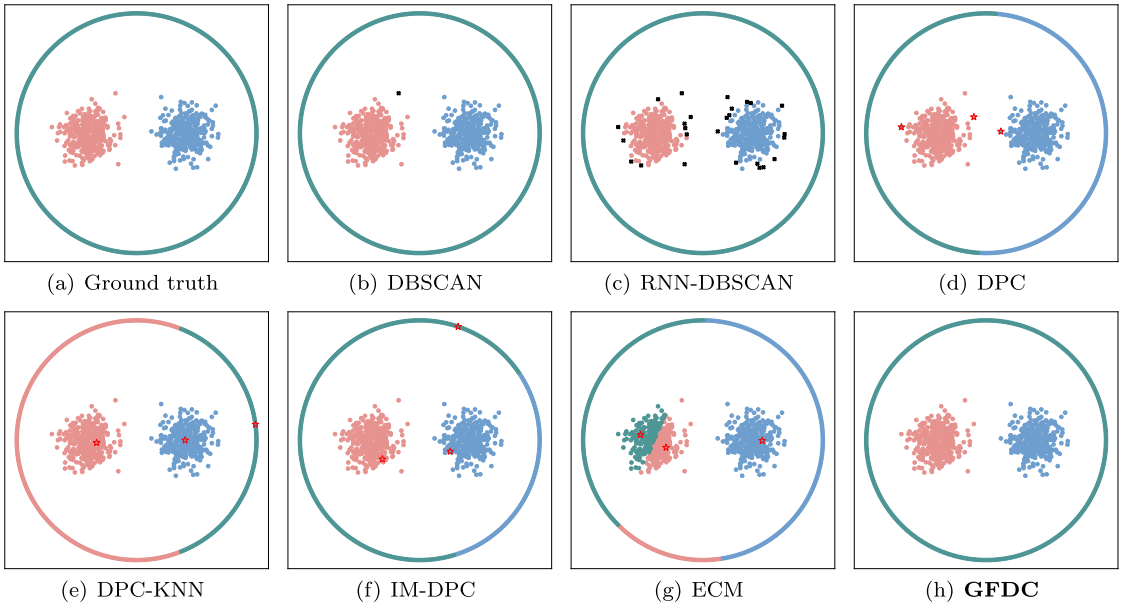


Fig. 11. Donut3.

4.3. Experiments on real-world datasets and results analysis

The characteristics of all real-world datasets used in experiments, including the number of samples, dimensions and clusters, are shown in Table 13. Before experiments, the datasets are preprocessed by first removing the missing values (the number of samples in Table 13 are the values after removing), and then standardizing each attribute of samples by the removal of the mean and the scaling to unit variance. For the six compared clustering algorithms, the optimal parameters of each real-world dataset are recorded in Table 14, where the parameters of the number of clusters that are necessary for some compared algorithms are set to the true number of clusters. Besides, the number of clusters and noises detected by DBSCAN and RNN-DBSCAN are shown in Table 14. The running times of multiple algorithms are shown in Table 12. Moreover, there are no thresholds set for identifying outliers in GFDC for real-world datasets, because none of the datasets in the experiments contains outliers. The evaluation of clustering results, including SiC, ARI, AMI and CP, of the six compared algorithms and the proposed algorithm on seven datasets are listed in Table 15.

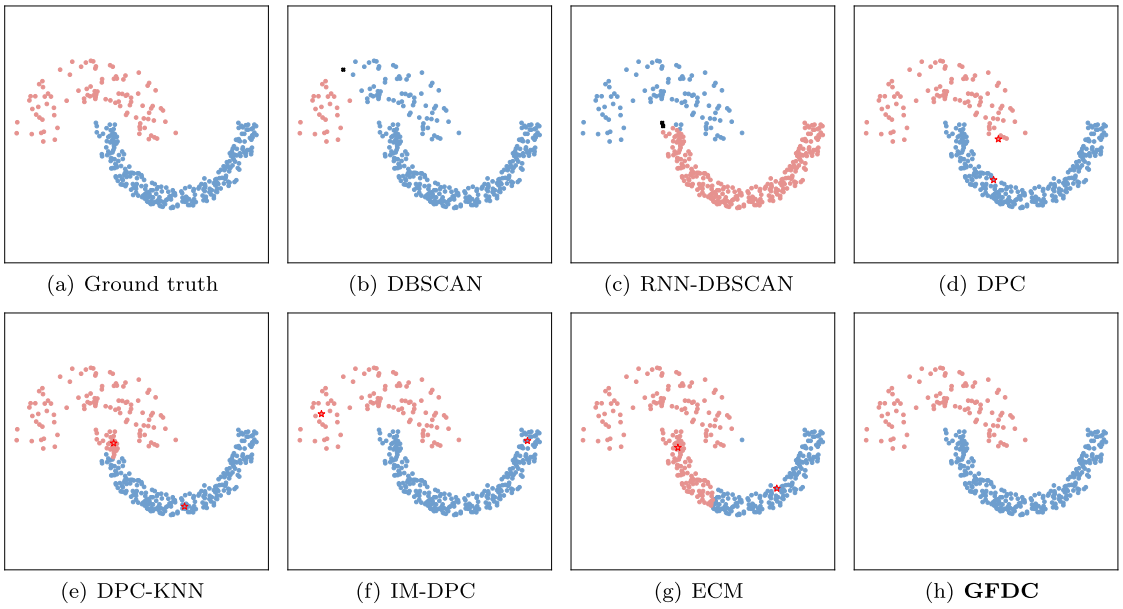


Fig. 12. Jain.

