DOI: 10.22034/tjee.2022.15642

Adaptive Gaussian Density Distance for Clustering

Mahdi Yazdian-Dehkordi[1,*], Farzane Nadi[2], Solmaz Abbasi[3]
Department of Computer Engineering, Yazd University, Yazd, Iran.
[1]yazdian@yazd.ac.ir, [2]farzane.nadi@stu.yazd.ac.ir, [3]soulmaz.abbasi@stu.yazd.ac.ir
[*] Corresponding author

Received: 01/08/2022, Revised: 10/15/2022, Accepted:. 11/20/2022

Abstract

Distance-based clustering methods categorize samples by optimizing a global criterion, finding ellipsoid clusters with roughly equal sizes. In contrast, density-based clustering techniques form clusters with arbitrary shapes and sizes by optimizing a local criterion. Most of these methods have several hyper-parameters, and their performance is highly dependent on the hyper-parameter setup. Recently, a Gaussian Density Distance (GDD) approach was proposed to optimize local criteria in terms of distance and density properties of samples. GDD can find clusters with different shapes and sizes without any free parameters. However, it may fail to discover the appropriate clusters due to the interfering of clustered samples in estimating the density and distance properties of remaining unclustered samples. Here, we introduce Adaptive GDD (AGDD), which eliminates the inappropriate effect of clustered samples by adaptively updating the parameters during clustering. It is stable and can identify clusters with various shapes, sizes, and densities without adding extra parameters. The distance metrics calculating the dissimilarity between samples can affect the clustering performance. The effect of different distance measurements is also analyzed on the method. The experimental results conducted on several well-known datasets show the effectiveness of the proposed AGDD method compared to the other well-known clustering methods.

Keywords

Density-based Clustering, Distance-based Clustering, Gaussian Density

1. Introduction

Clustering can be considered the most important unsupervised learning problem. In clustering, several objects (data points) are grouped such that data points within each cluster are similar to each other while data points from different clusters are dissimilar. Clustering is used in many areas, including machine learning, pattern recognition, image analysis, information retrieval, bioinformatics, data compression, computer graphics, web pages, and robotics [1]-[4].

There are many well-known clustering methods. Hard and soft clustering are the two main groups in clustering approaches [5]. Soft or Fuzzy clustering [6], [7] is a form of clustering in which each data point can belong to more than one cluster, but in hard clustering, each data point can only belong to one cluster.

Generally, hard clustering can be categorized into hierarchical clustering and partitioning clustering [8], [9]. In hierarchical clustering methods, data points are categorized into hierarchical tree structures called dendrograms. All the data points are placed in the dendrogram's root, and each leaf node is a record, and the middle nodes determine the similarity of records to each other.

Partitioning clustering decomposes a data set into multiple groups based on their similarity through an iterative process. Two main categories of partitioning methods are distance-based and density-based methods which follow different theoretical intuitions to categorize data into clusters. Distance-based clustering methods such as k-means, k-medoids, and fuzzy c-means [10], [11] optimize global criteria based on the distance between samples and cluster centroids. These methods need the number of clusters as prior knowledge while not available in many real applications [12]. Besides, they are not repeatable, i.e., using different initial cluster centers, they produce different clustering results. Another important issue for distance-based approaches is that they tend to create ellipsoid clusters with roughly equal sizes, and they are not appropriate for finding non-convex clusters.

The theoretical intuition behind density-based clustering methods, such as DBSCAN[1] [13] and OPTICS [14] is that they optimize local criteria according to the density distribution of patterns to find clusters with arbitrary shapes and sizes. For example, the DBSCAN method connects nearby neighbors to form clusters. This algorithm takes input parameters, Ɛ-neighborhood radios, and min-points (minimum number of data points in neighborhood radios) in defining the core object. If a data point is within reach of Ɛ, and the connected data points are more than min-points then the area is clustered. However, DBSCAN doesn't work well for clusters with varying density rates and high-dimension data. DVBSCAN handels density variation within clusters, but it's parameters cannot be determined automatically [15]. OPTICS extends DBSCAN to produce a cluster ordering obtained from a wide range of parameter settings. This algorithm performs

[1] Density-based spatial clustering of applications with noise

DOI: 10.22034/tjee.2022.15642

clustering without considering a fixed value for parameter Ɛ.

The previously mentioned methods have some free parameters which the user should determine. These parameters significantly affect the clustering results. Additionally, they defined a density factor for detecting noisy points. Smiti et al. [16] proposed DBSCAN-GM[2] to automatically set the parameters of the DBSCAN method. It utilizes Gaussian Means to find Ɛ and min-points in a cluster. Like the DBSCAN algorithm, it assumes similar densities for the clusters and degrades when the densities of clusters are different. Gaussian density is also used in Gaussian Mixture Model (GMM) for clustering [17], [18]. This method forms each cluster using a Gaussian element whose mean and covariance are estimated through Expectation Maximization (EM) algorithm. The number of Gaussians can be determined either by the user or automatically. The computational complexity of these algorithms is high. Besides, in some cases, the shape of the data does not follow the Gaussian model. Varsha et al. [19] proposed ADBSCAN (Adaptive DBSCAN) that uses techniques such as grid search and Gaussian kernel to search optimized values for the threshold density of clusters. It's a free parameter clustering method but, due to using grid search algorithm, it has high time complexity.

Güngör et al. [20] have proposed a new free parameter approach for clustering based on Gaussian Density Distance (GDD). The main contribution of this work is its use of GMM in creating Gaussian values to represent data points. Standard deviation was used to create cluster regions. The densest point was defined as the cluster centroid. This clustering algorithm is parameter-free and can calculate all necessary parameters based on the dataset. The main issue of GDD is that this method involves all samples (clustered and unclustered samples) to form each cluster. However, it is clear that when a cluster is formed, the samples in the cluster must not be interfered with in creating the next cluster. It can be ignored if the clustered samples have little effect on forming the next clusters, but this issue might be negligible. But in some situations, it can lead to inappropriate selection of the next cluster centroid.

One of the drawbacks of GDD is that it cannot perform well for datasets with sudden density variation. In GDD, since all parameters are calculated once. The clustered samples may undesirably interfere in creating new clusters. In this paper, a modification of the GDD algorithm (called Adaptive GDD) is proposed by adaptively calculating the parameters during the clustering process. The GDD and the proposed AGDD methods optimize local criteria based on both the distance and density properties of samples. In the GDD method, these properties are inappropriately affected by the clustered samples. However, in AGDD, after grouping similar data points into one cluster, the clustered samples are ignored, and both density and distance properties are updated before creating the next cluster. In this study, a toy problem is designed to show the drawback of the GDD method and visualize how the suggested AGDD algorithm can rectify this defect.

