

Article

A New Clustering Method Based on the Inversion Formula

Mantas Lukauskas *  and Tomas Ruzgas 

Department of Applied Mathematics, Faculty of Mathematics and Natural Sciences,
Kaunas University of Technology, 44249 Kaunas, Lithuania; tomas.ruzgas@ktu.lt

* Correspondence: mantas.lukauskas@ktu.lt

Abstract: Data clustering is one area of data mining that falls into the data mining class of unsupervised learning. Cluster analysis divides data into different classes by discovering the internal structure of data set objects and their relationship. This paper presented a new density clustering method based on the modified inversion formula density estimation. This new method should allow one to improve the performance and robustness of the k-means, Gaussian mixture model, and other methods. The primary process of the proposed clustering algorithm consists of three main steps. Firstly, we initialized parameters and generated a T matrix. Secondly, we estimated the densities of each point and cluster. Third, we updated mean, sigma, and phi matrices. The new method based on the inversion formula works quite well with different datasets compared with K-means, Gaussian Mixture Model, and Bayesian Gaussian Mixture model. On the other hand, new methods have limitations because this one method in the current state cannot work with higher-dimensional data ($d > 15$). This will be solved in the future versions of the model, detailed further in future work. Additionally, based on the results, we can see that the MIDEv2 method works the best with generated data with outliers in all datasets (0.5%, 1%, 2%, 4% outliers). The interesting point is that a new method based on the inversion formula can cluster the data even if data do not have outliers; one of the most popular, for example, is the Iris dataset.

Keywords: artificial intelligence; unsupervised machine learning; clustering; nonparametric density estimation; inversion formula

MSC: 62G05; 62G07; 62G30



Citation: Lukauskas, M.; Ruzgas, T. A New Clustering Method Based on the Inversion Formula. *Mathematics* **2022**, *10*, 2559. <https://doi.org/10.3390/math10152559>

Academic Editor: Alicia Nieto-Reyes

Received: 21 June 2022

Accepted: 18 July 2022

Published: 22 July 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Artificial intelligence was first mentioned in 1956, but it was not so widely applied for a long time. Artificial intelligence has been widely used in recent decades. The ever-increasing power of possible computations has driven the high availability, development, and applications of artificial intelligence. Data mining is one of the most critical areas, as it is not limited to business, manufacturing, or other services. For this reason, data research has attracted a large number of researchers. Data clustering is one area of data mining that falls into the class of data mining of unsupervised learning. Cluster analysis divides data into different classes by discovering the internal structure of data set objects and their relationship. Clustering aims to create groups of similar observations/elements. The most similar elements, in this case, will be in one cluster and different elements in separate clusters [1].

With the increasing application of data mining, cluster analysis of data is also being applied in many areas: pattern recognition [2,3], bioinformatics [4,5], environment sciences [6], feature selection [7,8], or to solve different healthcare tasks. Clustering algorithms can be used to detect various diseases [9]. For example, different clustering techniques are used to identify breast cancer [10], Parkinson's disease [11,12], various psychological and psychiatric disorders [13], heart diseases and diabetes [14], and Alzheimer's disease [15,16], among many others.

Although there are many clustering methods, this problem is being addressed and remains a complex issue. Different clustering methods often do not work well with all data sets, and different methods are very much needed. Although one of the most widely used algorithms currently used is the k-means, as these methods are fast-acting and work well with certain data sets, there are still a lot of possible improvements to increase this method's accuracy.

Research focuses a lot on developing new density estimation procedures [17,18]. Moreover, in the last years, different scientists started to propose different robust density estimation methods even based on neural networks. There are different researches on this topic: Parzen neural networks [19], soft constrained neural networks [20], and others [21]. Some time ago, we presented a modified inversion formula for density estimation [22]. In this research, we found that this density estimation works better with different data than multiple density estimators. Therefore, we raised the hypothesis that modified inversion formula density estimation would be suitable for data clustering. Due to these facts, this paper aimed to present a new density clustering method based on the modified inversion formula density estimation. This new method should allow one to improve the performance and robustness of the k-means, Gaussian mixture model, and other methods. The main process of the proposed clustering algorithm consists of three main steps. Firstly, we initialized parameters and generated a T matrix. Secondly, we estimated the densities of each point and cluster. Third, we updated mean, sigma, and phi matrices. To compare results in this paper we used k-means, Gaussian mixture model (GMM), and Bayesian Gaussian mixture model (BGMM) clustering methods.

This paper is organized as follows. In the Section 2, we present introduction of the inversion formula and modified inversion formula density estimations and explain the idea behind these estimations. Then, the process of the proposed algorithm is presented in Section 2. In Section 3, we show empirical results with datasets used in the research, evaluation metrics, and experimental results. Finally, conclusions and future work with the new clustering method are given in Section 4

2. Estimation of the Density of the Modified Inversion Formula

Estimating probability density functions (pdf) is considered one of the most important parts of statistical modeling. This feature allows us to express random variables as a function of other variables while simultaneously allowing the detection of potentially hidden relationships in the data. If distribution density $f(x)$ satisfies the equation

$$f(x) = \sum_{k=1}^q p_k f_k(x) = f(x, \theta) \quad (1)$$

the random vector $X \in \mathbb{R}^d$ satisfies the distribution mixture model. The formula above (1), θ is a multi-dimensional parameter of the model. The function $f_k(x)$ is a function of the distribution density. X is a d-dimensional random vector with a distribution density $f(x)$. Additionally, we have independent copies of X ($X(1), \dots, X(n)$) (sample of X).

We say that the sample satisfies the mixture model if $X(t)$ satisfies (1). We call the size n the sample size (volume). The parameter q is called the mixture number of components, and p_k is the a priori probability. They meet the following conditions:

$$p_k > 0, \sum_{k=1}^q p_k = 1 \quad (2)$$

2.1. Gaussian Mixture and Inversion Density Estimation

It is important to notice that Projection (3) of the observations of the Gaussian mixture (1) is also distributed by the (one-dimensional) Gaussian mixture model:

$$f_{\tau}(x) = \sum_{k=1}^q p_{k,\tau} \varphi_{k,\tau}(x) = f_{\tau}(x, \theta_{\tau}) \tag{3}$$

here $\varphi_{k,\tau}(x) = \varphi(x; m_{k,\tau}, \sigma_{k,\tau}^2)$ —one-dimensional Gaussian density. Multivariate mixture parameter and data projection distribution parameters $\theta_{\tau} = (p_{k,\tau}, m_{k,\tau}, \sigma_{k,\tau}^2)$, $k = 1, \dots, q$ links equality

$$\begin{aligned} p_{j,\tau} &= p_j \\ m_{j,\tau} &= \tau' M_j \\ \sigma_{j,\tau}^2 &= \tau' R_j \tau \end{aligned} \tag{4}$$

