mathematics

MDPI

Article

Reduced Clustering Method Based on the Inversion Formula Density Estimation

Mantas Lukauskas * and Tomas Ruzgas

Department of Applied Mathematics, Faculty of Mathematics and Natural Sciences, Kaunas University of Technology, 44249 Kaunas, Lithuania

* Correspondence: mantas.lukauskas@ktu.lt

Abstract: Unsupervised learning is one type of machine learning with an exceptionally high number of applications in various fields. The most popular and best-known group of unsupervised machine learning methods is clustering methods. The main goal of clustering is to find hidden relationships between individual observations. There is great interest in different density estimation methods, especially when there are outliers in the data. Density estimation also can be applied to data clustering methods. This paper presents the extension to the clustering method based on the modified inversion formula density estimation to solve previous method limitations. This new method's extension works within higher dimensions (d > 15) cases, which was the limitation of the previous method. More than 20 data sets are used in comparative data analysis to prove the effectiveness of the developed method improvement. The results showed that the new method extension positively affects the data clustering results. The new reduced clustering method, based on the modified inversion formula density estimation, outperforms popular data clustering methods on test data sets. In cases when the accuracy is not the best, the data clustering accuracy is close to the best models' obtained accuracies. Lower dimensionality data were used to compare the standard clustering based on the inversion formula density estimation method with the extended method. The new modification method has better results than the standard method in all cases, which confirmed the hypothesis about the new method's positive impact on clustering results.

Keywords: nonparametric density estimation; unsupervised machine learning; clustering; inversion formula; dimensions reduction

MSC: 62G05; 62G07; 62G30

Citation: Lukauskas, M.; Ruzgas, T. Reduced Clustering Method Based on the Inversion Formula Density Estimation. *Mathematics* **2023**, *11*, 661. https://doi.org/10.3390/math11030661

Academic Editor: José Antonio Roldán-Nofuentes

Received: 29 December 2022
Revised: 23 January 2023
Accepted: 25 January 2023
Published: 28 January 2023

1. Introduction

Scientists first mentioned artificial intelligence long ago, but in the last decade, it has gained immense popularity, and it can now be used in all areas of life. Machine learning is the most widely used group of mathematical methods, often called artificial intelligence methods, to gain more popularity. Researchers classify machine learning into various types, such as supervised, semi-supervised, reinforcement, and unsupervised learning. This paper focuses explicitly on unsupervised learning methods. Unsupervised machine learning is one type of machine learning with an exceptionally high application in various fields. The most popular and best-known group of unsupervised machine learning methods is clustering methods used in cluster analysis. In 1932, Driver and Kroeber first mentioned cluster analysis, which noticeably was mentioned earlier than artificial intelligence. The main goal of cluster analysis is to find hidden relationships between individual observations. These observations can be the company's customers, products, and goods, as well as doctors' patients and other data. The main aim is to divide observations into groups that the researcher does not immediately know. Data clustering has applications in different fields: healthcare for detection of breast cancer [1], Alzheimer's

disease [2,3], and others. Clustering can also be applied in pattern recognition [4,5] and big data text clustering [6,7]. Clustering methods often use different distance measures to determine how close individual observations are to each other and determine the best clustering. In current studies, the k-means clustering method is the most used, but there are also many different clustering methods. Various clustering methods are used to solve various practical problems. However, although the choice of these methods is wide, this problem remains challenging and new methods that can solve clustering problems are constantly being sought. It can also be noted that much attention is paid to developing various density estimation methods that can be used to solve the problem of data clustering. There is also an ongoing search for ways to estimate density if there are outliers in the data. Various methods based on neural targets have been proposed to solve this problem: soft-constrained neural networks [8] and Parzen neural networks [8]. This work is also based on data clustering based on density estimation. This research paper is a follow-up work of a more extensive study on developing a new data clustering method [9]. A previous research paper observed that the newly developed clustering method based on the modified inversion formula density estimation (CBMIDE) performs poorly when the data dimensions are higher ($d > 10$). In this paper, as a novelty, we suggest modification of the CBMIDE clustering method for higher-dimension data to allow the method to work in higher dimensions; moreover, to increase the previous method's accuracy in the lower-dimensionality case. For this reason, this research paper examines the impact of data dimensionality reduction modification on the method. Data dimensions are essential for the accuracy of clustering methods, calculation time, and using different computing resources. In a typical case, it is observed that with increasing dimension, the time of data clustering also increases significantly and can even increase exponentially. In such a case, one solution is to reduce the dimensionality of the data. It can be done using different dimensionality reduction methods. Combining data dimensionality reduction with clustering is quite common because it saves many resources. Data dimensionality reduction and the application of reduced dimensions in clustering are also discussed in the scientific literature, where the simplest k-means and principal component analysis (PCA) methods were first combined [10]. These studies are popular even now. The main reason for this is the increasing amount of data every year. A relatively popular field of combining these methods is gene analysis. Data dimensionalities reduction methods such as principal component analysis (PCA) [11], non-negative matrix factorization (NMF) [12], independent component analysis (ICA) [13] and clustering methods such as k-means, density-based spatial clustering of applications with noise (DBSCAN), and others used to study gene sequences [14]. Furthermore, more complex methods are noticed to reduce the dimensions of the data, such as t-distributed stochastic neighbour embedding (t-SNE) [15], uniform manifold approximation and projection (UMAP) [16], and different combinations of methods [17–22].