Various distance measurement methods are available to calculate the distance between the samples, such as Euclidean distance, Manhattan distance, Chord distance, Cosine similarity, Czekanowski Coefficient, and Mean Character Difference distance [21], [22]. Depending on the distance measurement method, the clustering results may be changed, which can affect the performance of the method. Here, the effect of different distance measurements has been analyzed in terms of the clustering output of the method. The experimental results are provided and discussed on different well-known datasets to study the effectiveness of the proposed AGDD method. The results show that the AGDD method could improve the clustering performance and decrease the number of samples used to calculate parameters employed in creating a cluster.

In sum, the main contributions of the paper can be enumerated as follows:

- For intelligent unsupervised clustering, we propose an adaptive version of the GDD method, which offers stability over different trials of the algorithms and the ability to identify clusters with different shapes and connectivity rates with no hyper-parameter tuning. We have improved the GDD method by eliminating the inappropriate effect of clustered samples in grouping remaining unclustered samples, which is well described in a toy problem.
- The effect of various distance measurements has been studied for the proposed method. Furthermore, the performance of the method has been analyzed and discussed using several indices on different well-known datasets.

The rest of the paper is organized as follows: Section 2 provides the background of the paper. Section 3 represents the GDD issues, explains the proposed AGDD method, and finally describes the behavior of both methods via a toy problem. Section 4 presents our experimental results and discussion, followed by a conclusion in Section 5.

2. GDD Clustering Method

The Gaussian Density Distance (GDD) method [20] has no hyper-parameter and performs clustering based on the distance of samples as well as the density of samples calculated by Gaussian kernel. In this way, it calculates the following parameters:

- Gaussian Matrix ($GM_{n\times n}$): Gaussian Matrix is a non-negative and symmetrical $n \times n$ matrix when n shows the number of samples. In this matrix, $\mathrm{GM}_{\mathrm{i,j}}$ is the Gaussian density of i-th sample in respect to j-th sample [20].
- Distance Matrix ($DM_{n\times n}$): It is a non-negative and symmetrical $\mathrm{n} \times \mathrm{n}$ matrix. In this matrix, $\mathrm{DM}_{\mathrm{i,j}}$ is the Euclidean distance between i-th sample and j-th sample [20].

The GDD method calculates GM and DM using all samples. For i-th sample the summation of its GM over

[2] Density-Based Spatial Clustering of Applications with Noise-Gaussian Means

all samples, i.e., $\sum_{j=1.i\neq j}^{n} \mathrm{GM}_{i.j}$ is utilized to measure how a sample is dense. At each step, the GDD method selects the densest sample as the centroid of a new cluster and forms the cluster C^k with this sample. Then, an unclustered sample x_j is added to the cluster C^k (with x_t as its centroid) if the two following conditions are satisfied:

$$\exists x_j \in C^k \quad where \quad \begin{cases} GM_{t.j} \geq FGDT - GGDT \\ DM_{t.j} \leq FDT + GDT \end{cases} \qquad (1)$$

where these parameters defined as FDT: “Fixed Distance Threshold”, and FGDT: “Fixed Gaussian Density Threshold” which are calculated based on the cluster centroid [20]. Besides, GDT: “Gradient Distance Threshold” and GGDT: “Gradient Gaussian Density Threshold” express changes in variance [20]. The calculation of these parameters needs many details, which is out of our focus here. Therefore, to avoid confusion, we do not indicate the formulation of these parameters. The readers are referred to [20] for more details on GDD calculations.

3. Proposed Approach

The GDD method is one of the nonparametric methods for clustering samples based on Gaussian kernel and density. This algorithm is not sensitive to the initial point i.e., it leads to the same results if a run is reapeted [20]. However, several issues degrade the performance of this method in certain situations. Here, the GDD issues are discussed in section 3-1; then, an improvement on the GDD, named AGDD (Adaptive GDD), is proposed in section 3-2. Finally, to have a better clarification of the proposed approach, GDD and AGDD methods are compared in terms of effectiveness in section 3-3.

3.1. GDD Issues

The main issue of GDD is that this method involves all samples (clustered and unclustered samples) to form each cluster. However, it is clear that when a cluster is formed, the samples in the cluster must not interfere in creating the next cluster. It can be ignored if the clustered samples have little effect on forming the next clusters, but in some situations, it can lead to inappropriate selection of the next cluster centroid, i.e., the most densely unclustered sample. This side effect is shown in Fig. 1. It can be seen that the cluster centroids (red points) are not located at the center of Clusters #2 and #3.

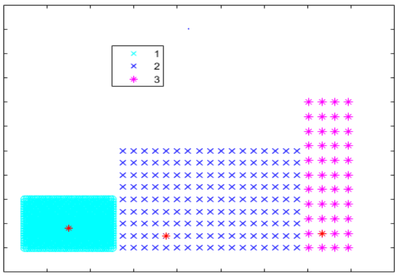


Fig. 1. Inappropriate selection of most dense cluster sample due to interference of clustered samples

In many applications, cluster centroids have a key role in decision-making. Employing an unsuitable cluster center might lead to incorrect estimation propagated to the next clustering steps. For example, two clusters are possible to be incorrectly formed in one cluster; and/or, one cluster is possible to be incorrectly shown by two disjoint clusters. It also might corrupt some calculations, such as the FDT calculation of the GDD method in equation (1). A toy problem is provided and discussed in Section 3-3 to illustrate these side effects of the GDD method.

3.2. Proposed Adaptive GDD (AGDD)

Let $\mathrm{X} = \{\mathrm{X}_1.\mathrm{X}_2.\dots.\mathrm{X}_n\}$ be an input dataset, and n is the number of samples. Each of which is $\mathrm{X}_i = \{x_{i1}.x_{i2}.\dots.x_{id}\}$, the i-th sample of the dataset, and d is the number of sample attributes. If C^k shows the samples of k-th cluster, then the unclustered samples would be:

$$X^k = X^{k-1} \setminus C^k \qquad (2)$$

where $X^0 = X$ *and* $C^0 = \emptyset$. The operator $A\setminus B$ means removing the samples of B from A. In the first step, all samples are unclustered so X^0 and C^0 are equal to X and $\emptyset$ (empty set), respectively. Let GM^k show the Gaussian Matrix after forming k-th cluster. In equation (3), the GDD method employed a static value at each time step, i.e., $\mathrm{GM}^0 = \mathrm{GM}^1 = \cdots = \mathrm{GM}^{\#\mathrm{cluster}}$. Here, at time step k (when k-th cluster is formed), the GM between i-th and j-th samples is adaptively calculated as:

$$GM_{i.j}^k = \exp\left(-\sum_{m=1}^{d} \frac{(x_{im}-x_{jm})^2}{2c^2}\right) \qquad (3)$$
$$where\ X_i.X_j \in X^k\ and\ c = \sqrt{\frac{\mu_m \sigma_m}{2\pi}}$$

where x_{im} is m-th attribute of i-th sample in X^k. In this equation, c illustrates the inter-connection coefficient among samples which explains how neighboring samples are scattered. In calculating c, μ_m and σ_m are the mean and deviation of m-th attribute of all samples, respectively.