Using the inversion formula,

$$f(x) = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} e^{-it'x} \psi(t) dt \tag{5}$$

where $\psi(t) = Ee^{it'x}$ denotes the characteristic function of the random variable X. First, the set of projections directions T is selected. Additionally, the characteristic function is changed using the following formula:

$$\hat{f}(x) = \frac{A(d)}{\#T} \sum_{\tau \in T} \int_0^{\infty} e^{-iu\tau'x} \hat{\psi}_{\tau}(u) u^{d-1} e^{-hu^2} du \tag{6}$$

where here and below, # denotes the number of elements in the set. With the formula for the volume of a d-dimensional sphere

$$V_d(R) = \frac{\pi^{\frac{d}{2}} R^d}{\Gamma(\frac{d}{2} + 1)} = \begin{cases} \frac{\pi^{\frac{d}{2}} R^d}{(\frac{d}{2})!}, & \text{kai } d \text{ mod } 2 \equiv 0 \\ \frac{2^{\frac{d+1}{2}} \pi^{\frac{d-1}{2}} R^d}{d!!}, & \text{kai } d \text{ mod } 2 \equiv 1 \end{cases} \tag{7}$$

one can calculate the constant A(d) depending on the dimension of the data:

$$A(d) = \frac{(V_d(1))'_R}{(2\pi)^d} = \frac{d2^{-d} \pi^{-\frac{d}{2}}}{\Gamma(\frac{d}{2} + 1)} \tag{8}$$

Simulation studies show that the density estimates of the inversion formula are discontinuous/rough. The multiplier e^{-hu^2} in the Formula (6) further smoothes the estimate $\hat{f}(x)$ with the Gaussian kernel function. It is worth noting that this form of the multiplier allows analytical calculation of the value of the integral. Furthermore, results from extended Monte Carlo studies have shown that using this multiplier reduces the estimation errors. Formula (6) can be used for various estimates of the characteristic function of the projected data. A parametric estimate of the characteristic function was used in the present case.

$$\hat{\psi}_{\tau}(u) = \sum_{k=1}^{\hat{q}_{\tau}} \hat{p}_{k,\tau} e^{iu\hat{m}_{k,\tau} - u^2 \hat{\sigma}_{k,\tau}^2 / 2} \tag{9}$$

The chosen form of the smoothing multiplier e^{-hu^2} allows us to relate the smoothing parameter h to the variances of the projection clusters.

2.2. Modified Inversion Density Estimation

It is worth noting that the Gaussian mixture Model (1) described by the estimate (where $f_k = \varphi_k$) only estimates the density of the distribution close to it well. This can be seen as a drawback of the inversion formula method (9). The density estimation of the inversion formula often becomes complicated due to a large number of components with low a priori probability when the aim is to approximate the density under study with a mixture of Gaussian distributions. This problem can be solved by using a noise cluster.

We discuss a modified density estimation algorithm based on a multivariate Gaussian mixture model (Algorithm 1). However, first, let us define the parametric estimate of the characteristic function of a uniform distribution density:

$$\hat{\psi}(u) = \frac{2}{(b-a)u} \sin \frac{(b-a)u}{2} \cdot e^{\frac{i u(a+b)}{2}} \tag{10}$$

In the formula for calculating the density estimate, construct the estimate of the characteristic Function 9 as a union of the characteristic functions of a mixture of Gaussian distributions and a uniform distribution with corresponding a priori probabilities.

$$\hat{\psi}_\tau(u) = \sum_{k=1}^{\hat{q}_\tau} \hat{p}_{k,\tau} e^{i u \hat{m}_{k,\tau} - u^2 \hat{\sigma}_{k,\tau}^2 / 2} + \hat{p}_{0,\tau} \frac{2}{(b-a)u} \sin \frac{(b-a)u}{2} \cdot e^{\frac{i u(a+b)}{2}} \tag{11}$$

Here, the second term describes the noise cluster with an even distribution, and \hat{p}_0 is the weight of the noise cluster,

$$a(\tau) = (\tau'x)_{\min} - \frac{(\tau'x)_{\max} - (\tau'x)_{\min}}{2(n-1)} \tag{12}$$

$$b(\tau) = (\tau'x)_{\max} + \frac{(\tau'x)_{\max} - (\tau'x)_{\min}}{2(n-1)}. \tag{13}$$

Algorithm 1: Clustering Algorithm Based on the Modified Inversion Formula Density Estimation (MIDE)

Input: Data set $X = [X_1, X_2, \dots, X_n]$, cluster number K

Output: C_1, C_2, \dots, C_t and $\hat{M}, \hat{p}_k, \hat{R}$

Possible initiation of mean vector:

- (1) random uniform initialization
- (2) k-means
- (3) random point initialization

Generate a T matrix. The set T is calculated when the design directions are evenly spaced on the sphere.

- 1 **For** $i = 1$: **t do**
 - 2 Density estimation for each point and cluster based on (9)
 - 2 Update $\hat{M}, \hat{p}_k, \hat{R}$ values based on (22, 23, 24)
 - 4 **End**
 - 5 **Return** C_1, C_2, \dots, C_t and $\hat{M}, \hat{p}_k, \hat{R}$
-

2.3. Modified Inversion Density Clustering Algorithm

This section aims to overview the critical aspects of the new modified inversion density estimation (MIDE) clustering method (Algorithm 1). This clustering algorithm uses the EM (expectation maximization) algorithm. The selection of the initial parameters of the EM algorithm is of particular importance for the clustering results, as each new combination of parameters can steer the cluster in a different direction. Random parameter selection is one of the most commonly used solutions for parameter initialization [23,24]. Random selection of initial parameters is a reasonably simple solution, as it is easy to implement. However, one of the significant disadvantages of this method is that such initialization

often results in significant deviations in the clustering results. In addition, the algorithm uses a continuous partition to initialize the initial cluster centers.

$$p(x) = \frac{1}{b - a} \tag{14}$$

In addition, another method of initialization presented in the software algorithm solution is the selection of random points. In this case, the initial cluster centers are selected not randomly from the entire space but by randomly selecting a point from the observations in the data set. However, this selection also has several drawbacks, as randomly selected points can be too close to each other, and selected points can also be exceptions in the data.

Hierarchical clustering can also be used to address the potential shortcomings of the random cluster center selection method. For the first time, such a classification algorithm that maintains a Gaussian mixture model to form a cluster tree was described by Fraley in 1998 [25]. Maitra [26] proposed a hierarchical clustering based on mean connectivity to obtain an initial model mean. Moreover, Meila and Heckerman [27] experimentally demonstrated that an algorithm using a pattern-based distance measure is better than a random method. This method is applied to the initial mean of the model. Perhaps the only major drawback it has observed so far is that computer computations take a long time and require a large amount of computer memory if there are many data.

One of the most commonly used methods for selecting cluster centers is the k-means and other heuristic clustering methods. This is one of the most widely used initial parameters selection methods. In the case of the initialization of K-means, firstly, the random cluster centers $\mu_1, \mu_2, \dots, \mu_k \in \mathcal{R}^n$ are first selected, and then the procedure is performed until convergence is achieved.

$$c^{(i)} = \operatorname{argmin}_j \|x^{(i)} - \mu_j\|^2 \tag{15}$$

$$\mu_j = \frac{\sum_{i=1}^m 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^m 1\{c^{(i)} = j\}} \tag{16}$$

Clustering using the modified inversion formula density estimation and the EM algorithm is explained below. If the distribution density X of a random vector has q maxima, then it can be approximated by a mixture of q single-mode distribution densities:

$$f(x) = \sum_{k=1}^q p_k f_k(x) \tag{17}$$

Suppose that the distribution of X depends on a random variable v , which acquires the values $1, \dots, q$ with the corresponding probabilities p_1, \dots, p_q . In classification theory, v is interpreted as the number of the class to which the observed object belongs. Thus, the $X(t)$ observations would correspond to $v(t), t = 1, \dots, n$. The functions f_k are treated as the density of the conditional distribution X under the condition $v = k$. Based on this approach, loose clustering of the sample is understood as a posteriori probabilities.