In this paper, we modify the CBMIDE method and test our hypothesis about the impact of dimensionality reduction on clustering results. The paper hypothesizes that the accuracy of data clustering can be maintained by reducing data dimensions using data dimensionality reduction methods and that the new RCBMIDE method has the advantage compared with an earlier method and other popular methods. This paper uses a much larger number of data dimensionality reduction methods and clustering methods to compare results.

This paper is organized as follows. In the second section of this paper, we present clustering based on the modified inversion formula density estimation and a reduced version of this method. The third section of the paper contains information about the methodology, methods used in the comparative analysis, data sets, and others. The fourth section of the paper reviews the results obtained from the research study. Finally, the last section of this paper presents a discussion of research and research results, conclusions, and future work.

2. Clustering Based on the Density of the Modified Inversion Formula

Data clustering is possible based on density estimation functions. In statistical modelling, estimating probability density functions (pdf) is one of the most critical parts. Probability density functions make it possible to describe random variables as hidden functions of other variables and simultaneously identify hidden relationships, which can be used to create data groups. For example, if we claim that the random vector $X \in \mathbf{R}^d$ satisfies the distribution mixture model if the distribution density $f(x)$ satisfies Equation (1).

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

In the given formula (1), $f_k(x)$—distribution density, θ—multivariate model parameter, and p_k—the probability of the k-th element. It is worth noting that when evaluating a mixture of distributions, the number of clusters q and a priori probability p_k is used, and the following conditions must be met:

$$p_k > 0, \sum_{k=1}^{q} p_k = 1. \tag{2}$$

Suppose X is a d-dimensional random vector with a distribution density f(x), and there is a sample of independent copies of X, where $X = (X(1), \ldots, X(n))$. We will say that the sample satisfies the mixture model if X(t) satisfies (1). We will call size n the sample size (volume). When examining the approximations of parametric methods, it should be emphasized that as the data dimension increases, the number of model parameters grows rapidly, which makes it more difficult to find accurate parameter estimates. One-dimensional data projections $X_\tau = \tau' X$ density f_τ is much easier to find than multidimensional data density f because a mutually unambiguous correspondence exists, $f \leftrightarrow \left\{ f_\tau,\ \tau \in R^d \right\}$. The projection density f_τ of one-dimensional data is much easier to find than the density f of multidimensional data. If the distributions used are Gaussian, then the one-dimensional Gaussian mixture model can be described by the following formula.

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

here $\varphi_{k,\tau}(x) = \varphi\left(x;\ m_{k,\tau},\ \sigma^2_{k,\tau}\right)$—Gaussian density and multidimensional parameter depends on the given parameters $\theta_\tau = \left(p_{k,\tau},\ m_{k,\tau},\ \sigma^2_{k,\tau}\right), k = 1, \ldots, q$. The following equations relate these parameters $p_{j,\tau} = p_j$, $m_{j,\tau} = \tau' M_j$, and $\sigma^2_{j,\,\tau} = \tau' R_j \tau$. Then we can use the inversion formula (4) given below.

$$f(x) = \frac{1}{(2\pi)^d} \int_{\mathbf{R}^d} e^{-it'x} \psi(t) dt, \tag{4}$$

where $\psi(t) = \mathrm{E} e^{it'x}$ denotes the characteristic function of the random variable X. To use the inversion formula, first of all, we select the number of design directions, which must be evenly distributed on the unit sphere, the set T, and by changing the characteristic function by its estimate, we therefore obtain the estimating formula for the density calculation [23,24]:

$$\hat{f}(x) = \frac{A(d)}{\#T} \sum_{\tau \in T} \int_0^\infty e^{-iu\tau' x} \hat{\psi}_\tau(u) u^{d-1} e^{-hu^2} du, \tag{5}$$

where # denotes the number of elements in the array. It is possible to calculate the constant A(d), which depends on the dimension of the data, using the formula for the volume of a d-dimensional sphere.