Table 8
Comparison of SiC for clustering algorithms on synthetic datasets.

Dataset	Algorithm						
	DBSCAN	RNN-DBCSAN	DPC	DPC-KNN	IM-DPC	ECM	**GFDC**
2d-10c	0.7393	0.5535	0.6201	**1.0000**	**1.0000**	0.2537	0.9987
2spiral	**1.0000**	**1.0000**	0.5501	**1.0000**	0.9554	0.0547	**1.0000**
Aggregation	0.5798	0.9873	0.9875	**0.9937**	0.9810	0.2339	0.9873
Banana-Ori	**1.0000**	0.9850	0.6575	0.0933	0.7335	0.5458	**1.0000**
Cure-t0-2000n-2D	**1.0000**	**1.0000**	**1.0000**	0.1248	**1.0000**	0.0444	**1.0000**
Dartboard1	**1.0000**	**1.0000**	-0.2904	-0.1557	0.0749	-0.0915	**1.0000**
Donut2	—	0.4980	0.2586	0.2011	0.3799	0.0029	**0.9880**
Donut3	0.6667	0.9479	0.4461	0.4600	0.6487	0.1538	**1.0000**
DS-577	0.6776	0.9302	**0.9948**	**0.9948**	**0.9948**	0.8946	**0.9948**
DS-850	0.9508	0.9552	0.9953	**1.0000**	**1.0000**	0.7668	0.9883
Jain	0.4021	0.2441	**1.0000**	0.7842	**1.0000**	0.3354	**1.0000**
Lsun	**1.0000**	0.7500	**1.0000**	0.8192	**1.0000**	0.1585	**1.0000**
Smile3	**1.0000**	0.9780	-0.1864	-0.0268	0.2161	-0.0054	**1.0000**
Triangle2	0.8750	0.9221	0.9880	0.9880	0.9801	0.7768	**0.9910**
Zelnik3	**1.0000**	**1.0000**	**1.0000**	0.2731	0.7061	0.2131	**1.0000**

Table 9
Comparison of ARI for clustering algorithms on synthetic datasets.

Dataset	Algorithm						
	DBSCAN	RNN-DBCSAN	DPC	DPC-KNN	IM-DPC	ECM	**GFDC**
2d-10c	0.8649	0.8935	0.8803	**1.0000**	**1.0000**	0.7958	0.9992
2spiral	**1.0000**	**1.0000**	0.4619	**1.0000**	0.9408	0.0322	**1.0000**
Aggregation	0.9065	0.9949	0.9942	**0.9978**	0.9920	0.6312	0.9949
Banana-Ori	**1.0000**	0.9849	0.5720	0.0578	0.6607	0.4570	**1.0000**
Cure-t0-2000n-2D	**1.0000**	**1.0000**	**1.0000**	0.4019	**1.0000**	0.2599	**1.0000**
Dartboard1	**1.0000**	**1.0000**	-0.0014	0.0353	0.2523	0.0114	**1.0000**
Donut2	—	0.9632	0.1842	0.1516	0.2889	0.0015	**0.9840**
Donut3	0.9985	0.9612	0.6390	0.5977	0.7501	0.3903	**1.0000**
DS-577	0.7529	0.9443	**0.9949**	**0.9949**	**0.9949**	0.8983	**0.9949**
DS-850	0.9657	0.9690	0.9966	**1.0000**	**1.0000**	0.8487	0.9944
Jain	0.2654	0.9501	**1.0000**	0.7055	**1.0000**	0.2172	**1.0000**
Lsun	**1.0000**	0.9736	**1.0000**	0.8197	**1.0000**	0.3519	**1.0000**
Smile3	**1.0000**	0.9905	0.4432	0.2846	0.3666	0.2016	**1.0000**
Triangle2	0.9042	0.9451	0.9867	0.9867	0.9810	0.8413	**0.9900**
Zelnik3	**1.0000**	**1.0000**	**1.0000**	0.4510	0.7657	0.4083	**1.0000**

Table 10
Comparison of AMI for clustering algorithms on synthetic datasets.