$$\pi_k(x) = P\{v = k | X = x\} \tag{18}$$

when all $x \in \{X(1), \dots, X(n)\}$. Strict clustering of the sample would be an estimate of the random variables $v(1), \dots, v(n)$ Take a breakdown into subsets based on equality

$$\hat{v}(t) = \operatorname{arg} \max_{k=1, \dots, q} \hat{\pi}_k(X(t)) \tag{19}$$

The estimates $\hat{\pi}_k$ are obtained by approximating the unknown distribution density components with the density estimates of the inversion formula and using the EM (expectation maximization) algorithm. We briefly describe it as follows. Suppose that

Equation (15) holds and f_k is the density function of the inversion formula for the Gaussian mixture model, $k = 1, \dots, q$, where q is the number of the clusters. In this case (17), let us denote the right side of the equation by $f(x, \theta)$, where $\theta = (p_k, M_k, R_k, a, b, k = 1, \dots, q)$. Equality applies:

$$\pi_k(x) = \frac{p_k f_k(x)}{f(x, \theta)} \text{ and } k = \overline{1, q} \tag{20}$$

Having an estimate of θ , the estimates of the probabilities π_k (k -th cluster probability) are obtained from (20) using the “embedding” method, i. y. replacing the unknown parameters on the right with their statistical estimates. The EM algorithm is a reciprocal procedure for estimating the maximum likelihood

$$\theta^* = \underset{\theta}{\operatorname{argmax}} L(\theta), \quad L(\theta) = \prod_{t=1}^n f(X(t), \theta) \tag{21}$$

and to calculate the corresponding estimates $\hat{\pi}_k$. Several authors have independently proposed this algorithm for Gaussian mixture analysis, including Hasselblad [28] and Behboodian [29]. Its properties were later well examined in refs. [30–32] and other works. The EM algorithm has received much attention in various review articles and monographs [33–35]. Suppose that after r cycles, we obtain the estimates $\hat{\pi}_k = \hat{\pi}_k^{(r)}$. The new estimate $\hat{\theta} = \hat{\theta}^{(r+1)}$ is then defined by the equations:

$$\hat{p}_k = \frac{1}{n} \sum_{t=1}^n \hat{\pi}_k(X(t)) \tag{22}$$

$$\hat{M}(k) = \frac{1}{n \hat{p}_k} \sum_{t=1}^n \hat{\pi}_k(X(t)) \cdot X(t) \tag{23}$$

$$\hat{R}(k) = \frac{1}{n \hat{p}_k} \sum_{t=1}^n \hat{\pi}_k(X(t)) [X(t) - \hat{M}(k)] \cdot [X(t) - \hat{M}(k)]' \tag{24}$$

where $k = 1, \dots, q$. Entering $\hat{\theta}^{(r+1)}$ to the right of (20), we find $\hat{\pi}_k^{(r+1)}(X(t))$, $k = \overline{1, q}$, $t = \overline{1, n}$. As a result of this recursive procedure, we obtain a non-decreasing sequence $L(\hat{\theta}^{(r)})$, but whether it converges to the point of the global maximum depends very much on the initial estimate $\hat{\theta}^{(0)}$ (or $\hat{\pi}^{(0)}$).

In the case of the high mixture model (GMM), the best number of clusters is selected based on the information criterion. This algorithm’s most commonly used information criteria are AIC, BIC, and others. When these information criteria reach their global minimum or maximum, an optimal number of clusters can be said to have been reached. However, there are also some problems in applying these criteria. First, it is necessary to calculate the global maximum of the function as the maximum value of the local maxima, but sometimes this is performed with exceptions. Therefore, applying any procedure cannot guarantee that such a global maximum will be found in such a case.

On the other hand, applying these criteria assumes that one of the parametric methods being compared is correct. This assumption makes the criterion unstable. The arguments presented to raise the question of whether it may be worthwhile to use nonparametric criteria to test the adequacy of the distribution mix model. Several problems can be encountered if the correct number of clusters is not selected. If the number of components selected is too small, then no clear clusters are formed, and one cluster includes more. Meanwhile, if the selected number of clusters is too large, it is much more challenging to calculate clusters in the first place, and less generalizing clusters are also obtained. An attempt to accurately select the number of clusters was provided by Xie, et al. [36], in which an adaptive selection of components/clusters of the Gaussian mixture model was proposed.

3. Experimental Analyses

This section provides information about the modified inversion function based on the proposed clustering method. This section consists of three parts. The first part provides information on the clustering assessment methods used in the empirical study. The second part of this chapter provides information on the data sets used in the study. Finally, the third part of the chapter presents the study’s main results.

3.1. Evaluation Metrics

This section presents the main evaluation metrics used in the empirical study. In order to evaluate the results of clustering, it is essential to choose the appropriate evaluation metrics, as they can also determine the evaluation of clustering. In this study, clustering methods were used to compare J-Score [37], Normalized Mutual Information (NMI) [38], Adjusted Rand Index (ARI) [39] and Accuracy (ACC) [40], and The Fowlkes–Mallows index (FMI) [41]. These metrics were chosen based on the fact that the actual data clusters are known in advance because if the clusters were not known in advance, then the evaluation metrics could be: Calinski and Harabasz score, also known as Variance Ratio Criterion [42], Davies–Bouldin score [43], or others.

J-score. Ahmadinejad and Liu [37] suggested a new clustering evaluation metric, J-score. The J-score is a simple and robust measure of clustering accuracy. It addresses the matching problem and reduces the risk of overfitting that challenge existing accuracy measures [37]. Bidirectional set matching: Suppose a dataset contains N datapoints belonging to T true classes, and cluster analysis produces K hypothetical clusters. To establish the correspondence between T and K , we first considered each class as reference and identify its best-matched cluster ($T \rightarrow K$). Specifically, for a class $t \in T$, we searched for a cluster $k \in K$ that has the highest Jaccard index,

$$I_t = \max_{k \in K} \frac{|V_t \cap V_k|}{|V_t \cup V_k|} \tag{25}$$

where V_t and V_k are the set of datapoints belonging to class t and cluster k , respectively, and $|\cdot|$ denotes the size of a set. We then considered each cluster as a reference and identified its best-matched class ($K \rightarrow T$) using a similar procedure. For a cluster $k \in K$, we searched for a class $t \in T$ with the highest Jaccard index,

$$I_k = \max_{t \in T} \frac{|V_t \cap V_k|}{|V_t \cup V_k|} \tag{26}$$

Calculating overall accuracy: To quantify the accuracy, we aggregated Jaccard indices of individual clusters and classes, accounting for their relative sizes (i.e., number of data points). We first calculated a weighted sum of I_t across all classes as $R = \sum_{t \in T} \left(\frac{V_t}{N} I_t \right)$, and a weighted sum of I_k across all clusters as $P = \sum_{k \in K} \left(\frac{V_k}{N} I_k \right)$. We then took their harmonic mean as J score,

$$J = \frac{2 \times R \times P}{R + P} \tag{27}$$

To work with this metric, we implemented the calculation of the J-score metric in the Python programming language. The program code is available in Appendix A.

Normalized Mutual Information (NMI). The mutual information (MI) of two random variables is a measure of the mutual dependence of the two variables. MI normalization was performed for greater comparability and better interpretation, and the NMI metric was obtained. The values of this metric can range from 0 to 1.0; in this case, zero would indicate no relationship between the variables, while one would indicate a perfect correlation.

$$NMI = \frac{MI(Y', Y)}{\sqrt{H(Y')H(Y)}} \quad (28)$$

Here, Y' are the predicted labels and Y are the actual classes known in advance. $MI(Y', Y)$ is the mutual information between predicted labels and actual labels. This formula also uses the entropy $H(*)$ of predicted labels and actual labels.