$$A(d) = \frac{(V_d(1))_R^{'}}{(2\pi)^d} = \frac{d2^{-d}\pi^{-\frac{d}{2}}}{\Gamma\left(\frac{d}{2}+1\right)}. \quad (6)$$

Formula (5) can be used for various estimates of the characteristic function of the projected data. One of the disadvantages of the method using the inversion formula defined in (5) is that the mixture model of Gaussian distributions described by this estimate (when $f_k = \varphi_k$) only estimates the density well of observations of the distribution close to it. The density estimation using a mixture of Gaussian distributions can become complicated when approximating the density under study due to many components with low a priori probabilities. To overcome this, we can introduce a noise cluster and use a modified algorithm based on a multidimensional Gaussian distribution mixture model. One such algorithm is the inversion formula (4). Using this formula, we can calculate the parametric estimate of the characteristic function of uniform distribution density.

$$\hat{\psi}(u) = \frac{2}{(b-a)u} \sin\frac{(b-a)u}{2} \cdot e^{\frac{iu(a+b)}{2}}. \quad (7)$$

The uniform distribution density function (7) defines b as the maximum value and a as the minimum value. In the density estimate calculation formula (5), we construct the estimation of the characteristic function by combining the characteristic functions of a mixture of Gaussian distributions and a uniform distribution, using the corresponding a priori probabilities. Then the calculated continuous characteristic function presented earlier and the characteristic function of Gaussian distributions are used to form the density estimate. A priori probabilities are used to calculate the density estimate, which allows for controlling the influence of outliers on the density estimate. In formula (7), the second component of the density estimate is evaluated as a noise cluster whose probability/weight is $\hat{p}_0$.

$$\hat{\psi}_\tau(u) = \sum_{k=1}^{\hat{q}_\tau} \hat{p}_{k,\tau} e^{iu\hat{m}_{k,\tau} - u^2\hat{\sigma}^2_{k,\tau}/2} + \hat{p}_{0,\tau}\frac{2}{(b-a)u}\sin\frac{(b-a)u}{2}\cdot e^{\frac{iu(a+b)}{2}}, \quad (8)$$

The EM algorithm is used to apply this density estimate to data clustering. The expectation-maximization (EM) algorithm can be used to estimate the parameters of a mixture model, which is a probabilistic model for representing a dataset as a mixture of multiple distributions. A mixture model is defined by a set of parameters, including the individual components' means, variances, and mixing proportions. E-step (expectation step): For each data point, the algorithm calculates the probability that the point belongs to each component in the mixture based on the current estimates of the model parameters. Model parameters estimation is done by computing the likelihood of the data point given each component and multiplying it by the mixing proportion of that component. M-step (maximization step): In this step, the algorithm updates the model parameters to maximize the expected likelihood of the data based on the probabilities computed in the E-step. Specifically, the components' means, variances, and mixing proportions are updated to maximize the likelihood of the data under the current assignment of data points to components. The algorithm alternates between the E-step and the M-step until convergence is reached, at which point the estimated parameters are considered the maximum likelihood estimates. In the first step of the clustering algorithm, initial parameters selection using the initialization of k-means is performed. In the scientific literature, the random selection of parameters is used quite often [25,26]. However, this parameter selection showed that, in this case, the clustering method has worse stability. Furthermore, another way to choose the initial parameters is to use hierarchical clustering to combine the data [27]. However, this parameter initialization also failed in the case of the method being created.

For this reason, it was decided to use a more stable initialization of k-means parameters, which is used for other density-based clustering methods, such as the Gaussian mixture model and the Bayesian Gaussian mixture model (in package scikit-learn). The EM algorithm is famous in scientific works [28–30]. Based on the EM algorithm, the parameters of the data clustering method calculated after the number of r cycles are $\hat{\pi}_k = \hat{\pi}_k^{(r)}$. Then the new estimate of the multivariate parameter is $\hat{\theta} = \hat{\theta}^{(r+1)}$, whose individual components (probability, mean matrix, and covariance matrix, respectively) are calculated according to the formulas below.

$$\hat{p}_k = \frac{1}{n}\sum_{t=1}^{n} \hat{\pi}_k(X(t)) \tag{9}$$

$$\hat{M}(k) = \frac{1}{n\hat{p}_k}\sum_{t=1}^{n} \hat{\pi}_k(X(t)) \cdot X(t) \tag{10}$$

$$\hat{R}(k) = \frac{1}{n\hat{p}_k}\sum_{t=1}^{n} \hat{\pi}_k(X(t))\left[X(t) - \hat{M}(k)\right] \cdot \left[X(t) - \hat{M}(k)\right]' \tag{11}$$

where $k = 1, \ldots, q$. Using the estimate of θ, the estimates of the probabilities π_k are obtained by replacing the unknown parameters with their statistical estimates from the formula

$$\pi_k(x) = \frac{p_k f_k(x)}{f(x,\theta)} \text{ and } k = \overline{1,q} \tag{12}$$