Dataset	Algorithm						
	DBSCAN	RNN-DBCSAN	DPC	DPC-KNN	IM-DPC	ECM	**GFDC**
2d-10c	0.9532	0.9135	0.9271	**1.0000**	**1.0000**	0.8823	0.9989
2spiral	**1.0000**	**1.0000**	0.3652	**1.0000**	0.9024	0.0233	**1.0000**
Aggregation	0.9427	0.9914	0.9926	**0.9956**	0.9882	0.7565	0.9914
Banana-Ori	**1.0000**	0.9659	0.5652	0.0410	0.6327	0.3655	**1.0000**
Cure-t0-2000n-2D	**1.0000**	**1.0000**	**1.0000**	0.5982	**1.0000**	0.4168	**1.0000**
Dartboard1	**1.0000**	**1.0000**	0.0017	0.0765	0.3516	0.0259	**1.0000**
Donut2	—	0.9393	0.1839	0.2586	0.3628	0.0011	**0.9663**
Donut3	0.9969	0.9442	0.7341	0.7177	0.7986	0.4488	**1.0000**
DS-577	0.7560	0.9102	**0.9901**	**0.9901**	**0.9901**	0.8682	**0.9901**
DS-850	0.9583	0.9603	0.9953	**1.0000**	**1.0000**	0.8463	0.9920
Jain	0.2458	0.9049	**1.0000**	0.6439	**1.0000**	0.3078	**1.0000**
Lsun	**1.0000**	0.9597	**1.0000**	0.8327	**1.0000**	0.4379	**1.0000**
Smile3	**1.0000**	0.9790	0.6341	0.5250	0.5413	0.3770	**1.0000**
Triangle2	0.8869	0.9128	0.9811	0.9811	0.9718	0.8325	**0.9865**
Zelnik3	**1.0000**	**1.0000**	**1.0000**	0.5614	0.8108	0.5204	**1.0000**

Table 11
Comparison of CP for clustering algorithms on synthetic datasets.

Dataset	Algorithm						
	DBSCAN	RNN-DBCSAN	DPC	DPC-KNN	IM-DPC	ECM	**GFDC**
2d-10c	0.8870	0.9729	0.9207	**1.0000**	**1.0000**	0.8877	0.9997
2spiral	**1.0000**	**1.0000**	0.8400	**1.0000**	0.9850	0.5910	**1.0000**
Aggregation	0.9530	0.9975	0.9975	**0.9987**	0.9962	0.8608	0.9975
Banana-Ori	**1.0000**	0.9975	0.8782	0.6205	0.9065	0.8381	**1.0000**
Cure-t0-2000n-2D	**1.0000**	**1.0000**	**1.0000**	0.9000	**1.0000**	0.8489	**1.0000**
Dartboard1	**1.0000**	**1.0000**	0.2550	0.3590	0.5580	0.3025	**1.0000**
Donut2	—	**0.9990**	0.7150	0.6950	0.7690	0.5219	0.9960
Donut3	**1.0000**	0.9890	0.8338	0.7938	0.9009	0.6781	**1.0000**
DS-577	0.9047	0.9792	**0.9983**	**0.9983**	**0.9983**	0.9653	**0.9983**
DS-850	0.9871	0.9859	0.9988	**1.0000**	**1.0000**	0.9292	0.9976
Jain	0.8123	0.9920	**1.0000**	0.9223	**1.0000**	0.7399	**1.0000**
Lsun	**1.0000**	**1.0000**	**1.0000**	0.9375	**1.0000**	0.6875	**1.0000**
Smile3	**1.0000**	0.9990	0.7000	0.6260	0.7220	0.6220	**1.0000**
Triangle2	0.9730	0.9750	0.9960	0.9960	0.9940	0.9395	**0.9970**
Zelnik3	**1.0000**	**1.0000**	**1.0000**	0.7744	0.9173	0.7462	**1.0000**

Table 12
Comparison of running time.

Dataset	DBSCAN	RNN-DBSCAN	DPC	DPC-KNN	IM-DPC	ECM	GFDC
2d-10c	0.0262	0.2517	0.2608	0.1902	0.3005	265.19	64.692
2spiral	0.0042	0.0331	0.0422	0.0284	0.1224	5.3367	3.4152
Aggregation	0.0032	0.0246	0.0295	0.0215	0.0920	45.123	1.0854
Banana-Ori	0.0461	0.7044	0.5807	0.4475	0.7758	6.9145	299.97
Cure-t0-2000n-2D	0.0108	0.1528	0.1396	0.0965	0.1645	7.6405	18.055
Dartboard1	0.0041	0.0317	0.0432	0.0349	0.0515	5.8595	1.3376
Donut2	0.0066	0.0339	0.0424	0.0294	0.1352	0.5536	2.1287
Donut3	0.0072	0.0331	0.0427	0.0363	0.0528	1.1937	1.9190
DS-577	0.0025	0.0141	0.0186	0.0138	0.0509	1.1274	0.5652
DS-850	0.0034	0.0303	0.0322	0.0235	0.0634	6.8731	1.4759
Jain	0.0019	0.0102	0.0101	0.0085	0.0435	0.9528	0.1771
Lsun	0.0021	0.0108	0.0117	0.0089	0.0257	1.2157	0.2193
Smile3	0.0052	0.0460	0.0404	0.0300	0.0924	8.9769	1.8358
Triangle2	0.0049	0.0354	0.0429	0.0289	0.0719	13.635	2.3452
Zelnik3	0.0015	0.0063	0.0070	0.0060	0.0137	0.7780	0.0755
Breast-Cancer-Wisconsin	0.0140	0.0224	0.0241	0.0210	0.0367	0.7630	1.3112
EEG Eye State	5.5046	13.364	5.4916	5.9085	7.9311	142.33	10460.6
Ionosphere	0.0026	0.0131	0.0108	0.0321	0.0916	0.2532	0.1848
Seeds	0.0014	0.0054	0.0046	0.0037	0.0248	0.6578	0.0608
SPECT-heart	0.0014	0.0181	0.0063	0.0067	0.0515	0.3961	0.1230
WDBC	0.0059	0.0262	0.0195	0.0246	0.0426	1.3308	0.4613
Yeast	0.0439	0.0886	0.0828	0.0860	0.3337	78.505	10.103

Table 13
Real-world datasets.