Suppose GPM_i^k illustrates the Gaussian Point Mean of i-th sample at step k, which is used to select the densest sample as the centroid of the new cluster. Here, instead of using static value [20], the GMP is adaptively calculated at step k using:

$$GPM_i^k = \frac{1}{|X^k| - 1} \sum_{X_i.X_j \in X^k.j\neq i} GM_{i.j}^k \qquad (4)$$

Analogously, the Distance Matrix between i-th and j-th samples at step k, $\mathrm{DM}_{i.j}^k$, is computed using:

$$DM_{i.j}^k = dis(X_i.X_j)\ where\ X_i.X_j \in X^k \qquad (5)$$

where dist(.) shows the distance function [21]. The parameters mentioned above are utilized to calculate the other GDD method variables (see [20] for more details). Using the proposed adaptive calculation of the parameters, apart from the GDD method, the unclustered samples cannot have negative side effects on the clustering process. Fig. 2 depicts the overview of the proposed Adaptive GDD method (AGDD).

Initially, at the pre-processing step, the samples with at least one missing attribute, redundant samples, and the attributes with zero variance (which have no useful information) are removed to reduce the computational cost. In the clustering process, the parameters GM,

DOI: 10.22034/tjee.2022.15642

GPM, and DM are calculated based on the unclustered samples.

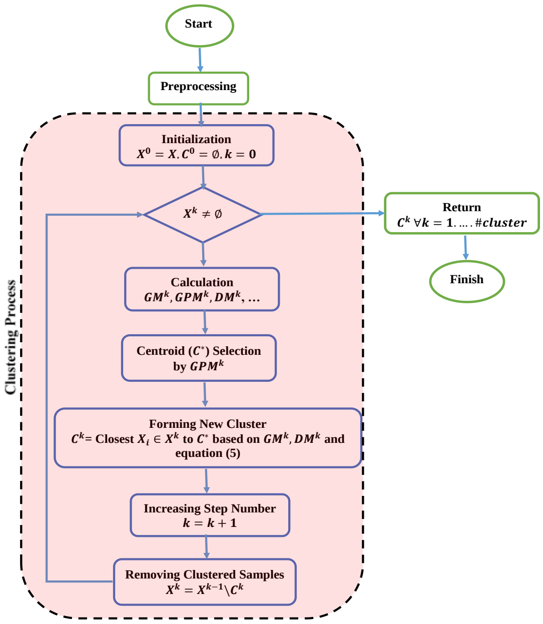

Start
Preprocessing
Initialization
$X^0 = X. C^0 = \emptyset. k = 0$
$X^k \neq \emptyset$
Return
$C^k\ \forall k = 1. \dots .\#cluster$
Finish
Calculation
$GM^k, GPM^k, DM^k, \dots$
Centroid (C^*) Selection by GPM^k
Forming New Cluster
C^k= Closest $X_i \in X^k$ to C^* based on GM^k, DM^k and equation (5)
Increasing Step Number
$k = k + 1$
Removing Clustered Samples
$X^k = X^{k-1} \backslash C^k$
Clustering Process

Fig. 2. Overview of the proposed AGDD method

At each iteration, the densest sample is selected as the centroid of the new cluster. Afterward, each unclustered sample $\mathrm{x_j}$ is added to this cluster (with $\mathrm{x_t}$ as its centroid) if:

$$\exists x_j \in C^k \quad where \quad \begin{cases} GM_{t.j}^k \geq FGDT - GGDT \\ DM_{t.j}^k \leq FDT + GDT \end{cases} \tag{6}$$

where $\mathrm{GM_{t.j}^k}$ and $\mathrm{DM_{t.j}^k}$ are calculated using (3) and (5), respectively, and the other parameters are defined as explained in Section 2. Then, the clustered samples are removed from the sample set, and this process is repeated until all samples are clustered.

3.3. Toy problem

To have a better clarification, in this section, the behavior of the GDD method versus the proposed AGDD approach is explained via a toy problem with 3 clusters. Consider the samples of three clusters shown in Fig. 3.

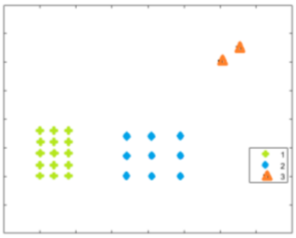


Fig. 3. A toy problem to show the drawback of the GDD method

At first, GDD is used for clustering these samples. Fig. 4 represents the clustering output and the GPM of each sample at each step. Recall that GPM is used to find the densest sample (shown with red background in GPMs of Fig. 4) as the centroid of a new cluster. Clearly, using static calculation of the GDD method, GPM is fixed at all steps. The cluster centroids are illustrated with red stars in the clustering outputs shown in the first row of Fig. 4. It is clear that due to the effect of clustered samples, the centroids are selected inappropriately, and five clusters are formed incorrectly.

The results of the proposed AGDD method are illustrated in Fig. 5. As shown, at every step, the GMP is updated using only unclustered samples. The clustered samples are shown with dashed lines in GMP vectors. Obviously, the AGDD method can successfully select appropriate cluster centroids and form three clusters. It should be noted that despite the GDD, the cluster centroids found with AGDD are located at the center of their clusters.

4. Experimental result

In this section, the efficiency of the proposed method is evaluated on different distance measurements. Besides, the performance of the method is compared with several well-known methods using different datasets.

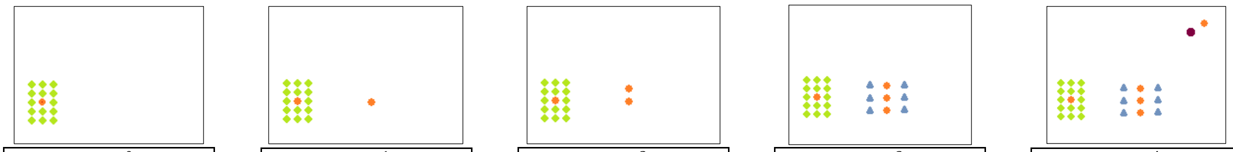
4.1. Data Description

The efficiency of the proposed AGDD and other widely used methods are compared using some benchmark datasets. These datasets are illustrated in Table 1. As shown, datasets with varied dimensions, clusters, and instances are employed.