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

$$\pi_k(x) = \mathbf{P}\{v = k \mid X = x\} \tag{13}$$

evaluation when all $x \in \{X(1), \ldots, X(n)\}$. Strict sample clustering would be the evaluation of random variables $v(1), \ldots, v(n)$, i.e., the sample is divided into subsets based on equality

$$\hat{v}(t) = arg\max_{k=1,\ldots,q} \hat{\pi}_k(X(t)) \tag{14}$$

The estimates $\hat{\pi}_k$ are obtained by approximating the unknown components of the distribution density with density estimates from the inversion formula and using the EM (expectation maximization) algorithm.

The following Algorithm 1 can describe the generalized clustering of data based on modified inversion formula density estimation (CBMIDE).

Algorithm 1: CBMIDE clustering algorithm

Input: Data set $X = [X_1, X_2, \ldots, X_n]$, cluster number K
Output: $C_1, C_2, \_\_, C_t$
Initiation of the mean vector using the k-means method
Generate a T matrix. The set T calculation where design directions are evenly spaced on the sphere.
1 For i = 1: t do
2 Density estimation for each point and cluster with the formula (8)
Update $\hat{M}$, $\hat{p}_k$, $\hat{R}$
3 End
4 Return $C_1, C_2, \_, C_t$ and $\hat{M}$, $\hat{p}_k$, $\hat{R}$

The biggest drawback of the earlier proposed method is its use for large-dimensional data. To use the inversion formula, first of all, we select the number of the design direction, which must be evenly distributed on the unit sphere, the set T. As the dimensions increase, finding design directions on a unit sphere becomes difficult. Therefore, it is not easy to generate the required matrix T. For this reason, an extension of the CBMIDE method is needed, allowing the use of this method for large-dimensional data. The data dimensionality reduction applied in the first clustering step is an extension of the method proposed in this paper.

$$Z = f(X,\ d) \tag{15}$$

where Z—reduced dimensions data in latent space, X—initial data in the original space, and d—number of dimensions. Further clustering of the data is then performed using the reduced dimensionality data. Dimensional data reduction allows for the separation of redundant characteristics of the observations, reducing the problem of multicollinearity in the data. Likewise, data dimensionality reduction allows easier data visualization when dimensions are reduced to two or three dimensions that can be represented visually. Data dimensionality reduction can be performed using feature selection or methods that form new dimensions that best represent the original data. Multiple data reduction techniques were implemented into the new algorithm to extend the original clustering method based on the inversion formula algorithm. To analyse and compare the results of these methods: principal component analysis (PCA), independent component analysis (ICA), factor analysis (FA), T-distributed stochastic neighbour embedding (t-SNE), TriMap, uniform manifold approximation and projection (UMAP), ISOMAP, multidimensional scaling (MDS), and locally linear embedding (LLE) were used. Different variants are used to determine which of the modifications has the best clustering results by modifying the general data clustering method based on the modified density estimation of the inversion formula. In the first modification of the algorithm, we use principal component analysis to reduce data dimensions into desired latent space. One of the most widely used data dimensionality reduction methods is principal component analysis (PCA) [31].

Let us say we have a multidimensional data matrix X, then we aim to calculate the correlation matrix R, and then the covariance matrix C is also created. If the individual variables are not covariate, then their covariance coefficient equals 0. To find the main components, the covariance matrix's real vectors E_k and real values λ_k are found, which are in the following equation

$$CE_k = \lambda_k E_k \tag{16}$$

solutions. In the above equation, E_k is the column, C is the previously defined covariance matrix, and the real vector λ_k is found in the characteristic equation $|C - \lambda_k I| = 0$, where I is the identity matrix.

After sorting the eigenvectors E_k according to the values of the corresponding true values in descending order, a matrix of principal components is formed as $A = (E_1, E_2, \ldots, E_n)$. Then the transformation of the data into the latent space is performed according to the formula:

$$Z_i = \left(X_i - \overline{X}\right) A_d \tag{17}$$

where Z_i—column of latent space, X_i—column of the original space, d—selected number of dimensions, and i—column number.

The second modification to improve the algorithm is the application of independent components for dimensionality reduction. The main difference between principal component analysis (PCA) and independent component analysis (ICA) is that PCA relies on uncorrelated factors while ICA relies on independent [32].