Dataset	#sample	#dimension	#cluster
Breast-Cancer-Wisconsin	699	9	2
EEG Eye State	14980	14	2
Ionosphere	351	34	2
Seeds	210	7	3
SPECT-heart	267	22	2
WDBC	569	30	2
Yeast	1484	8	10

Table 14
Parameters of clustering algorithms on real-world datasets.

Dataset	DBSCAN				RNN-DBSCAN			DPC	DPC-KNN	IM-DPC	ECM
	eps	*minpts*	#detected cluster	#detected noise	*k*	#detected cluster	#detected noise	*p*	*p*	*k*	α
Breast-Cancer-Wisconsin	1.9788	7	2	87	2	49	92	0.43	0.010	1	3.0
EEG Eye State	0.8366	7	2	12	26	3	33	0.06	0.005	3	1.0
Ionosphere	2.5203	6	2	132	5	2	87	0.20	0.020	17	1.0
Seeds	0.7586	8	3	122	14	2	12	0.07	0.010	14	3.0
SPECT-heart	4.3030	5	2	51	71	1	37	0.90	0.001	26	1.0
WDBC	2.3456	5	2	264	9	1	32	0.49	0.020	2	1.0
Yeast	0.7837	5	6	687	3	15	42	0.11	0.010	28	2.5

Table 15
Comparison of SiC, ARI, AMI and CP for clustering algorithms on real-world datasets.

Dataset		Algorithm						
		DBSCAN	RNN-DBSCAN	DPC	DPC-KNN	IM-DPC	ECM	**GFDC**
Breast-Cancer-Wisconsin	SiC	0.1320(5)	-0.4735(6)	0.8193(4)	-0.5335(7)	**0.8897**(1)	0.8523(3)	0.8647(2)
	ARI	0.2559(5)	0.0531(6)	0.7660(4)	-0.0027(7)	**0.8551**(1)	0.8073(3)	0.8230(2)
	AMI	0.2478(5)	0.2131(6)	0.6483(4)	-0.0019(7)	**0.7566**(1)	0.6981(3)	0.7191(2)
	CP	0.7725(6)	0.8927(5)	0.9385(4)	0.6552(7)	**0.9628**(1)	0.9499(3)	0.9542(2)
EEG Eye State	SiC	-0.5142(6)	-0.9797(7)	**0.0233**(1)	0.0204(2)	0.0204(2)	0.0197(4)	-0.1393(5)
	ARI	-0.0003(7)	0.0008(6)	**0.0129**(1)	0.0018(3)	0.0018(3)	0.0104(2)	0.0012(5)
	AMI	0.0010(5)	**0.0188**(1)	0.0099(2)	0.0010(5)	0.0010(5)	0.0048(4)	0.0057(3)
	CP	0.5512(7)	**0.5569**(1)	**0.5569**(1)	0.5533(4)	0.5533(4)	0.5533(4)	0.5537(3)
Ionosphere	SiC	-0.2329(6)	-0.4866(7)	0.1856(5)	0.2876(3)	0.3069(2)	0.2534(4)	**0.5442**(1)
	ARI	0.4727(3)	0.5048(2)	0.1200(7)	0.2015(5)	0.2191(4)	0.1729(6)	**0.5185**(1)
	AMI	**0.4289**(1)	0.4413(2)	0.0858(7)	0.1229(6)	0.1387(4)	0.1323(5)	0.4173(3)
	CP	**0.8974**(1)	0.8718(2)	0.6752(7)	0.7293(5)	0.7379(4)	0.7094(6)	0.8376(3)
Seeds	SiC	-0.1896(7)	0.2244(6)	0.7366(4)	0.7268(5)	0.7370(3)	**0.7402**(1)	0.7396(2)
	ARI	0.1976(7)	0.4616(6)	0.7895(2)	0.7744(5)	**0.7902**(1)	0.7846(4)	0.7877(3)
	AMI	0.3904(7)	0.5053(6)	0.7486(2)	0.7283(5)	**0.7562**(1)	0.7367(4)	0.7388(3)
	CP	0.6524(6)	0.6429(7)	**0.9238**(1)	0.9190(5)	**0.9238**(1)	**0.9238**(1)	**0.9238**(1)
SPECT-heart	SiC	-0.2122(6)	0.1016(4)	0.1305(3)	-0.7597(7)	0.1368(2)	0.0405(5)	**0.1496**(1)
	ARI	0.0610(4)	0.0497(5)	0.0798(3)	0.0017(7)	**0.0828**(1)	0.0225(6)	0.0825(2)
	AMI	**0.0482**(1)	0.0305(5)	0.0451(4)	-0.0040(7)	0.0480(3)	0.0173(6)	0.0481(2)
	CP	0.6367(4)	0.6292(5)	0.6479(3)	0.5918(6)	0.6517(2)	0.5880(7)	**0.6592**(1)
WDBC	SiC	-0.1925(7)	0.1284(6)	0.6041(4)	0.5432(5)	**0.8094**(1)	0.7747(2)	0.6794(3)
	ARI	0.1764(6)	0.0344(7)	0.5106(4)	0.4402(5)	**0.7548**(1)	0.7130(2)	0.6931(3)
	AMI	0.1363(6)	0.0204(7)	0.4323(5)	0.4379(4)	**0.6472**(1)	0.5921(3)	0.6289(2)
	CP	0.7118(6)	0.6450(7)	0.8576(4)	0.8366(5)	**0.9350**(1)	0.9227(2)	0.8875(3)
Yeast	SiC	-0.4488(3)	-0.7883(7)	-0.5005(6)	-0.4090(2)	-0.4829(5)	-0.4820(4)	**-0.3067**(1)
	ARI	0.0333(6)	0.0159(7)	0.0895(5)	0.1950(3)	**0.2277**(1)	0.1101(4)	0.2230(2)
	AMI	0.0617(7)	0.0642(6)	0.1550(4)	0.2510(3)	0.2955(2)	0.1250(5)	**0.3079**(1)
	CP	0.3720(5)	0.3403(7)	0.4353(4)	0.4704(3)	**0.5168**(1)	0.3480(6)	0.4953(2)
Average rank of SiC		5.71	6.14	3.86	4.43	2.29	3.29	**2.14**
Average rank of ARI		5.43	5.57	3.71	5.00	**1.71**	3.86	2.57
Average rank of AMI		4.57	4.71	4.00	5.29	2.43	4.29	**2.29**
Average rank of CP		5.00	4.86	3.43	5.00	**2.00**	4.14	2.14