Adjusted Rand Index (ARI). The Rand index evaluates the similarity between two clusters. Pairs of all observations are used to calculate this similarity. When calculating this index, there are observations in the assignment to clusters, and how this coincides with the real labels of the clusters.

$$ARI = \frac{2(ad - bc)}{(a + b)(b + d) + (a + c)(c + d)} \quad (29)$$

In the given formula, a is a number that describes how many data points are correctly assigned to a cluster. b is the number of observations in a pair assigned to the same cluster (predicted and actual cluster values match). Here, c is the number of observations in a pair for which the predicted cluster matches, but the actual cluster values do not match. Finally, d is the number of data points in a pair that neither the predicted case nor the actual case belongs to the same cluster.

Accuracy (ACC). Accuracy is often used to measure the quality of classification. It is also used for clustering. It is calculated as the sum of the diagonal elements of the confusion matrix, divided by the number of samples to obtain a value between 0 and 1.

$$ACC = \frac{1}{N} \sum_{i=1}^k n_i \quad (30)$$

where N is the total number of data points in the dataset, n_i is the number of data points correctly divided into the corresponding cluster i , and k is the cluster number.

The Fowlkes–Mallows index (FMI). The Fowlkes–Mallows score FMI is defined as the geometric mean of pairwise precision and recall.

$$FMI = \frac{TP}{\sqrt{(TP + FP)(TP + FN)}} \quad (31)$$

True Positive (TP) is the number of pairs of points belonging to the same clusters (true label = predicted label). False Positive (FP) is the number of the points that belong to the same cluster in the true labels but do not belong to the same cluster in the predicted clusters. False Negative (FN) is the number of the pairs of points that belong in the same clusters in the predicted labels and not in the true labels. The higher the metric value, the better the cluster separation is (the maximum possible value of the metric is 1, and the minimum is 0).

3.2. Experimental Datasets

To test the developed method and compare it with other methods, 25 data sets were used in this study. Data sets can be divided into three categories: synthetic, real, and generated data with outliers. Synthetic data sets are data sets that have been generated by other authors and are often used in research on clustering methods. Actual datasets include datasets such as Iris, Wine, Diabetes, and others, and these datasets are also selected based on datasets used by other authors. The third category of generated data with outliers is generated as Gaussian data, including a certain amount of outliers: 0.5%, 1%, 2%, and 4%. These datasets aim to evaluate how different methods work with data with an appropriate amount of outliers. The table below (see Table 1) shows the data sets used.

Table 1. A description of the data set used.

ID	Data Sets	Sample Size (N)	Dimensions (D)	Classes
<i>Synthetic</i>				
1	Aggregation	788	2	7
2	Atom	800	3	2
3	D31	3100	2	31
4	R15	600	2	15
5	Gaussians1	100	2	2
6	Threenorm	1000	2	2
7	Twenty	1000	2	20
8	Wingnut	1016	2	2
<i>Real</i>				
9	Breast	570	30	2
10	CPU	209	6	4
11	Dermatology	366	17	6
12	Diabetes	442	10	4
13	Ecoli	336	7	8
14	Glass	214	9	6
15	Heart-statlog	270	13	2
16	Iono	351	34	2
17	Iris	150	4	3
18	Wine	178	13	3
19	Thyroid	215	5	3
<i>Generated clusters with outliers</i>				
20	2 clusters (0.5% outliers)	1005	2	2
21	2 clusters (1% outliers)	1010	2	2
22	2 clusters (2% outliers)	1020	2	2
23	2 clusters (4% outliers)	1040	2	2
25	3 clusters (0.5% outliers)	1005	2	3
26	3 clusters (1% outliers)	1010	2	3
27	3 clusters (2% outliers)	1020	2	3
28	3 clusters (4% outliers)	1040	2	3

3.3. Performances of Clustering Methods

To avoid the possible influence of successful parameter initialization on test results, all experiments were performed 10,000 times. For the k-means method, initial cluster centers were randomly selected based on the 100 runs best solutions. For GMM (Gaussian Mixture Model), BGMM (Bayesian Gaussian Mixture Model) and clustering based on modified inversion density estimation (MIDE) initial center were selected based on the k-means centers initialization. The following table provides information on the Accuracy metric values for the different clustering algorithms. Other evaluation metrics like NMI, ARI, FMI, and J-Score can be found in the Appendix B tables.

The accuracy results for different datasets are presented in Table 2. It can be seen that the new method based on the inversion formula works quite well with different datasets compared with K-means, Gaussian Mixture Model (GMM), and Bayesian Gaussian Mixture model (BGMM).

Table 2. Different models were comparative (means and standard deviation) based on the accuracy (ACC) for 10,000 runs.

Dataset	K-Means		GMM		BGMM		MIDEv1		MIDEv2	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
<i>Synthetic</i>										
Aggregation	0.857	0.005	0.835	0.075	0.907	0.042	0.889	0.008	0.895	0.009
Atom	0.710	0.002	0.618	0.028	0.637	0.022	0.723	0.002	0.746	0.004
D31	0.972	0.015	0.928	0.028	0.601	0.022	0.721	0.017	0.723	0.013
R15	0.997	0.000	0.979	0.036	0.669	0.011	0.768	0.008	0.855	0.007
Gaussians1	1.000	0.000	1.000	0.000	1.000	0.000	1.000	0.000	1.000	0.000
Threenorm	0.591	0.001	0.612	0.047	0.549	0.006	0.649	0.003	0.679	0.003
Twenty	1.000	0.000	0.985	0.029	0.838	0.075	-	-	-	-
Wingnut	0.909	0.000	0.964	0.000	0.965	0.000	0.876	0.000	0.880	0.000
<i>Real</i>										
Breast	0.908	0.003	0.940	0.001	0.933	0.001	-	-	-	-
CPU	0.738	0.008	0.574	0.073	0.590	0.093	0.808	0.007	0.828	0.006
Dermatology	0.739	0.044	0.737	0.080	0.756	0.109	-	-	-	-
Diabetes	0.356	0.010	0.419	0.043	0.439	0.033	0.420	0.008	0.448	0.007
Ecoli	0.649	0.013	0.753	0.018	0.739	0.006	0.714	0.011	0.754	0.009
Glass	0.447	0.016	0.468	0.025	0.483	0.025	0.465	0.013	0.487	0.017
Heart-statlog	0.837	0.002	0.794	0.045	0.791	0.045	-	-	-	-
Iono	0.707	0.000	0.810	0.029	0.803	0.023	-	-	-	-
Iris	0.831	0.007	0.953	0.065	0.838	0.049	0.933	0.006	0.955	0.005
Wine	0.966	0.000	0.953	0.048	0.977	0.038	0.943	0.003	0.953	0.004
Thyroid	0.874	0.000	0.953	0.029	0.917	0.035	0.754	0.007	0.778	0.009
<i>Generated blobs with outliers</i>										
2 clusters (0.5% outliers)	0.995	0.000	0.995	0.000	0.995	0.000	0.995	0.000	1.000	0.000
2 clusters (1% outliers)	0.989	0.000	0.990	0.000	0.990	0.000	0.990	0.000	0.996	0.000
2 clusters (2% outliers)	0.979	0.000	0.980	0.000	0.980	0.000	0.981	0.000	0.997	0.000
2 clusters (4% outliers)	0.961	0.000	0.962	0.000	0.962	0.000	0.964	0.000	0.996	0.000
3 clusters (0.5% outliers)	0.994	0.000	0.994	0.000	0.994	0.000	0.994	0.000	0.999	0.000
3 clusters (1% outliers)	0.989	0.000	0.989	0.000	0.989	0.000	0.989	0.000	0.997	0.000
3 clusters (2% outliers)	0.979	0.000	0.979	0.000	0.979	0.000	0.981	0.000	0.997	0.000
3 clusters (4% outliers)	0.961	0.000	0.951	0.000	0.945	0.000	0.965	0.000	0.996	0.000

Bold underlined values indicate best results for each dataset.