The third modification used in this paper is factor analysis. Factor analysis is a method that considers the correlations of variables, and the aim is to find latent variables that describe the original variables [33]. Suppose that there are k variables $X_1 \ldots X_k$, and we seek to determine the factors that describe these variables, the number of which is m, then, the mathematical model of factor analysis can be summarized as

$$X_k = \lambda_{k1}F_1 + \lambda_{k2}F_2 + \ldots + \lambda_{km}F_m + \varepsilon_k \tag{18}$$

Multipliers λ_i are called factor weights. The factor analysis problem is the inverse of the linear regression problem, i.e., we know X_k values, and we want to find out what can be said about the common factors F_m.

The fourth modification of the CBMIDE method is to reduce the dimensionality of the t-SNE data by constructing latent dimensions. T-distributed stochastic neighbour embedding (t-SNE) is a method that is often intended for the visualization of multidimensional data after reducing this data to a more convenient two-dimensional or three-dimensional space [34]. This method is performed using two main steps. In the first step, a probabilistic calculation is performed. If we have X as our dataset, we try to compute probabilities p_{ij} based on the formula

$$p_{i|j} = \frac{\exp(-||x_i - x_j||_2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-||x_i - x_k||_2/2\sigma_i^2)} \tag{19}$$

One of the most important things is that $p_{i|i} = 0$ and $\sum\limits_{i,j} p_{i,j} = 1$. Then, in the second step, Kullback–Leibler error (KL) optimization is performed, thus aiming to determine the exact locations of the observations in a smaller space. T-SNE aims to create d-dimensional data, so the similarity between two points located in the reduced dimension z_i and z_j, is calculated, which can be written in the given formula

$$q_{i,j} = \frac{\left(1 + ||z_i - z_j||^2\right)^{-1}}{\sum_k \sum_{l \neq k} \left(1 + ||z_k - z_l||^2\right)^{-1}} \tag{20}$$

Here, the Cauchy distribution is used to determine the similarity between observations. Then, to find the most suitable reduced data matrix, the previously mentioned Kullback–Leibler error (KL) is used, which can be defined as:

$$KL(P||Q) = \sum_{i \neq j} p_{i,j} log \frac{p_{i,j}}{q_{i,j}} \tag{21}$$

The gradient descent is used to do this. The Euclidean distance is mainly used in this method; therefore, this distance is also used in this paper. The T-SNE data reduction method is used quite often, so the inclusion of this method in the study is strongly justified [35–37]. The perplexity parameter of the t-SNE method used in the study is 30, the distance is Euclidean, and the method used is Barnes–Hut.

The fifth, sixth, and seventh modifications of the method are similar to the previously discussed modification of the t-SNE method since methods of the same family perform data dimensionality reduction. UMAP (uniform manifold approximation and projection) is a data dimensionality reduction technique such as t-SNE. Embedding is found by finding a low-dimensional data projection with the closest possible equivalent fuzzy topological structure. TriMap is a dimensionality reduction method that uses a triple constraint to form an embedding of a low-dimensional set of points. TriMap provides a significantly better global representation of data than other dimensionality reduction methods such as t-SNE, LargeVis, and UMAP [38]. The ISOMAP method calculates a quasi-isometric, low-dimensional embedding that best represents the original data set.

The eighth and ninth modifications of the method are performed using the MDS and LLE methods. The MDS method uses similarity or dissimilarity matrices to identify the nearest neighbours. Given a proximity matrix with proximities between each pair of objects in a set and a selected number of dimensions N, the MDS algorithm places each object in N-dimensional space (lower-dimensional representation) so that the distances between objects are preserved as best as possible. Locally linear embedding (LLE) tries to reduce these dimensions by trying to preserve the geometric features of the original non-linear structure [39]. The LLE first finds the k-nearest neighbours of points. Then it approximates each data vector as a weighted linear combination of k nearest neighbours. Finally, it calculates the weights that best restore the vectors from the neighbours and then produces the low-dimensional vectors best restored by these weights. One of the advantages of the LLE algorithm is that only one parameter needs to be tuned, which is the value of k or the number of nearest neighbours considered part of a cluster. Furthermore, the following methods are used as further relevant modifications: Spectral embedding and kernel PCA. Spectral embedding is a technique used in machine learning and data analysis to represent data points in a high-dimensional space in a lower-dimensional space, typically for visualization or dimensionality reduction. Representation is achieved by constructing a graph representation of the data points and using the eigenvectors of the graph's Laplacian matrix to define the embedding. The resulting embedding preserves the pairwise distances between data points up to a scaling factor and can be used for various tasks such as clustering, classification, and visualization. Spectral embedding has been widely applied in various fields, including computer vision, natural language processing, and social network analysis, due to its ability to reveal the underlying structure of the data and facilitate downstream tasks. Kernel principal component analysis (kernel PCA) is a non-linear dimensionality reduction technique that extends the traditional PCA by projecting the data points onto a higher-dimensional feature space, where the data is linearly separable. Linear separation is achieved using a kernel function to compute the dot product of the data points in the feature space. The resulting embedding preserves the pairwise distances between data points up to a scaling factor and can be used for various tasks such as clustering, classification, and visualization. Kernel PCA has been widely applied in various fields, including computer vision, natural language processing, and genomics, due to its ability to capture non-linear patterns in the data and facilitate downstream tasks. However, kernel PCA suffers from high computational complexity, as it requires the computation of the kernel matrix, which has a quadratic time complexity concerning the number of data points. Complexity makes kernel PCA impractical for large datasets.