[1] The bolded values in each row represent the best performance.
[2] The values in brackets represent the rank of the value before the brackets in each row. If the value is the same in a row, their ranks are shared equally.
[3] The SiC, ARI, AMI and CP values of ECM are the average results after ten experiments.

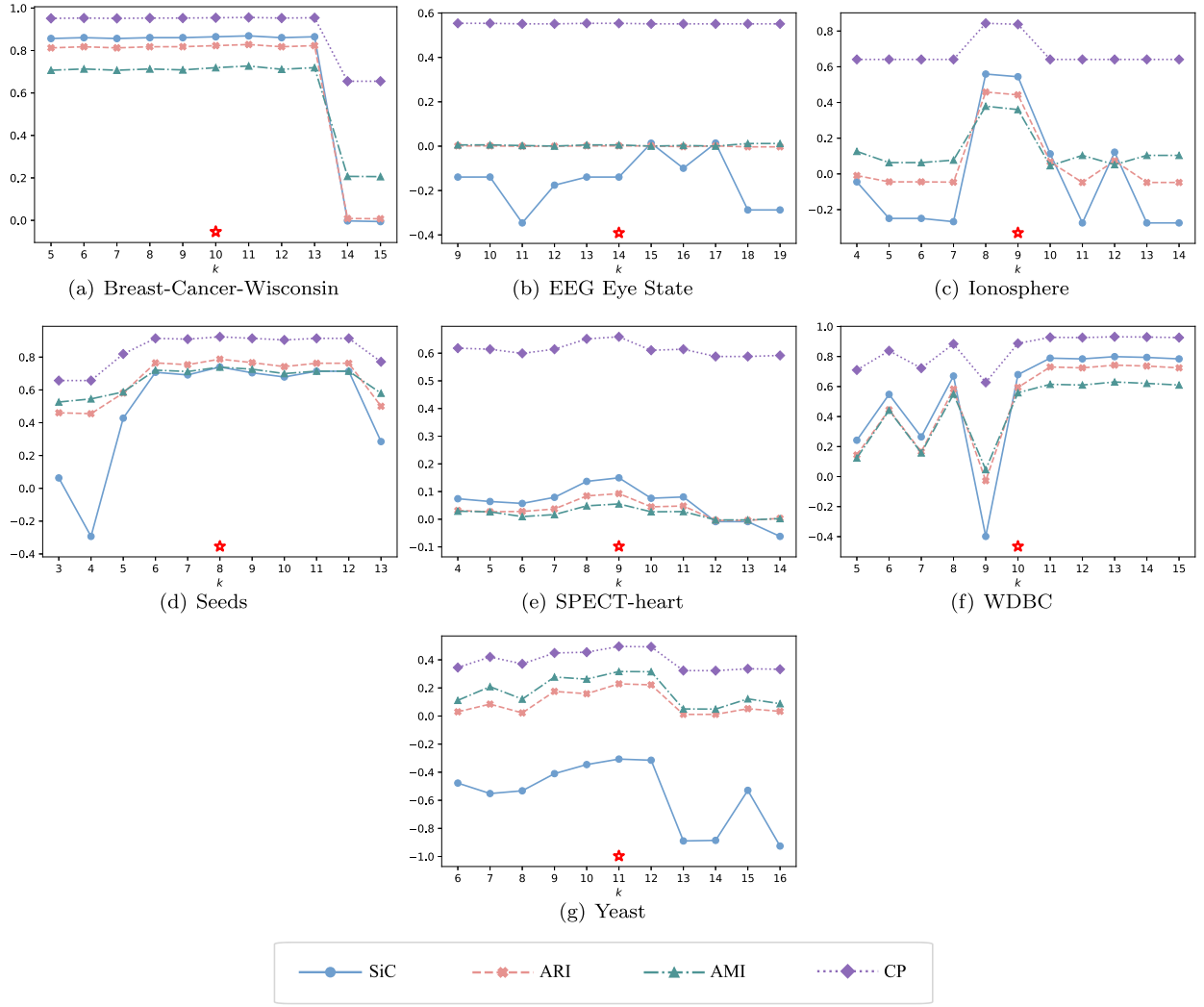


Fig. 13. Analysis of the parameter k.

As can be seen in Table 15, GFDC produces optimal clustering results on the SPECT-heart and Yeast datasets, and superior clustering results on the Breast-Cancer-Wisconsin, Ionosphere and Seeds datasets, but average clustering results on the EEG Eye State and WDBC datasets. In other words, GFDC can handle datasets with a larger number of clusters or larger dimensions better, but does not perform particularly well on datasets with large amounts of samples. Probably because, when the size of data is large and the sample space is sparse, a relatively small number of core-granules are generated and the basis for calculation of evidence in the process of evidential assignment is reduced, which in turn results in poor clustering results. Although some of the compared algorithms outperform GFDC on this type of dataset, the values of evaluation indexes for the clustering results obtained by all algorithms are low, that is, most of the existing algorithms perform poorly on this type of dataset. Moreover, considering the overall effectiveness, GFDC is indistinguishable from the state-of-the-art IM-DPC. In general, GFDC outperforms compared algorithms for clustering on small-sized or medium-sized, high-dimensional and cluster-laden datasets, and has a moderate performance for clustering on large-sized datasets.

4.4. Analysis of the parameter k

In GFDC, there is a key parameter k that affects the quality of clustering results, which is recommended to be taken as $\lceil log_2(n) \rceil$, where n is the number of samples in a dataset. In the following, different k will be taken for each real-world dataset to observe the impact of k on clustering results and demonstrate the advantages of the recommended value.

For each real-world dataset, 10 different values of k around the recommended value of k are taken for the experiments. The corresponding evaluations for the clustering results are shown in Fig. 13, where the location of the red hollow star on the horizontal axis represents the recommended value of k taken.