4. Discussion

Research focuses a lot on developing new density estimation procedures [17,18]. Moreover, in the last years, different scientists started to propose different robust density estimation methods even based on neural networks such as Parzen neural networks [19], soft constrained neural networks [20], and others [21]. This paper presented a new clustering method based on the modified inversion formula density estimation (MIDE). This new method improves the performance and robustness of the k-means, Gaussian mixture model, and other methods. Method working: Firstly, we initialized parameters and generated a T matrix. Secondly, we estimated the densities of each point and cluster based on the modified inversion formula. Third, we updated mean, sigma, and phi matrices. Based on the results presented earlier, it is possible to conclude that the newly presented method works well with different clustering datasets even if the datasets do not have any outliers. Results based on the generated clusters data with outliers showed that the newly presented method (MIDEv2) works the best in all situations (0.5%, 1%, 2%, and 4%). Based on the accuracy metric with all of these datasets, accuracy was higher than 0.995. The interesting point is that a new method based on the inversion formula can cluster the data even if data do not have outliers; one of the most popular, for example, is the Iris data set. When we compared the accuracy results in other datasets, it can be mentioned that the MIDE method achieved 0.955 accuracy on the Iris dataset compared with the second-best GMM

method with 0.953 accuracy; using the ARI metric for this dataset, MIDE methods as well showed better results compared with other methods. Based on the NMI, J-Score, and FMI metrics (see Table A1), a better method for the Iris dataset would be GMM. It is hard to compare and use multiple metrics because, in this research, we used accuracy as our main metric. After all, all datasets have labels, and it is possible to calculate accuracy of our clustering methods. Compared with other researchers' results in the past, Sun et al. were able to achieve 0.925 accuracy with the SVC-KM approach [44], and Hyde and Angelov achieved 0.950 accuracy with DDC (Data Density-Based Clustering) [45]. Additionally, it is notable that the MIDE method has a lower standard deviation than other methods used in this research. It is worth mentioning that this method also has limitations. Based on the experimental study, this one method in the current state can not work with higher dimensional data ($d > 15$). This occurs due to T matrix generation; as dimensions grow, finding a suitable T matrix becomes harder. This one will be solved in the future versions of the model; we will present more about it in future work. Another method problem is speed; the current stage method is slower than other methods, but this problem can be solved with parallelization of the process on the programming side. The future direction of the newly created method is this method application for deep clustering. It can be seen that MIDEv1 and MIDEv2 methods do not work very well with higher-dimension data. Due to that, the deep clustering method with an encoder structure could solve this problem.

Author Contributions: Conceptualization, T.R. and M.L.; methodology, T.R.; software, T.R. and M.L.; formal analysis, T.R. and M.L.; investigation, T.R. and M.L.; writing—original draft preparation, T.R., M.L.; writing—review and editing, M.L.; supervision, T.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors thank the area editor and the reviewers for giving valuable comments and suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

J-score metric calculation program code with Python language

```
import numpy as np
def JScore(truth, pred):
    if (len(truth) == len(pred)):
        print("Equal lengths")
        A = np.empty([0, len(truth)], bool)
        test = list(set(pred))
        for i in test:
            A = np.vstack([A, (np.array(pred) == i)])
        suma = A.sum(axis=1)
        B = np.empty([0, len(truth)], bool)
        test = list(set(truth))
        for i in test:
            B = np.vstack([B, (np.array(truth) == i)])
        suma2 = B.sum(axis=1)
        C = np.empty([len(suma), len(suma2)], float)
        for i in range(0, len(suma)):
            for j in range(0, len(suma2)):
                C[i, j] = sum(A[i,] & B[j,])/sum(A[i,] | B[j,])
```

```

M1 = sum(np.amax(C, axis=1) * suma)/A.shape[1]
M11 = sum(np.amax(C, axis=0) * suma2)/A.shape[1]
M2 = 2 * M1 * M11/(M1 + M11)
return M2
else:
    print('Truth and Pred have different lengths.')

```

Appendix B

Table A1. Comparative table of different models (means and standard deviation) based on Normalized Mutual Information (NMI) for 10,000 runs.

Dataset	K-Means		GMM		BGMM		MIDEv1		MIDEv2	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
<i>Synthetic</i>										
Aggregation	0.836	0.004	0.886	0.035	<u>0.909</u>	0.041	0.779	0.006	0.845	0.005
Atom	0.289	0.003	0.170	0.036	0.194	0.028	0.310	0.004	<u>0.319</u>	0.003
D31	<u>0.969</u>	0.005	0.951	0.008	0.871	0.004	0.791	0.007	0.822	0.006
R15	<u>0.994</u>	0.000	0.989	0.012	0.868	0.014	0.881	0.001	0.909	0.001
Gaussians1	1.000	0.000	1.000	0.000	1.000	0.000	<u>1.000</u>	0.000	<u>1.000</u>	0.000
Threenorm	0.024	0.001	0.047	0.039	0.007	0.002	0.069	0.001	<u>0.076</u>	0.001
Twenty	<u>1.000</u>	0.000	0.996	0.008	0.956	0.026	<u>1.000</u>	0.000	0.988	0.005
Wingnut	0.562	0.000	0.778	0.002	<u>0.779</u>	0.000	0.459	0.000	0.420	0.001
<i>Real</i>										
Breast	0.547	0.011	0.659	0.003	0.630	0.003	-	-	-	-
CPU	0.487	0.013	0.398	0.025	0.389	0.033	0.467	0.013	<u>0.529</u>	0.011
Dermatology	0.862	0.009	0.809	0.044	0.862	0.049	-	-	-	-
Diabetes	0.090	0.004	0.084	0.041	0.105	0.017	0.089	0.004	<u>0.106</u>	0.003
Ecoli	0.636	0.004	0.636	0.016	<u>0.639</u>	0.010	0.592	0.004	0.534	0.004
Glass	0.303	0.019	0.327	0.052	0.364	0.042	0.304	0.020	<u>0.369</u>	0.024
Heart-statlog	<u>0.363</u>	0.005	0.270	0.055	0.263	0.058	0.339	0.008	0.308	0.007
Iono	0.125	0.000	0.305	0.052	0.299	0.024	-	-	-	-
Iris	0.657	0.006	<u>0.890</u>	0.04	0.751	0.011	0.841	0.007	0.763	0.008
Wine	<u>0.876</u>	0.000	0.856	0.055	0.926	0.054	0.822	0.001	0.799	0.003
Thyroid	0.559	0.000	<u>0.783</u>	0.059	0.661	0.051	0.382	0.009	0.390	0.008
<i>Generated clusters with outliers</i>										
2 clusters (0.5% outliers)	0.976	0.000	0.976	0.000	0.976	0.000	0.977	0.000	<u>1.000</u>	0.000
2 clusters (1% outliers)	0.947	0.000	0.957	0.000	0.957	0.000	0.958	0.000	<u>0.974</u>	0.000
2 clusters (2% outliers)	0.916	0.000	0.925	0.000	0.925	0.000	0.928	0.000	<u>0.976</u>	0.000
2 clusters (4% outliers)	0.867	0.000	0.876	0.000	0.876	0.000	0.886	0.000	<u>0.972</u>	0.000
3 clusters (0.5% outliers)	0.978	0.000	0.978	0.000	0.978	0.000	0.978	0.000	<u>0.993</u>	0.000
3 clusters (1% outliers)	0.964	0.000	0.964	0.000	0.964	0.000	0.964	0.000	<u>0.986</u>	0.000
3 clusters (2% outliers)	0.943	0.000	0.943	0.000	0.943	0.000	0.945	0.000	<u>0.985</u>	0.000
3 clusters (4% outliers)	0.907	0.000	0.901	0.000	0.898	0.000	0.911	0.000	<u>0.982</u>	0.000