Table 1. Data Description [23]–[25]

Dataset Name	#Dimension	#Cluster	#Instance
Jain	2	2	373
R15	2	15	600
Spiral	2	3	312
Aggregation	2	7	788
Compound	2	6	399
DiffDense	2	6	139
New_thyroid	5	3	215
Breast-canser-wisconsin	9	2	263
Iris	4	3	150
Zoo	16	7	101

DOI: 10.22034/tjee.2022.15642



GPM^0	GPM^1	GPM^2	GPM^3	GPM^4
0.0235	0.0235	0.0235	0.0235	0.0235
0.0293	0.0293	0.0293	0.0293	0.0293
0.0293	0.0293	0.0293	0.0293	0.0293
0.0293	0.0293	0.0293	0.0293	0.0293
0.0235	0.0235	0.0235	0.0235	0.0235
0.0404	0.0404	0.0404	0.0404	0.0404
0.0477	0.0477	0.0477	0.0477	0.0477
0.0478	0	0	0	0
0.0477	0.0477	0.0477	0.0477	0.0477
0.0404	0.0404	0.0404	0.0404	0.0404
0.0235	0.0235	0.0235	0.0235	0.0235
0.0293	0.0293	0.0293	0.0293	0.0293
0.0293	0.0293	0.0293	0.0293	0.0293
0.0293	0.0293	0.0293	0.0293	0.0293
0.0235	0.0235	0.0235	0.0235	0.0235
0.0027	0.0027	0.0027	0.0027	0.0027
0.0028	0.0028	0.0028	0.0028	0.0028
0.0027	0.0027	0.0027	0.0027	0.0027
0.00388	0.00388	0.00388	**0.00388**	0
0.0039	**0.0039**	0	0	0
0.00388	0.00388	**0.00388**	0	0
0.0012	0.0012	0.0012	0.0012	0.0012
0.0012	0.0012	0.0012	0.0012	0.0012
0.0012	0.0012	0.0012	0.0012	0.0012
0.0002	0.0002	0.0002	0.0002	**0.0002**
0.0002	0.0002	0.0002	0.0002	0.0002
Step 0	Step 1	Step 2	Step 3	Step 4

Fig. 4. Clustered samples and corresponding value of GPM vector in every step of GDD. At each step, the densest sample (determined by red color) is selected as the centroid of a new cluster and the unclustered samples, which satisfy equation (1), are added to this new cluster.

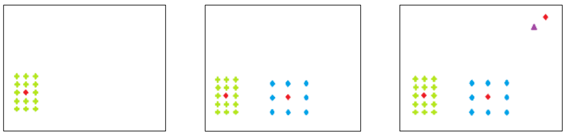
Here, several datasets with different shapes (convex and non-convex shaped), densities (varying within and between clusters), sizes, and connectivity rates are utilized to study the performance of the method. The datasets (except DiffDense) are ones that the GDD approach has been applied to them.

4.2. Evaluation metrics

Clustering analysis can be performed using either internal or external validity indices. Internal validity indices measure the compactness and the degree of separation between clusters; whereas, external validity indices measure the degree of agreement between the estimated clustering results and the ground truth partitions. In the present study, in addition to demonstrating the results, we validate the results using validity measures including MPSR[1] and NMI[2] [26]–[37]. MPSR has been utilized in the GDD paper, and NMI is added to better evaluation of the results. These metrics are calculated as below:

- MPSR measures the prediction error between the predicted clustering label and true target label calculated as:

$$MPSR = \frac{\#mispredicted\ samples}{\#total\ samples} \qquad (7)$$



GPM^0	GPM^1	GPM^2
0.0235	-	-
0.0293	-	-
0.0293	-	-
0.0293	-	-
0.0235	-	-
0.0404	-	-
0.0477	-	-
0.0478	-	-
0.0477	-	-
0.0404	-	-
0.0235	-	-
0.0293	-	-
0.0293	-	-
0.0293	-	-
0.0235	-	-
0.0027	0.0086	-
0.0028	0.0162	-
0.0027	0.0086	-
0.00388	0.0089	-
0.0039	**0.0165**	-
0.00388	0.0089	-
0.0012	0.0078	-
0.0012	0.0153	-
0.0012	0.0078	-
0.0002	0.0026	**4.4e-16**
0.0002	0.0026	4.4e-16
Step 0	Step 1	Step 2

Fig. 5: Clustered samples and corresponding value of GPM vector in every step of the proposed AGDD. In contrast to the GDD method, the clustered samples are removed from the sample set at each step.

[1] Missed Predicted Sample Rate

[2] Normalized Mutual Information

DOI: 10.22034/tjee.2022.15642

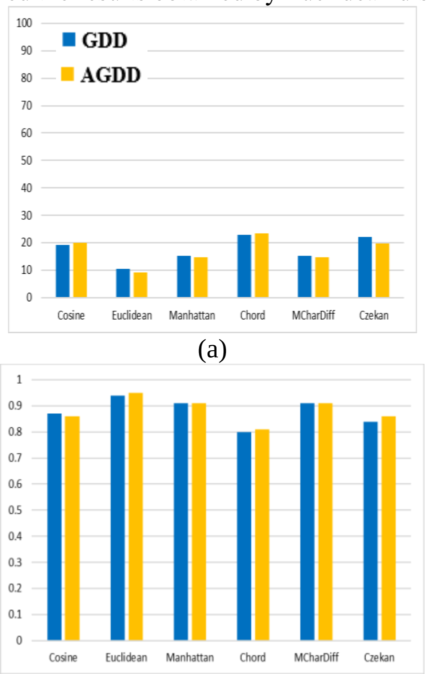
- NMI is a measurement based on mutual information for comparing disjoint partitions and measuring diversity among different clusters which is calculated as [29], [38]:

$$NMI(\Omega.C)=\frac{I(\Omega;C)}{[H(\Omega)+H(C)]/2} \qquad (8)$$

where **I** and **H** are mutual information and entropy, respectively.

4.3. Evaluation through different distance metrics

Depending on the distance measurement method [21], the clustering results can change, which may affect the performance of the method. Here, the effect of different distance measurements including Euclidean distance, Manhattan distance, Chord distance, Cosine similarity, Czekanowski Coefficient, and Mean Character Difference distance [21] have been analyzed on the clustering output of the GDD and AGDD methods. Fig. 6 depicts the mean value of the indices calculated on all datasets described in Table 1 for both GDD and AGDD methods. As Fig. 6 demonstrates, it is evident that the AGDD method performs better than the GDD method according to all distance measurements. Besides, the results show that the Euclidian distance achieves a better performance than the other distance measurements. Therefore, in the rest of this paper, we have only considered the results obtained by Euclidean distance.