One possible metric for successful data dimensionality reduction is trustworthiness. The trustworthiness metric is used to evaluate the success of the data dimensionality reduction methods used in the study [40]. This metric is calculated using the following formula.

$$T(k) = 1 - \frac{2}{nk(2n - 3k - 1)} \sum_{i=1}^{n} \sum_{j \in N_i^k} \max(0,\ (r(i,j) - k)) \tag{22}$$

where for each observation i, N_i^k are its k nearest neighbours (the output space obtained after applying the methods), and for each observation j, $r(i,j)$ is its original data space. If an unintended observation is observed in the output space, it is penalized. The number of neighbours used in the study is 5. The generalized algorithm of the extended CBMIDE is shown in the Algorithm 2.

Algorithm 2: RCBMIDE clustering algorithm

Input: Data set X= [X_1, X_2, . . . , X_n], cluster number K, the smoothness parameter h, and the percentage ratio of outliers p_0, number of iterations t, number of dimensions d, data reduction method.
Output: C_1, C_2, _ _, C
Input data reduction:
For j= 2: d < 15 do
Reduce dimension with the dimensionality reduction method
Calculate the trustworthiness of the reduced dimensions with (13)
Choose the best-reduced dimensions data based on trustworthiness.
Initiation of the mean vector using the k-means and k-means++ method
Generate a T matrix. The set T calculation where directions are evenly spaced on the sphere.
1 For i = 1: t do
2 Density estimation for each point and cluster
Update $\hat{M}$, $\hat{p}_k$, $\hat{R}$ values
3 End
4 Return C_1, C_2, _, C_t and $\hat{M}$, $\hat{p}_k$, $\hat{R}$

3. Materials and Methods

This section provides information on materials and methods. The first part of the section discusses the clustering evaluation metrics used in the research. Then, research data sets and data preparation for the research. Finally, the experimental setup was used in the study.

3.1. Clustering Evaluation Metrics

Clustering results can be evaluated differently depending on whether the true values are known. In the real-world case, the true values are unknown, so evaluation metrics such as Calinski–Harabasz [41], Davies–Bouldin [42], or the silhouette coefficient must be used. Meanwhile, in this study, the true values of the clusters are known, whereas synthetic data sets are used. For this reason, other performance evaluation metrics can be used: J score [43], normalized mutual information (NMI) [44], adjusted rand index (ARI) [45], accuracy (ACC) [46], and Fowlkes–Mallows index (FMI) [47]. In this study, the primary metric for evaluating data clustering is ACC. Furthermore, if the accuracy is close to or equal, the clustering results are evaluated by NMI, ARI, and other metrics mentioned previously. With the metric values remaining the same, the created database also stores more metrics to evaluate the clustering results, such as completeness and homogeneity scores. Additionally, visual representation of two-dimensional and three-dimensional data to visually evaluate the clustering results.

Data clustering accuracy is evaluated based on the sum of the diagonal elements of the confusion matrix, divided by the number of samples to obtain a value between 0 and 1. This work considers this accuracy the only primary metric by which it is evaluated for final clustering success.

$$ACC = \frac{1}{N}\sum_{i=1}^{k} n_i \tag{23}$$

The N is the total number of observations, n_i is the number of data points correctly divided into the corresponding cluster i, and k is the cluster number.