Bold underlined values indicate best results for each dataset.

Table A2. Comparative analysis of different models (means and standard deviation) based on the adjusted Rand Index (ARI) for 10,000 runs.

Dataset	K-Means		GMM		BGMM		MIDEv1		MIDEv2	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
<i>Synthetic</i>										
Aggregation	0.725	0.008	0.795	0.069	0.860	0.089	0.687	0.035	<u>0.862</u>	0.023
Atom	0.176	0.003	0.058	0.028	0.076	0.024	0.204	0.006	<u>0.221</u>	0.004
D31	<u>0.949</u>	0.016	0.903	0.027	0.634	0.017	0.494	0.037	0.529	0.026
R15	<u>0.993</u>	0.000	0.975	0.036	0.608	0.020	0.747	0.021	0.786	0.018
Gaussians1	1.000	0.000	1.000	0.000	1.000	0.000	1.000	0.000	<u>1.000</u>	1.000
Threernorm	0.032	0.001	0.058	0.045	0.009	0.002	0.088	0.003	<u>0.089</u>	0.002
Twenty	1.000	0.000	0.986	0.028	0.836	0.096	1.000	0.000	<u>1.000</u>	0.000
Wingnut	0.670	0.000	0.862	0.001	<u>0.863</u>	0.000	0.565	0.007	0.533	0.005
<i>Real</i>										
Breast	0.664	0.008	0.772	0.003	0.747	0.003	-	-	-	-
CPU	0.529	0.014	0.315	0.070	0.336	0.081	0.461	0.043	<u>0.708</u>	0.026
Dermatology	0.712	0.038	0.697	0.096	0.728	0.112	-	-	-	-
Diabetes	0.058	0.003	0.059	0.046	0.079	0.028	0.059	0.005	<u>0.086</u>	0.002
Ecoli	0.505	0.008	0.649	0.011	<u>0.665</u>	0.014	0.551	0.013	0.423	0.015
Glass	0.162	0.014	0.178	0.055	0.211	0.040	0.151	0.024	<u>0.229</u>	0.011
Heart-statlog	0.451	0.005	0.352	0.072	0.344	0.075	0.422	0.013	<u>0.452</u>	0.011
Iono	0.168	0.000	0.383	0.066	0.368	0.049	-	-	-	-
Iris	0.617	0.009	0.888	0.077	0.654	0.030	0.819	0.029	<u>0.888</u>	0.008
Wine	0.897	0.000	0.869	0.072	<u>0.932</u>	0.063	0.835	0.031	0.865	0.012
Thyroid	0.583	0.000	<u>0.850</u>	0.075	0.735	0.074	0.297	0.045	0.356	0.015
<i>Generated blobs with outliers</i>										
2 clusters (0.5% outliers)	0.991	0.000	0.990	0.000	0.990	0.000	0.993	0.000	<u>1.000</u>	0.000
2 clusters (1% outliers)	0.976	0.000	0.980	0.000	0.980	0.000	0.980	0.000	<u>0.992</u>	0.000
2 clusters (2% outliers)	0.957	0.000	0.961	0.000	0.961	0.000	0.961	0.000	<u>0.989</u>	0.000
2 clusters (4% outliers)	0.920	0.000	0.924	0.000	0.924	0.000	0.928	0.000	<u>0.990</u>	0.000
3 clusters (0.5% outliers)	0.990	0.000	0.990	0.000	0.990	0.000	0.991	0.000	<u>0.997</u>	0.000
3 clusters (1% outliers)	0.982	0.000	0.982	0.000	0.982	0.000	0.984	0.000	<u>0.993</u>	0.000
3 clusters (2% outliers)	0.967	0.000	0.967	0.000	0.967	0.000	0.967	0.000	<u>0.993</u>	0.000
3 clusters (4% outliers)	0.938	0.000	0.925	0.000	0.918	0.000	0.941	0.000	<u>0.992</u>	0.000

Bold underlined values indicate best results for each dataset.

Table A3. Comparative table of different models (means and standard deviation) based on the J-Score for 10,000 runs.

Dataset	K-Means		GMM		BGMM		MIDEv1		MIDEv2	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
<i>Synthetic</i>										
Aggregation	0.780	0.007	0.800	0.071	0.870	0.062	0.831	0.009	<u>0.871</u>	0.012
Atom	0.556	0.002	0.501	0.004	0.503	0.005	0.575	0.004	<u>0.582</u>	0.004
D31	<u>0.951</u>	0.017	0.901	0.029	0.581	0.019	0.556	0.031	0.609	0.042
R15	<u>0.993</u>	0.000	0.975	0.038	0.664	0.011	0.756	0.041	0.834	0.027
Gaussians1	1.000	0.000	1.000	0.000	1.000	0.000	1.000	0.000	<u>1.000</u>	0.000
Threernorm	0.420	0.001	0.443	0.050	0.381	0.005	0.481	0.003	<u>0.496</u>	0.004
Twenty	1.000	0.000	0.984	0.030	0.838	0.075	<u>1.000</u>	0.002	0.986	0.005
Wingnut	0.834	0.000	0.931	0.001	<u>0.932</u>	0.000	0.779	0.000	0.808	0.000

Table A3. Cont.