(a)

(b)

Fig. 6. Results of different distance metrics in the GDD and AGDD algorithms. (a) MPSR, (b) NMI

4.4. Evaluation on simulated datasets

The efficiency of GDD and the proposed AGDD methods are compared using some benchmarks [20], depicted in the first column of Fig. 7. Visual results of applying both GDD and AGDD algorithms are shown in the second and third columns of Fig. 7, respectively. Like the toy problem illustrated in Section 3.3, it can be seen that in Aggregation and DiffDense datasets, the GDD method fails to discover clusters with sudden density variations and generates several single-sample clusters [26].

To have a more precise comparison between GDD and AGDD on these datasets, we have reported the following criteria for both methods on the simulated datasets:

- ESC (Error of Single Cluster): this criterion shows the number of clusters with only one or two samples.
- ENC (Error of Number of Clusters): The difference between the number of clusters in the input samples and the number of clusters outputted clustering method.
- ECC (Error of Centroid Cluster): if centroids of all clusters are located correctly after applying the clustering method, this method does not have ECC.
- USR (Used Sample Rate): sample rate used in forming clusters is calculated as:

$$USR=\frac{\sum_{c_j}\#\,samples\ involved\ in\ forming\ cluster\ c_j}{\#cluster\times\#sample} \qquad (11)$$

The first three criteria are shown in Table 2, and the later one is depicted in Fig. 8.

Table 2. Results of GDD and AGDD for ESC, ENC and ECC criteria

Method / *Datasets*	*AGDD*			*GDD*		
	ESC	*ENC*	*ECC*	*ESC*	*ENC*	*ECC*
Jain	*0*	*0*	✓	*0*	*0*	✓
R15	*0*	*5*	✕	*0*	*6*	✕
Spiral	*0*	*0*	✓	*0*	*0*	✓
Aggregation	*0*	*2*	✕	*8*	*10*	✓
Compound	*0*	*1*	✓	*0*	*1*	✓
DiffDense	*0*	*0*	✕	*1*	*1*	✓

The results show that:

- For Jain, Spiral, and Compound datasets, the results of GDD and AGDD methods are the same while, according to Fig. 8, it can be seen that averaged used sample rate (USR) of the AGDD method is less than GDD in all datasets.
- According to Fig. 7, in Aggregation dataset, both GDD and AGDD methods merge Cluster#6 and cluster#3. This is because when two neighboring clusters have a connection (even a thin connection), the GDD and AGDD methods which utilize distance as well as density property of samples, link the clusters together.

DOI: 10.22034/tjee.2022.15642

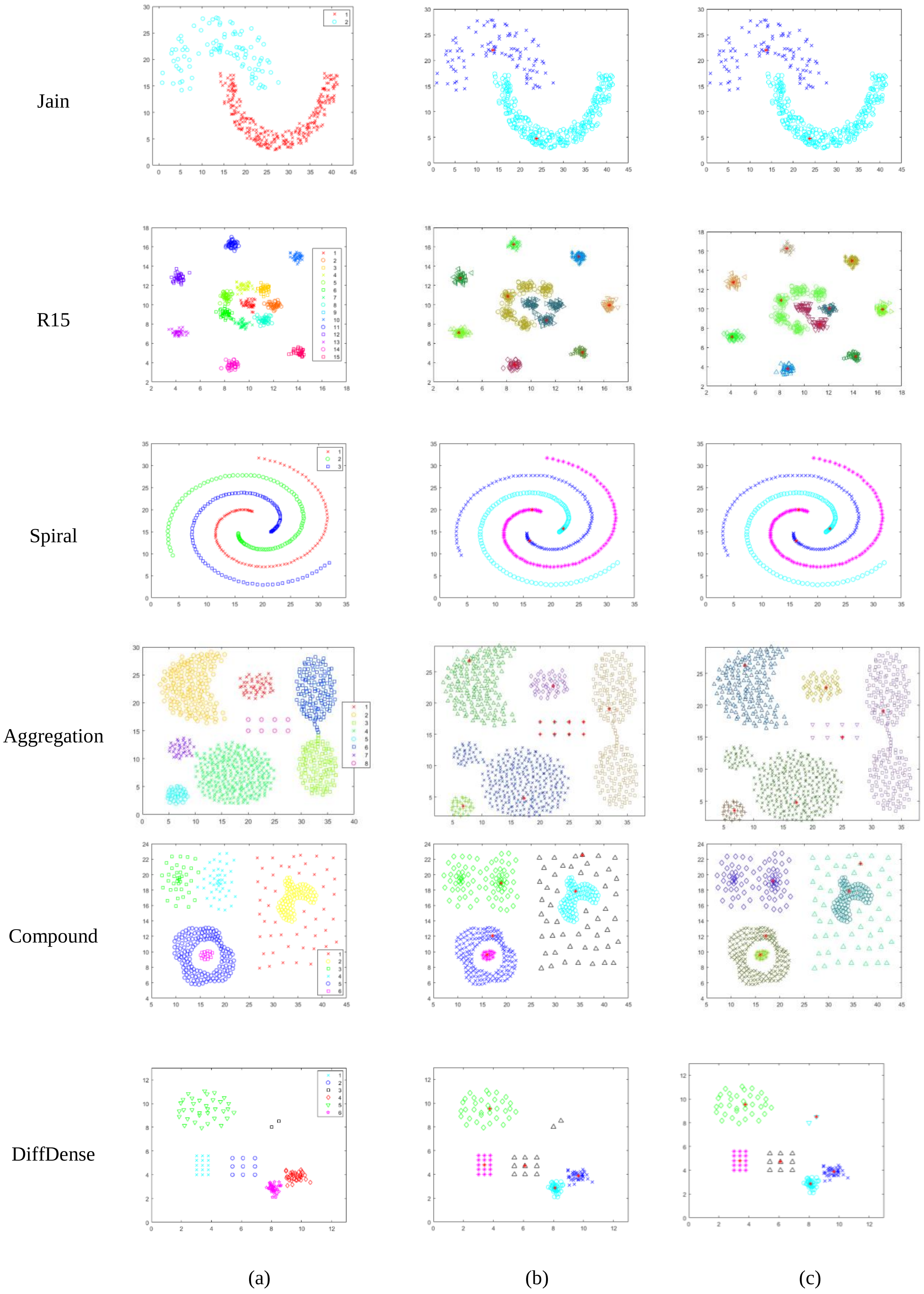


Fig. 7. Original simulated datasets and the results of GDD and the proposed AGDD methods. (a) Original Data, (b) GDD Result, (c) AGDD Result.