As shown in Fig. 13, when k is taken to the recommended value, the SiC, ARI, AMI and CP values of the clustering results are almost the highest. Most of the recommended values of k are located in the flat region where evaluation results are less variable, which indicates that it is reasonable to make the parameter k take the value $\lceil log_2(n) \rceil$ in GFDC.

5. Conclusions

This paper proposes a granule fusion framework for density-based clustering, which has advantages in handling clusters with arbitrary shapes and varying densities. The main advantages of the proposed algorithm GFDC include four aspects. First of all, the proposed sparse degree metric is capable of measuring both local and global densities of samples, which integrates the notion of optimal information granularity and k-nearest neighbors. In addition, the granulation process with sparse degrees of samples, which takes full account of different density regions in data, enables GFDC to handle data with large density differences among clusters. Furthermore, the emerging fusion strategies break the limitation brought by the convex structure of granules, which improves the performance of GFDC for detecting irregular clusters. Lastly, the improved evidential assignment method based on an initial clusters structure mitigates the probability of error propagation and identifies outliers. From extensive experimental results, the effectiveness and superiority of GFDC are demonstrated on both synthetic and real-world datasets. In the future, there are some directions for extending the proposed algorithm. The design of more fusion strategies and the extension to dynamic data are interesting research directions.

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.

Data availability

Data will be made available on request.

Acknowledgements

The authors would like to thank the editors and anonymous reviewers for their constructive comments. This work is supported by NSFC (No. 12231007), Hunan Provincial Natural Science Foundation of China (No. 2023JJ30113), Changsha Municipal Natural Science Foundation of China (No. kq2202138) and Guangdong Basic, Applied Basic Research Foundation (No. 2023A1515012342) and Hunan Provincial Innovation Foundation for Postgraduate (No. CX20210398).