3.2. Research Datasets

This subsection provides basic information about the data used in the study. This study uses more than 20 popular data sets from other scientific studies to evaluate clustering. Table 1 provides basic information about the data. The following methods are used for data preparation/normalization: Raw, MinMax, Standard, Robust, Max-Abs, QuantileNormal, QuantileUniform, PowerTransformer, and Normalizer. It is worth noting that raw data carries a high risk of possible differences between variables. However, the inclusion of

these data allows evaluation of the influence of different data preparation methods on the results of data clustering. For example, the MinMax scaler's data is compressed to a (0,1) scale based on the maximum and minimum values.

$$X_{std} = \frac{(X - X_{min})}{(X_{max} - X_{min})} \tag{24}$$

In the case of the standard scaler, the data is standardized by removing the mean and scaling to unit variance.

$$X_{std} = \frac{X - \overline{X}}{s} \tag{25}$$

The robust scaler removes the mean and scales the data by the quantile range (default IQR: interquartile range). The IQR is the interval between the first quartile (25th quantile) and the third (75th) quartile).

$$X_{std} = \frac{X - Q_1(X)}{Q_3(X) - Q_1(X)} \tag{26}$$

Table 1. A description of the data set used.

ID	Data Sets	Sample Size (N)	Dimensions (D)	Classes
1	Balance-scale	625	4	3
2	Arrhythmia	452	262	13
3	atom	800	3	2
4	Breast	570	30	2
5	Coil20	1440	1024	20
6	CPU	209	6	4
7	Dermatology	366	17	6
8	Diabetes	442	10	4
9	Ecoli	336	7	8
10	German	1000	60	2
11	Glass	214	9	6
12	Haberman	306	3	2
13	Heart-statlog	270	13	2
14	Iono	351	34	2
15	Iris	150	4	3
16	pmf	649	3	5
17	segment	2310	19	7
18	spambase	4601	57	2
19	Thyroid	215	5	3
20	wdbc	569	30	2
21	Wine	178	13	3

3.3. Experimental Setup

This subsection reviews the experimental setup of this research. Considering that the research uses many different data dimension reduction methods, scaler methods, and clustering methods, an appropriate experimental setup is required. First, data preparation and dimension reduction are performed on a Linux server (30 CPUs, A30 24 GB GPU). Then, with the help of DVC, the data is versioned and transferred to other machines used in the study. All research results are stored in a PostgreSQL database, which allows nonrepetitive calculations to be performed on multiple machines simultaneously. In addition, Grafana is available for data visualization and Prometheus for error logging. Calculations are performed using five separate machines: three servers and two computers. The parameters of these machines: 1S—30 CPUs, 64 GB RAM; 2S—8 CPUs 128 GB RAM, 3S—16 CPUs, 64 GB RAM; 1C—Intel Core i7-8750H (6 cores, 12 logical processors), 32 GB RAM, 2060 6 GB GPU; and 2C—Intel Core i7-12700 K (12 cores, 20 threads), 32 GB RAM, NVIDIA 3080 Ti 12 GB GPU. The calculations of combinations of methods can be performed in parallel,

and then the results are stored in a database table. Different clustering methods are used to provide different parameters, so the parameters of each method are selected separately. A predefined rule is used for parameter selection: Each parameter has at least 20 steps. More than 65 million models have been created and saved in the current database.

4. Results

This section presents the main results of the research and compares reduced clustering based on the modified inversion formula density estimation (RCBMIDE) with other clustering methods. The notation used in the following results is as follows: Agg—agglomerative clustering, GMM—Gaussian mixture model, and BGMM—Bayesian Gaussian mixture; the full names of other methods are given. A rich set of different parameters is used for data clustering. Using the Agg method, the affinity values used were Euclidean, Manhattan, and cosine, and the linkage values used were ward, complete, average, and single. In the case of the BIRCH method, two main parameters are changed: Threshold and branching factor. The threshold was changed from 0.05 to 1 and branching factor from 2 to 10. The k-means method initializes the GMM, BGMM, CBMIDE, and RCBMIDE centres. The CBMIDE and RCBMIDE methods also changed the smoothness parameter h from 0 to 0.5 and the size of the generated matrix T from 10 to 1000 observations. The DBSCAN and HDBSCAN methods used Euclidean, Manhattan, and Chebyshev distances and changed the values of eps and min samples. The minimum and maximum limits of the eps values were determined by calculating the closest and farthest distance between points in the data set: the Min sample value changes from 1 to $N/2$, where N is the number of data samples. The database currently contains more than 60 million models (including models using dimensionality reduction), and the current best-performing models are shown in the subsections of this section. In the case of each model, different scalers are also used, which allows one to find the most optimal model. The best clustering methods differ for different datasets, and there is no single best method. It is also worth noting that the CBMIDE method cannot work for some datasets due to many dimensions.