DOI: 10.22034/tjee.2022.15642

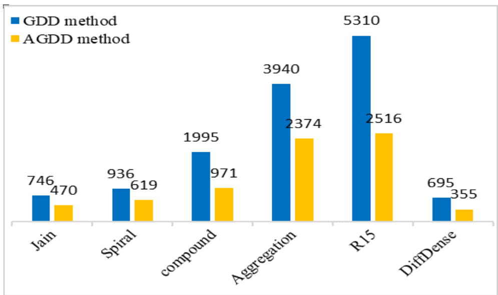


Fig. 8. Used Sample Rate (USR) for GDD and AGDD

- The Error of single clusters (ESC) and the Error of Number of Clusters (ENC) show that the GDD method fails to discover the clusters with sudden density change and therefore produces several single-sample clusters. However, as shown in the toy problem (Section 3-3), the AGDD method improves the clustering output by eliminating the effect of samples in other clusters.

To have a better evaluation, the result of the proposed AGDD is compared to the results of the GDD and the well-known methods, including k-means, DBSCAN, and OPTICS clustering methods in Table 3. In this table, at each row, the first best result is shown by the underlined red number, and the blue number illustrates the second-best result. For parametric methods such as k-means and DBSCAN, it is a big challenge to select parameters to achieve the best performance. Here, min-points, which are the parameters of DBSCAN, are considered fixed and set to 4 as suggested by Güngör et al. [20]. The best value for another parameter of DBSCAN (Ɛ) and the only parameter of OPTICS (min-points) are chosen via cross-validation. For k-means clustering, the value of k is pre-determined based on the true number of clusters according to Table 1. Besides, since k-means results in different outputs at each run, the average validity indices are obtained over 100 runs.

Table 3. Results of the methods on simulated datasets in term of MPSR and NMI (Underlined red: first best result, Blue: second best result)

Dataset	Evaluation Metric	AGDD	GDD	DBSCAN	OPTICS	k-means
Jain	MPSR	0	0	13.14	86.06	21.45
	NMI	1	1	0.7	0.21	0.37
R15	MPSR	27.8	33.8	46.52	64.6	20.2
	NMI	0.86	0.84	0.74	0.69	0.91
Spiral	MPSR	0	0	4.17	62.18	65.06
	NMI	1	1	0.9	0.4	0
Aggregation	MPSR	17.08	17.96	18.22	8.79	22.24
	NMI	0.89	0.89	0.88	0.89	0.86
Compound	MPSR	9.52	9.52	15.79	69.42	21.55
	NMI	0.95	0.95	0.82	0.48	0.72
DiffDense	MPSR	0	1.43	28.78	64.03	18.12
	NMI	1	0.99	0.85	0.55	0.8

According to Table 3, it can be said that for Jain, Spiral, and Compound datasets, the results of GDD and AGDD methods yield better results rather than k-means, DBSCAN, and OPTICS methods. For all Aggregation, R15, and, DiffDense datasets, the proposed AGDD method performs better than GDD in all indices. In the Aggregation dataset, OPTICS can separate one of the linked clusters by selecting proper min-points (which are determined via exhaustive cross-validation) and achieve the best results; the proposed AGDD method reaches the best results without requiring any free parameters. In R15, k-means have the best results since this dataset has ellipsoid clusters with similar sizes. More importantly, this method already knows the actual number of clusters in advance. Here, the AGDD method achieves the second-best results without knowing the number of clusters as prior knowledge.

4.5. Evaluation on multi-dimensional datasets

In order to have a better analysis, the performance of the proposed method is also evaluated on real-world benchmark datasets with higher dimension. Table 4 shows the results of all methods using the evaluation metrics. The parameters for DBSCAN, OPTICS, and k-means are tuned like simulated datasets via cross-validation.

According to the results in Table 4, AGDD achieves the best results on New_thyroid dataset. In Breast-cancer-wisconsin, AGDD reaches the second-best results after k-means which knows the true number of clusters. The performance of the AGDD method degrades on Zoo datasets. Analysis of the results shows that it happened due to the curse of dimensionality. When the dimensionality increases, the volume of the sample space increases so fast and the available data points become sparse. In this condition, the density estimation employed in AGDD recursions is not promising; Therefore, the performance of the method degrades.

4.6. Time complexity analysis

In this section, the time complexity of the proposed AGDD method is analyzed. If we consider the iterative steps of the proposed method shown in Fig. 2, the time complexity can be expressed as follows.

Let n is the number of samples, d is the dimension of the samples, and k is the number of clusters. At the Calculation step, several $n \times n$ matrixes (GM and DM)

DOI: 10.22034/tjee.2022.15642

and $n \times 1$ vectors (GPM and DPM) are calculated based on the

Table 4. Results of the methods on multi-dimontional datasets in term of MPSR and NMI (Underlined red: first best result, Blue: second best result

Dataset	Evaluation Metric	AGDD	GDD	DBSCAN	OPTICS	k-means
New_thyroid	MPSR	**15.81**	**18.14**	29.30	30.23	32.56
	NMI	**0.61**	**0.53**	0.41	0.27	0.41
Breast-cancer-wisconsin	MPSR	**7.17**	41.58	34.85	34.99	**3.95**
	NMI	**0.49**	0.33	0.08	0.19	**0.75**
Iris	MPSR	**32.67**	34	**33.33**	71.33	34.66
	NMI	**0.69**	**0.73**	**0.73**	0.43	0.58
Zoo	MPSR	40.59	58.42	**19.8**	**27.72**	30.3
	NMI	0.13	0.49	**0.85**	0.6	**0.78**

unclustered samples. If we suppose that on average m samples ($m < n$) are clustered at each iteration, the time complexity at the Calculation step is:

$$(dn^2) + (d(n-m)^2) + (d(n-2m)^2) + \cdots + (d(n-(k-1)m)^2) \le (kdn^2) = O_1(kdn^2) \quad (12)$$