References

[1] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: Berkeley Symposium on Mathematical Statistics and Probability, 1967, pp. 281–297.
[2] D. Arthur, S. Vassilvitskii, K-means++: the advantages of careful seeding, in: Proceedings of the 8th Annual ACM-SIAM Symposium on Discrete Algorithms, New Orleans, Louisiana, USA, January 7-9, 2007, pp. 1027–1035.
[3] T. Zhang, R. Ramakrishnan, M. Livny, BIRCH: an efficient data clustering method for very large databases, SIGMOD Rec. 25 (1996) 103–114.
[4] U. von Luxburg, A tutorial on spectral clustering, Stat. Comput. 17 (4) (2007) 395–416.
[5] M. Ester, H.-P. Kriegel, J. Sander, X.W. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, Portland, Oregon, USA, August 2-4, 1996, pp. 226–231.
[6] A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks, Science 344 (6191) (2014) 1492–1496.
[7] W. Wang, J. Yang, R. Muntz, STING: a statistical information grid approach to spatial data mining, in: Proceedings of the 23rd International Conference on Very Large Databases, San Francisco, California, USA, August 25-29, 1997, pp. 186–195.
[8] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. R. Stat. Soc., Ser. B, Methodol. 39 (1) (1977) 1–22.
[9] P.H. Huang, Y. Huang, W. Wang, L. Wang, Deep embedding network for clustering, in: Proceedings of the 22nd International Conference on Pattern Recognition, Stockholm, Sweden, August 24-28, 2014, pp. 1532–1537.
[10] C.-C. Hsu, C.-W. Lin, CNN-based joint clustering and representation learning with feature drift compensation for large-scale image data, IEEE Trans. Multimed. 20 (2) (2018) 421–429.
[11] M.J. Du, S.F. Ding, H.J. Jia, Study on density peaks clustering based on k-nearest neighbors and principal component analysis, Knowl.-Based Syst. 99 (2016) 135–145.
[12] J.Y. Xie, H.C. Gao, W.X. Xie, X.H. Liu, P.W. Grant, Robust clustering by detecting density peaks and assigning points based on fuzzy weighted k-nearest neighbors, Inf. Sci. 354 (2016) 19–40.
[13] Y.H. Liu, Z.M. Ma, F. Yu, Adaptive density peak clustering based on k-nearest neighbors with aggregating strategy, Knowl.-Based Syst. 133 (2017) 208–220.
[14] A. Lotfi, P. Moradi, H. Beigy, Density peaks clustering based on density backbone and fuzzy neighborhood, Pattern Recognit. 107 (2020) 107449.
[15] F. Fang, L. Qiu, S.F. Yuan, Adaptive core fusion-based density peak clustering for complex data with arbitrary shapes and densities, Pattern Recognit. 107 (2020) 107452.
[16] Z.G. Long, Y. Gao, H. Meng, Y.Q. Yao, T.R. Li, Clustering based on local density peaks and graph cut, Inf. Sci. 600 (2022) 263–286.
[17] D.D. Cheng, S.L. Zhang, J.L. Huang, Dense members of local cores-based density peaks clustering algorithm, Knowl.-Based Syst. 193 (2020) 105454.
[18] J. Hou, A.H. Zhang, N.M. Qi, Density peak clustering based on relative density relationship, Pattern Recognit. 108 (2020) 107554.
[19] J.Y. Sun, G.J. Liu, An improvement of density peaks clustering algorithm based on KNN and gravitation, in: Proceedings of the 4th International Conference on Intelligent Autonomous Systems, Wuhan, Hubei, China, May 14-16, 2021, pp. 234–239.
[20] A.P. Dempster, Upper and lower probabilities induced by a multivalued mapping, Ann. Math. Stat. 38 (2) (1967) 325–339.