Dataset	K-Means		GMM		BGMM		MIDEv1		MIDEv2	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
<i>Real</i>										
Breast	0.833	0.004	0.887	0.001	0.874	0.002	-	-	-	-
CPU	0.656	0.013	0.489	0.058	0.500	0.077	0.733	0.011	<u>0.751</u>	0.010
Dermatology	0.719	0.038	0.699	0.079	0.730	0.106	-	-	-	-
Diabetes	0.252	0.004	0.283	0.033	0.299	0.028	0.275	0.008	<u>0.307</u>	0.004
Ecoli	0.557	0.009	0.655	0.018	0.663	0.006	0.606	0.008	<u>0.663</u>	0.007
Glass	0.340	0.010	0.362	0.036	0.365	0.032	0.397	0.012	<u>0.412</u>	0.009
Heart-statlog	0.720	0.003	0.663	0.043	0.659	0.045	0.714	0.005	<u>0.727</u>	0.004
Iono	0.549	0.000	0.686	0.018	0.673	0.031	-	-	-	-
Iris	0.730	0.008	<u>0.923</u>	0.064	0.752	0.029	0.889	0.012	0.905	0.009
Wine	0.935	0.000	0.917	0.052	<u>0.958</u>	0.046	0.904	0.012	0.917	0.011
Thyroid	0.787	0.000	<u>0.914</u>	0.035	0.856	0.038	0.639	0.007	0.675	0.008
<i>Generated blobs with outliers</i>										
2 clusters (0.5% outliers)	0.993	0.000	0.993	0.000	0.993	0.000	0.993	0.000	<u>1.000</u>	0.000
2 clusters (1% outliers)	0.983	0.000	0.985	0.000	0.985	0.000	0.985	0.000	<u>0.991</u>	0.000
2 clusters (2% outliers)	0.969	0.000	0.971	0.000	0.971	0.000	0.972	0.000	<u>0.994</u>	0.000
2 clusters (4% outliers)	0.942	0.000	0.944	0.000	0.944	0.000	0.946	0.000	<u>0.996</u>	0.000
3 clusters (0.5% outliers)	0.991	0.000	0.991	0.000	0.991	0.000	0.993	0.000	<u>0.998</u>	0.000
3 clusters (1% outliers)	0.983	0.000	0.983	0.000	0.983	0.000	0.985	0.000	<u>0.995</u>	0.000
3 clusters (2% outliers)	0.969	0.000	0.969	0.000	0.969	0.000	0.972	0.000	<u>0.994</u>	0.000
3 clusters (4% outliers)	0.941	0.000	0.932	0.000	0.927	0.000	0.945	0.000	<u>0.993</u>	0.000

Bold underlined values indicate best results for each dataset.

Table A4. Different models were compared (means and standard deviation) based on the Fowlkes–Mallows index (FMI) for 10,000 runs.

Dataset	K-Means		GMM		BGMM		MIDEv1		MIDEv2	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
<i>Synthetic</i>										
Aggregation	0.785	0.006	0.840	0.055	<u>0.891</u>	0.070	0.875	0.011	<u>0.867</u>	0.015
Atom	0.654	0.001	0.653	0.006	0.649	0.003	0.659	0.002	<u>0.669</u>	0.003
D31	<u>0.951</u>	0.015	0.906	0.025	0.681	0.012	0.645	0.011	0.689	0.016
R15	<u>0.993</u>	0.000	0.977	0.033	0.682	0.016	0.779	0.011	0.817	0.009
Gaussians1	1.000	0.000	1.000	0.000	1.000	0.000	1.000	0.000	<u>1.000</u>	0.000
Threenorm	0.518	0.000	0.535	0.030	0.514	0.002	0.552	0.002	<u>0.559</u>	0.003
Twenty	1.000	0.000	0.987	0.026	0.857	0.075	<u>1.000</u>	0.000	0.984	0.004
Wingnut	0.835	0.000	0.931	0.001	<u>0.932</u>	0.000	0.792	0.001	0.764	0.001
<i>Real</i>										
Breast	0.847	0.004	0.893	0.001	0.881	0.001	-	-	-	-
CPU	0.771	0.006	0.619	0.052	0.633	0.065	0.802	0.012	<u>0.871</u>	0.009
Dermatology	0.769	0.030	0.760	0.074	0.784	0.087	-	-	-	-
Diabetes	0.326	0.002	0.382	0.017	0.378	0.028	0.375	0.008	<u>0.389</u>	0.007
Ecoli	0.625	0.006	0.740	0.008	<u>0.762</u>	0.009	0.678	0.006	0.698	0.006
Glass	0.393	0.012	0.435	0.058	0.437	0.048	<u>0.540</u>	0.021	0.519	0.015
Heart-statlog	0.734	0.002	0.683	0.026	0.679	0.028	0.724	0.011	<u>0.737</u>	0.009
Iono	0.601	0.000	0.711	0.004	0.698	0.023	-	-	-	-
Iris	0.743	0.006	<u>0.927</u>	0.041	0.781	0.011	0.899	0.005	0.877	0.005
Wine	0.932	0.000	0.914	0.042	0.955	0.038	0.895	0.011	0.886	0.008
Thyroid	0.841	0.000	0.931	0.023	0.888	0.022	0.705	0.013	0.736	0.009

Table A4. Cont.

Dataset	K-Means		GMM		BGMM		MIDEv1		MIDEv2	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
<i>Generated blobs with outliers</i>										
2 clusters (0.5% outliers)	0.995	0.000	0.995	0.000	0.995	0.000	0.996	0.000	1.000	0.000
2 clusters (1% outliers)	0.988	0.000	0.990	0.000	0.990	0.000	0.990	0.000	0.994	0.000
2 clusters (2% outliers)	0.978	0.000	0.980	0.000	0.980	0.000	0.981	0.000	0.996	0.000
2 clusters (4% outliers)	0.960	0.000	0.961	0.000	0.951	0.000	0.963	0.000	0.995	0.000
3 clusters (0.5% outliers)	0.993	0.000	0.993	0.000	0.993	0.000	0.993	0.000	0.998	0.000
3 clusters (1% outliers)	0.988	0.000	0.988	0.000	0.988	0.000	0.991	0.000	0.996	0.000
3 clusters (2% outliers)	0.978	0.000	0.978	0.000	0.978	0.000	0.981	0.000	0.996	0.000
3 clusters (4% outliers)	0.959	0.000	0.951	0.000	0.948	0.000	0.964	0.000	0.995	0.000

Bold underlined values indicate best results for each dataset.