Then, the time complexity for selecting the densest samples as the centroid of the new clusters is:

$$(n) + (n-m) + (n-2m) + \cdots + (n-(k-1)m) \le (kn) = O_2(kn) \quad (13)$$

Afterward, similar to the above calculation, the time complexity of forming new clusters (i.e. assigning unclustered samples into their corresponding clusters) is also:

$$O_3(kn) \quad (14)$$

Finally, the time complexity to remove the clustered samples from the main sample set is:

$$O_4(1) \quad (15)$$

In all the calculations, the tasks with constant time complexity are considered as $O(1)$. By summing up these time complexities, the overall time complexity of the proposed AGDD method can be expressed as:

$$O_1(kdn^2) + O_2(kn) + O_3(kn) + O_4(1) = O(kdn^2) \quad (16)$$

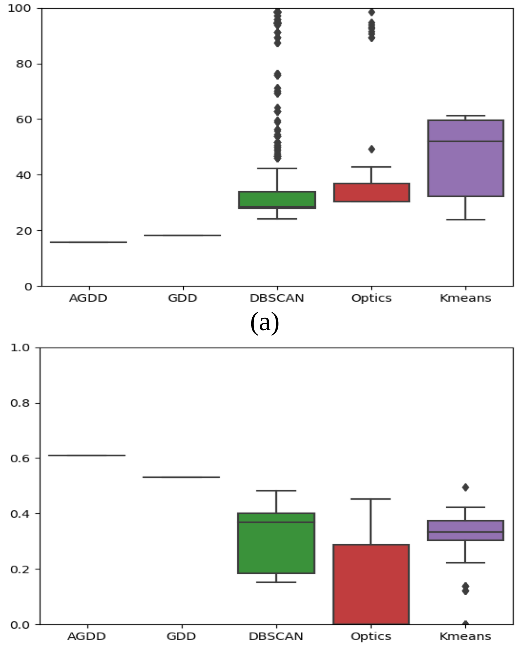
In the case of parallel computation, the time complexity of the Calculation step, $O_1(kdn^2)$, and forming a new cluster step, $O_3(kn)$, can be reduced into $O_1(kd)$ and $O_3(k)$, respectively. Therefore, the parallel processing reduces the overall time complexity into linear form, i.e.:

$$O_1(kd) + O_2(kn) + O_3(k) + O_4(1) = O(k(d+n)) \quad (17)$$

4.7. Advantages and limitations of AGDD method

The main advantages of the proposed method over the other methods can be summarized as follows:

- In contrast to k-means which tends to produce ellipsoid clusters with similar sizes [39], and need the number of clusters, GDD and AGDD can discover clusters with arbitrary shape and size without any hyper-parameter.
- The hyper-parameters of k-means, DBSCAN, and OPTICS need to be tuned through cross-validation, and their performance is highly dependent on their hyperparameters. In contrast, both GDD and AGDD form the clusters automatically without free parameters, and they are stable and repeatable, i.e., the clustering results do not change at different runs. To study the sensitivity of k-means, DBSCAN, and OPTICS to their parameters, the clustering results of these methods with different values of their parameters are illustrated in Fig. 9. Evidently, GDD and AGDD offer stability over different runs, whereas, in the other methods, different initializations can lead to different results.



(a)

(b)

Fig. 9. Sample box chart for methods with different hyper-parameter tuning. (a) MPSR, (b) NMI.

- GDD may fail to discover clusters with sudden density variations and generate several single-sample clusters [20] due to the interference of clustered samples in estimating the density and distance properties of remaining unclustered samples. However, the proposed AGDD method improves this implication by adaptive calculation of the parameters during clustering.

The limitations of the method can be clarified as follows:

- In practice, the performance of GDD and AGDD methods may degraded as the dimensionality increases due to the curse of dimensionality. This is because, density estimation using Gaussian model is not promising in high-dimensional space, especially when the number of samples is limited.
- Like most density-based clustering methods such as DBSCAN and OPTICS, when two neighboring clusters have a connection (e.g., Aggregation dataset), both GDD and AGDD link the clusters together. Therefore, if a user wants these clusters separately, another clustering algorithm is advised.
- In general, when there is no density or distance gap between the samples, most density-based clustering methods, including GDD and AGDD, may group all samples into one cluster. In this condition, a clustering technique such as k-means that gives the numbers of clusters as an input parameter performs better.

5. Conclusion

The GDD method has been proposed as a density-based clustering algorithm with no free parameters. In the GDD method, clustered samples and unclustered samples are incorporated into forming a cluster, leading to an inappropriate effect on clusters with different densities. In this paper, an adaptive approach called the AGDD method was proposed. It discards the clustered samples at each iteration and adaptively updates the parameters during the clustering process. The results showed the proposed AGDD performs similarly or better than GDD while AGDD incorporates smaller sample rates in the clustering process. Same as most density-based clustering methods, the performance of GDD and AGDD may decrease when there is no density or distance gap between the samples. For future work, we are going to optimize the method for image segmentation in which data includes spatial coordination and other features such as color information.

References

[1] P. Saini, J. Kaur, and S. Lamba, “A Review on Pattern Recognition Using Machine Learning,” *Lecture Notes in Mechanical Engineering*, pp. 619–627, 2021.
[2] C. Li, F. Kulwa, J. Zhang, Z. Li, H. Xu, and X. Zhao, “A review of clustering methods in microorganism image analysis,” *Advances in Intelligent Systems and Computing*, vol. 1186, Springer, pp. 13–25, 2021.
[3] M. Subramaniam, A. Kathirvel, E. Sabitha, and H. A. Basha, “Modified firefly algorithm and fuzzy c-mean clustering based semantic information retrieval,” *Journal of Web Engineering*, vol. 20, no. 1, pp. 33–52, 2021.
[4] A. R., Sardar, and R. Havangi, “Performance improvement of automatic clustering algorithm of colored images through preprocessing using Self-Organizing Maps (SOM) neural network,” *Tabriz Journal of Electrical Engineering*, vol. 47, no. 3, pp. 1073-1082, 2017.
[5] M. Kearns, Y. Mansour, and A. Y. Ng, “An Information-Theoretic Analysis of Hard and Soft Assignment Methods for Clustering,” *Learning in Graphical Models*, Springer Netherlands, pp. 495–520, 1998.
[6] D. J. Bora and D. A. K. Gupta, “A Comparative study Between Fuzzy Clustering Algorithm and Hard Clustering Algorithm,” *International Journal of Computer Trends and Technology*, vol. 10, no. 2, pp. 108–113, 2014.
[7] S, Rafiei, and P. Moradi, “Improving Performance of Fuzzy C-means Clustering Algorithm using Automatic Local Feature Weighting.,” *Tabriz Journal of Electrical Engineering*, vol. 46, no. 2, 2016.
[8] Rasmussen and E. M, “Clustering algorithms.,” *Information Retrieval: data Structures & algorithms*, vol. 419, p. 442, 1992, Accessed: Apr. 29, 2021.
[9] S. Kaushik, D. Kundu, S. Ghosh, S. Das, and A. Abraham. “Data clustering using multi-objective differential evolution algorithms. .” *Fundamenta Informaticae*, vol. 97, no. 4, pp. 381-403, 2009.
[10] J. Ak and R. Dubes, “Algorithms for clustering data. Englewood Cliff,” *NJ Prentice Hall*, 1988, Accessed: Apr. 29, 2021.
[11] J. MacQueen, “Some methods for classification and analysis of multivariate observations,” *Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, pp. 281–296, 1967, Accessed: Apr. 29, 2021.
[12] Ezugwu, A. E., Ikotun, A. M., Oyelade, O. O., Abualigah, L., Agushaka, J. O., Eke, C. I., & Akinyelu, A. A. “A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects.” *Engineering Applications of Artificial Intelligence*, vol. 110, pp. 104743, 2022.
[13] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise,” *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, pp. 226–231, 1996, Accessed: Apr. 29, 2021.
[14] M. Ankerst, M. M. Breunig, H. P. Kriegel, and J. Sander, “OPTICS: Ordering Points to Identify the Clustering Structure,” *ACM Sigmod record*, vol. 28, no. 2, pp. 49–60, 1999.
[15] Ram A, Jalal S, Jalal A S, Kumar M. “A density based algorithm for discovering density varied clusters in large spatial databases,” *International Journal of Computer Applications*, vol. 3, no. 6, pp. 1-4, 2010.
P. Bhattacharjee and P. Mitra, “A survey of density based clustering algorithms,” *Frontiers of Computer Science*, vol. 15, no. 1. Higher Education Press Limited Company, 2021.
[16] A. Smiti and Z. Elouedi, “DBSCAN-GM: An improved clustering method based on Gaussian Means and DBSCAN techniques,” *INES 2012 - IEEE 16th International Conference on Intelligent Engineering Systems*, pp. 573–578, 2012.
[17] C. Fraley and A. E. Raftery, “Model-based clustering, discriminant analysis, and density