[21] Z.G. Su, T. Denœux, BPEC: belief-peaks evidential clustering, IEEE Trans. Fuzzy Syst. 27 (1) (2019) 111–123.
[22] C.Y. Gong, Z.G. Su, P.H. Wang, Q. Wang, Cumulative belief peaks evidential k-nearest neighbor clustering, Knowl.-Based Syst. 200 (2020) 105982.
[23] C.Y. Gong, Z.G. Su, P.H. Wang, Q. Wang, An evidential clustering algorithm by finding belief-peaks and disjoint neighborhoods, Pattern Recognit. 113 (2021) 107751.
[24] L. Ni, W.J. Luo, W.J. Zhu, W.J. Liu, Clustering by finding prominent peaks in density space, Eng. Appl. Artif. Intell. 85 (2019) 727–739.
[25] H. Yu, L.Y. Chen, J.T. Yao, A three-way density peak clustering method based on evidence theory, Knowl.-Based Syst. 211 (2021) 106532.
[26] L.A. Zadeh, Fuzzy sets and information granularity, in: Advances in Fuzzy Set Theory and Applications, North-Holland Publishing Co., Amsterdam, 1979, pp. 3–18.
[27] L.A. Zadeh, Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic, Fuzzy Sets Syst. 90 (2) (1997) 111–127.
[28] Y.C. Tang, B. Jin, Y. Sun, Y.Q. Zhang, Granular support vector machines for medical binary classification problems, in: Proceedings of the 2004 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, La Jolla, California, USA, October 7-8, 2004, pp. 73–78.
[29] S.F. Ding, Y.Z. Han, J.Z. Yu, Y.X. Gu, A fast fuzzy support vector machine based on information granulation, Neural Comput. Appl. 23 (2013) 139–144.
[30] S.Y. Xia, Y.S. Liu, X. Ding, G.Y. Wang, H. Yu, Y.G. Luo, Granular ball computing classifiers for efficient, scalable and robust learning, Inf. Sci. 483 (2019) 136–152.
[31] S.Y. Xia, D.W. Peng, D.Y. Meng, C.Q. Zhang, G.Y. Wang, E. Giem, W. Wei, Z.Z. Chen, Ball k-means: fast adaptive clustering with no bounds, IEEE Trans. Pattern Anal. Mach. Intell. 44 (1) (2022) 87–99.
[32] S.E. Toulmin, The Uses of Argument, Cambridge University Press, Cambridge, 1958.
[33] J.B. Yang, D.L. Xu, On the evidential reasoning algorithm for multiple attribute decision analysis under uncertainty, IEEE Trans. Syst. Man Cybern., Part A, Syst. Hum. 32 (3) (2002) 289–304.
[34] F.J. Li, Y.H. Qian, J.T. Wang, J.Y. Liang, Multigranulation information fusion: a Dempster-Shafer evidence theory-based clustering ensemble method, Inf. Sci. 378 (2017) 389–409.
[35] C.F. Lian, S. Ruan, T. Denœux, An evidential classifier based on feature selection and two-step classification strategy, Pattern Recognit. 48 (7) (2015) 2318–2327.
[36] Z. Wang, R.X. Wang, J.M. Gao, Z.Y. Gao, Y.J. Liang, Fault recognition using an ensemble classifier based on Dempster–Shafer theory, Pattern Recognit. 99 (2020) 107079.
[37] C.F. Lian, S. Ruan, T. Denœux, H. Li, P. Vera, Spatial evidential clustering with adaptive distance metric for tumor segmentation in FDG-PET images, IEEE Trans. Biomed. Eng. 65 (1) (2018) 21–30.
[38] L. Dymova, K. Kaczmarek, P. Sevastjanov, An extension of rule base evidential reasoning in the interval-valued intuitionistic fuzzy setting applied to the type 2 diabetes diagnostic, Expert Syst. Appl. 201 (2022) 117100.
[39] T. Denœux, M. Masson, EVCLUS: evidential clustering of proximity data, IEEE Trans. Syst. Man Cybern., Part B, Cybern. 34 (1) (2004) 95–109.
[40] M. Masson, T. Denœux, ECM: an evidential version of the fuzzy c-means algorithm, Pattern Recognit. 41 (4) (2008) 1384–1397.
[41] T. Denœux, O. Kanjanatarakul, S. Sriboonchitta, EK-NNclus: a clustering procedure based on the evidential k-nearest neighbor rule, Knowl.-Based Syst. 88 (2015) 57–69.
[42] K. Zhou, A. Martin, Q. Pan, Z.G. Liu, ECMdd: evidential c-medoids clustering with multiple prototypes, Pattern Recognit. 60 (2016) 239–257.
[43] Z.W. Zhang, Z. Liu, A. Martin, Z.G. Liu, K. Zhou, Dynamic evidential clustering algorithm, Knowl.-Based Syst. 213 (2021) 106643.
[44] Y.H. Qian, H. Zhang, F.J. Li, Q.H. Hu, J.Y. Liang, Set-based granular computing: a lattice model, Int. J. Approx. Reason. 55 (3) (2014) 834–852.
[45] G. Shafer, A Mathematical Theory of Evidence, Princeton University Press, Princeton, NJ, 1976.
[46] P. Smets, R. Kennes, The transferable belief model, Artif. Intell. 66 (2) (1994) 191–234.
[47] P. Smets, The combination of evidence in the transferable belief model, IEEE Trans. Pattern Anal. Mach. Intell. 12 (5) (1990) 447–458.
[48] Y.W. Chen, S.Y. Tang, S.W. Pei, C. Wang, J.X. Du, N.X. Xiong, DHeat: a density heat-based algorithm for clustering with effective radius, IEEE Trans. Syst. Man Cybern. Syst. 48 (4) (2018) 649–660.
[49] T. Denœux, A k-nearest neighbor classification rule based on Dempster-Shafer theory, IEEE Trans. Syst. Man Cybern. 25 (5) (1995) 804–813.
[50] A. Bryant, K. Cios, RNN-DBSCAN: a density-based clustering algorithm using reverse nearest neighbor density estimates, IEEE Trans. Knowl. Data Eng. 30 (6) (2018) 1109–1121.
[51] C. Sun, M.J. Du, J.R. Sun, K.K. Li, Y.Q. Dong, A three-way clustering method based on improved density peaks algorithm and boundary detection graph, Int. J. Approx. Reason. 153 (2023) 239–257.
[52] P.J. Rousseeuw, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, J. Comput. Appl. Math. 20 (1987) 53–65.
[53] L. Hubert, P. Arabie, Comparing partitions, J. Classif. 2 (1) (1985) 193–218.
[54] N.X. Vinh, J. Epps, J. Bailey, Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance, J. Mach. Learn. Res. 11 (2010) 2837–2854.
[55] E. Rendón, I. Abundez, A. Arizmendi, E.M. Quiroz, Internal versus external cluster validation indexes, Comput. Sci. 5 (1) (2011) 27–34.
[56] M. Parmar, D. Wang, X.F. Zhang, A.H. Tan, C.Y. Miao, J.H. Jiang, Y. Zhou, REDPC: a residual error-based density peak clustering algorithm, Neurocomputing 348 (2019) 82–96.