References

- Ding, S.; Jia, H.; Du, M.; Xue, Y. A semi-supervised approximate spectral clustering algorithm based on HMRF model. *Inf. Sci.* **2018**, *429*, 215–228. [\[CrossRef\]](#)
- Liu, A.-A.; Nie, W.-Z.; Gao, Y.; Su, Y.-T. View-based 3-D model retrieval: A benchmark. *IEEE Trans. Cybern.* **2017**, *48*, 916–928. [\[CrossRef\]](#) [\[PubMed\]](#)
- Nie, W.; Cheng, H.; Su, Y. Modeling temporal information of mitotic for mitotic event detection. *IEEE Trans. Big Data* **2017**, *3*, 458–469. [\[CrossRef\]](#)
- Karim, M.R.; Beyan, O.; Zappa, A.; Costa, I.G.; Rebholz-Schuhmann, D.; Cochez, M.; Decker, S. Deep learning-based clustering approaches for bioinformatics. *Brief. Bioinform.* **2021**, *22*, 393–415. [\[CrossRef\]](#)
- Kim, T.; Chen, I.R.; Lin, Y.; Wang, A.Y.Y.; Yang, J.Y.H.; Yang, P. Impact of similarity metrics on single-cell RNA-seq data clustering. *Brief. Bioinform.* **2019**, *20*, 2316–2326. [\[CrossRef\]](#) [\[PubMed\]](#)
- Govender, P.; Sivakumar, V. Application of k-means and hierarchical clustering techniques for analysis of air pollution: A review (1980–2019). *Atmos. Pollut. Res.* **2020**, *11*, 40–56. [\[CrossRef\]](#)
- Xu, S.; Yang, X.; Yu, H.; Yu, D.-J.; Yang, J.; Tsang, E.C. Multi-label learning with label-specific feature reduction. *Knowl. -Based Syst.* **2016**, *104*, 52–61. [\[CrossRef\]](#)
- Liu, K.; Yang, X.; Yu, H.; Mi, J.; Wang, P.; Chen, X. Rough set based semi-supervised feature selection via ensemble selector. *Knowl. -Based Syst.* **2019**, *165*, 282–296. [\[CrossRef\]](#)
- Wiwie, C.; Baumbach, J.; Röttger, R. Comparing the performance of biomedical clustering methods. *Nat. Methods* **2015**, *12*, 1033–1038. [\[CrossRef\]](#) [\[PubMed\]](#)
- Chen, C.-H. A hybrid intelligent model of analyzing clinical breast cancer data using clustering techniques with feature selection. *Appl. Soft Comput.* **2014**, *20*, 4–14. [\[CrossRef\]](#)
- Polat, K. Classification of Parkinson's disease using feature weighting method on the basis of fuzzy C-means clustering. *Int. J. Syst. Sci.* **2012**, *43*, 597–609. [\[CrossRef\]](#)
- Nilashi, M.; Ibrahim, O.; Ahani, A. Accuracy improvement for predicting Parkinson's disease progression. *Sci. Rep.* **2016**, *6*, 1–18. [\[CrossRef\]](#) [\[PubMed\]](#)
- Trevithick, L.; Painter, J.; Keown, P. Mental health clustering and diagnosis in psychiatric in-patients. *BJPsych Bull.* **2015**, *39*, 119–123. [\[CrossRef\]](#) [\[PubMed\]](#)
- Yilmaz, N.; Inan, O.; Uzer, M.S. A new data preparation method based on clustering algorithms for diagnosis systems of heart and diabetes diseases. *J. Med. Syst.* **2014**, *38*, 48–59. [\[CrossRef\]](#)
- Alashwal, H.; El Halaby, M.; Crouse, J.J.; Abdalla, A.; Moustafa, A.A. The application of unsupervised clustering methods to Alzheimer's disease. *Front. Comput. Neurosci.* **2019**, *13*, 31. [\[CrossRef\]](#)
- Farouk, Y.; Rady, S. Early diagnosis of alzheimer's disease using unsupervised clustering. *Int. J. Intell. Comput. Inf. Sci.* **2020**, *20*, 112–124. [\[CrossRef\]](#)
- Li, D.; Yang, K.; Wong, W.H. Density estimation via discrepancy based adaptive sequential partition. *Adv. Neural Inf. Process. Syst.* **2016**, *29*.
- Rothfuss, J.; Ferreira, F.; Walther, S.; Ulrich, M. Conditional density estimation with neural networks: Best practices and benchmarks. *arXiv* **2019**, arXiv:1903.00954.
- Trentin, E.; Lusnig, L.; Cavalli, F. Parzen neural networks: Fundamentals, properties, and an application to forensic anthropology. *Neural Netw.* **2018**, *97*, 137–151. [\[CrossRef\]](#)
- Trentin, E. Soft-constrained neural networks for nonparametric density estimation. *Neural Process. Lett.* **2018**, *48*, 915–932. [\[CrossRef\]](#)
- Huynh, H.T.; Nguyen, L. Nonparametric maximum likelihood estimation using neural networks. *Pattern Recognit. Lett.* **2020**, *138*, 580–586. [\[CrossRef\]](#)

22. Ruzgas, T.; Lukauskas, M.; Čepkauskas, G. Nonparametric Multivariate Density Estimation: Case Study of Cauchy Mixture Model. *Mathematics* **2021**, *9*, 2717. [[CrossRef](#)]
23. Biernacki, C.; Celeux, G.; Govaert, G. Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate Gaussian mixture models. *Comput. Stat. Data Anal.* **2003**, *41*, 561–575. [[CrossRef](#)]
24. Xu, Q.; Yuan, S.; Huang, T. Multi-dimensional uniform initialization Gaussian mixture model for spar crack quantification under uncertainty. *Sensors* **2021**, *21*, 1283. [[CrossRef](#)]
25. Fraley, C. Algorithms for model-based Gaussian hierarchical clustering. *SIAM J. Sci. Comput.* **1998**, *20*, 270–281. [[CrossRef](#)]
26. Maitra, R. Initializing partition-optimization algorithms. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2009**, *6*, 144–157. [[CrossRef](#)]
27. Meila, M.; Heckerman, D. An experimental comparison of several clustering and initialization methods. *arXiv* **2013**, arXiv:1301.7401.
28. Hasselblad, V. Estimation of parameters for a mixture of normal distributions. *Technometrics* **1966**, *8*, 431–444. [[CrossRef](#)]
29. Behboodian, J. On a mixture of normal distributions. *Biometrika* **1970**, *57*, 215–217. [[CrossRef](#)]
30. Ćwik, J.; Koronacki, J. Multivariate density estimation: A comparative study. *Neural Comput. Appl.* **1997**, *6*, 173–185. [[CrossRef](#)]
31. Tsuda, K.; Akaho, S.; Asai, K. The em algorithm for kernel matrix completion with auxiliary data. *J. Mach. Learn. Res.* **2003**, *4*, 67–81.
32. Lartigue, T.; Durrleman, S.; Allasonnière, S. Deterministic approximate EM algorithm; Application to the Riemann approximation EM and the tempered EM. *Algorithms* **2022**, *15*, 78. [[CrossRef](#)]
33. Dempster, A.P.; Laird, N.M.; Rubin, D.B. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. Ser. B* **1977**, *39*, 1–22.
34. Everitt, B. *Finite Mixture Distributions*; Springer: Berlin/Heidelberg, Germany, 2013.
35. Redner, R.A.; Walker, H.F. Mixture densities, maximum likelihood and the EM algorithm. *SIAM Rev.* **1984**, *26*, 195–239. [[CrossRef](#)]
36. Xie, C.-H.; Chang, J.-Y.; Liu, Y.-J. Estimating the number of components in Gaussian mixture models adaptively for medical image. *Optik* **2013**, *124*, 6216–6221. [[CrossRef](#)]
37. Ahmadinejad, N.; Liu, L. J-Score: A Robust Measure of Clustering Accuracy. *arXiv* **2021**, arXiv:2109.01306.
38. Zhong, S.; Ghosh, J. Generative model-based document clustering: A comparative study. *Knowl. Inf. Syst.* **2005**, *8*, 374–384. [[CrossRef](#)]
39. Lawrence, H.; Phipps, A. Comparing partitions. *J. Classif.* **1985**, *2*, 193–218.
40. Wang, P.; Shi, H.; Yang, X.; Mi, J. Three-way k-means: Integrating k-means and three-way decision. *Int. J. Mach. Learn. Cybern.* **2019**, *10*, 2767–2777. [[CrossRef](#)]
41. Fowlkes, E.B.; Mallows, C.L. A method for comparing two hierarchical clusterings. *J. Am. Stat. Assoc.* **1983**, *78*, 553–569. [[CrossRef](#)]
42. Caliński, T.; Harabasz, J. A dendrite method for cluster analysis. *Commun. Stat. -Theory Methods* **1974**, *3*, 1–27. [[CrossRef](#)]
43. Davies, D.L.; Bouldin, D.W. A cluster separation measure. *IEEE Trans. Pattern Anal. Mach. Intell.* **1979**, 224–227. [[CrossRef](#)]
44. Sun, Y.; Wang, Y.; Wang, J.; Du, W.; Zhou, C. A novel SVC method based on K-means. In Proceedings of the 2008 Second International Conference on Future Generation Communication and Networking, Hainan, China, 13–15 December 2008; pp. 55–58.
45. Hyde, R.; Angelov, P. Data density based clustering. In Proceedings of the 2014 14th UK Workshop on Computational Intelligence (UKCI), Bradford, UK, 8–10 September 2014; pp. 1–7.