estimation,” *Journal of the American Statistical Association*, vol. 97, no. 458, pp. 611–631, 2002.

[18] C. Fraley, A. E. Raftery, T. B. Murphy, and L. Scrucca, “mclust Version 4 for R: Normal Mixture Modeling for Model-Based Clustering, Classification, and Density Estimation.” Technical Report, Department of Statistics, University of Washington, no. 597, 2012.

[19] Jenni, V. R., Dua, A., Shobha, G., Shetty, J., & Dev, R. “Hybrid Density-based Adaptive Clustering using Gaussian Kernel and Grid Search,” *Recent Trends on Electronics, Information, Communication & Technology (RTEICT)*, pp. 221-226, 2021.

[20] E. Güngör and A. Özmen, “Distance and density based clustering algorithm using Gaussian kernel,” *Expert Systems with Applications*, vol. 69, pp. 10–20, 2017.

[21] A. S. Shirkhorshidi, S. Aghabozorgi, and T. Ying Wah, “A Comparison study on similarity and dissimilarity measures in clustering continuous data,” *PLoS One*, vol. 10, no. 12, 2015.

[22] C. Luo, Y. Li, and S. M. Chung, “Text document clustering based on neighbors,” *Data & Knowledge Engineering*, vol. 68, no. 11, pp. 1271–1288, 2009.

[23] K. Bache and M. Lichman, “UCI machine learning,”. https://ergodicity.net/2013/07/, accessed Apr. 29, 2021.

[24] “UCI machine learning repository,” Jul. 10, 2020. http://archive.ics.uci.edu/ml (accessed Apr. 29, 2021).

[25] A. K. Jain and M. H. C. Law, “Data Clustering : A User ’ s Dilemma,” pp. 1–10, 2005.

[26] S. Aghabozorgi, A. S. Shirkhorshidi, and T. Y. Wah, “Time-series clustering--a decade review,” *Information Systems*, vol. 53, pp. 16–38, 2015.

[27] J. C. Bezdek and N. R. Pal, “Some New Indexes of Cluster Validity,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 28, no. 3, pp. 301-315, 1998. Accessed: Apr. 29, 2021.

[28] H. Cui, M. Xie, Y. Cai, X. Huang, and Y. Liu, “Cluster validity index for adaptive clustering algorithms; Cluster validity index for adaptive clustering algorithms,” *IET Communications*, vol. 8, no. 13, pp. 2256–2263, 2013.

[29] R. Kashef and M. S. Kamel, “Enhanced bisecting k-means clustering using intermediate cooperation,” *Pattern Recognition*, vol. 42, no. 11, pp. 2557-2569, 2009.

[30] J. Lipor and L. Balzano, “Clustering quality metrics for subspace clustering.” *Pattern Recognition*, vol. 104, pp. 107328, 2020.

[31] M. Gaurav and S. K. Mohanty, “A fast hybrid clustering technique based on local nearest neighbor using minimum spanning tree,” Expert Systems with Applications, vol. 132, pp. 28-43, 2019.

[32] J. P. W. Pluim, J. B. A. Maintz, and M. A. Viergever, “Mutual-Information-Based Registration of Medical Images: A Survey,” *IEEE Transactions on Medical Imaging*, vol. 22, no. 8, 2003.

[33] W. M. Rand, “Objective criteria for the evaluation of clustering methods,” *Journal of the American Statistical Association*, vol. 66, no. 336, pp. 846–850, 1971.

[34] J. C. Rojas-Thomas, M. Santos, and M. Mora, “New internal index for clustering validation based on graphs,” *Expert Systems with Applications*, vol. 86, pp. 334–349, 2017.

[35] F. Ros and S. Guillaume, “Munec: a mutual neighbor-based clustering algorithm,” *Information Sciences*, vol. 486, pp. 148–170, 2019.

[36] J. Xie, Z. Y. Xiong, Q. Z. Dai, X. X. Wang, and Y. F. Zhang, “A new internal index based on density core for clustering validation,” *Information Sciences*, vol. 506, pp. 346–365, 2020.

[37] H. Yu, L. Y. Chen, J. T. Yao, and X. N. Wang, “A three-way clustering method based on an improved DBSCAN algorithm,” *Physica A: Statistical Mechanics and its Applications*, vol. 535, pp. 122289, 2019.

[38] H. Schütze, C. Manning, and P. Raghavan, “Introduction to information retrieval,” *Cambridge: Cambridge University Press*, vol. 39, pp. 234-65, 2008.

[39] A. Chokniwal and M. Singh, “Faster Mahalanobis K-means clustering for Gaussian distributions,” *International Conference on Advances in Computing, Communications and Informatics*, pp. 947–952, 2016.