4.1. Performances of Reduced Clustering Based on the Modified Inversion Density Estimation for Lower Dimensions Datasets

This subsection reviews the results obtained with lower-dimensional data sets. This section is designed to compare the original CBMIDE method and the modified RCBMIDE method. Given that the original CBMIDE method suffers in high dimensions, for this reason, the comparison can only be made for lower dimensional datasets. Based on the results obtained (see Table 2), it can be observed that the accuracy of the data clustering using the newly modified reduced CBMIDE method (RCBMIDE) provides better clustering results than the original CBMIDE method. These results prove our hypothesis about reduced method application for data clustering and result improvement compared with the original method. It can also be noticed that for five data sets, the best results are obtained using the RCBMIDE method compared with other popular methods. In other cases, the accuracy results are close to the best clustering method for the corresponding data set, which proves the idea of reduced/modified method application in data clustering. Talking about modifications, best results were achieved mainly with the sixth modification (TriMap) and fifth modification (UMAP). For complete best modifications in each dataset, see Table A2.

Table 2. Different clustering methods results for lower dimensions datasets.

Dataset	Agg	BIRCH	GMM	BGMM	DBSCAN	K-Means	HDBSCAN	CBMIDE	RCBMIDE
1balance-scale	0.624	0.658	0.568	0.586	0.464	0.603	0.597	0.576	**0.693**
atom	**1.000**	0.868	0.883	0.960	**1.000**	0.719	**1.000**	0.891	**1.000**
cpu	0.823	0.828	0.746	0.641	0.833	0.761	0.813	0.815	**0.858**
diabetes	0.507	**0.514**	0.459	0.455	0.482	0.428	0.477	0.502	0.512
ecoli	0.804	**0.845**	0.762	0.747	0.682	0.688	0.646	0.754	0.817
glass	0.514	0.565	0.509	0.528	0.514	0.547	0.528	0.527	**0.607**
Haberman	0.748	**0.761**	0.667	0.716	0.758	0.748	0.739	0.735	0.742
iris	0.967	0.973	0.967	0.893	0.94	0.967	0.700	0.975	**0.983**
pmf	0.977	0.977	0.92	0.977	**0.983**	0.844	0.978	0.934	0.981
thyroid	0.93	0.949	**0.963**	0.949	0.874	0.944	0.823	0.778	0.834
Wine	0.978	**0.994**	0.972	0.983	0.949	0.978	0.876	0.953	0.963

Values in bold and underlined indicate the best results for each dataset.

4.2. Performances of Reduced Clustering Based on the Modified Inversion Density Estimation for Higher-Dimensional Datasets

This subsection reviews the results obtained using datasets with more dimensions. It is important to note that CBMIDE does not work for large data dimensions, so it is impossible to include it in this comparison. Based on these results (see Table 3), it can be said that the proposed extension of the method is valid since, in this case, it is possible to apply the clustering method based on the density estimation of the inversion formula. The results showed that the extended RCBMIDE method for specific data sets has the highest clustering accuracy compared to other existing clustering methods. For example, the best results were achieved on german and segment datasets. For five more datasets, the achieved results are second best in the comparison. For the german dataset, the best method modification is the eighth modification (multidimensional scale (MDS)), and for the segment dataset, the best results were achieved using the sixth modification (TriMap). For complete best modifications in each dataset, see Table A1.

Table 3. Different clustering methods results for higher dimensions datasets.

Dataset	Agg	BIRCH	GMM	BGMM	DBSCAN	K-Means	HDBSCAN	RCBMIDE
arrhythmia	**0.600**	0.571	0.485	0.431	0.573	0.438	0.582	0.597
Breast	0.942	**0.954**	0.951	0.953	0.903	0.928	0.743	0.909
Coil20	0.738	0.738	0.638	0.675	0.867	0.733	**0.884**	0.882
dermatology	0.956	**0.978**	0.91	0.855	0.694	0.962	0.809	0.867
german	0.705	0.713	0.696	0.638	0.712	0.673	0.704	**0.716**
heart-statlog	0.807	0.811	0.819	0.826	0.815	**0.848**	0.626	0.831
iono	0.729	0.795	0.849	0.809	**0.929**	0.712	0.906	0.899
segment	0.708	0.732	0.631	0.612	0.529	0.665	0.530	**0.789**
spambase	0.878	**0.918**	0.856	0.857	0.693	0.854	0.690	0.905
wdbc	0.942	0.954	0.951	**0.958**	0.903	0.928	0.743	0.951

Values in bold and underlined indicate the best results for each dataset.

5. Discussion

This paper reviews a data clustering method based on modified inversion formula density estimation (CBMIDE) and its extension. It uses data dimensionality reduction—reduced clustering based on the modified inversion formula density estimation (RCBMIDE). This extension of the data clustering method is carried out because it was observed that the CBMIDE method does not work for higher dimensions cases. For this reason, this paper proposes an extension of the method by including data dimensionality reduction in the data clustering algorithm. The application of data dimensionality reduction makes it possible to apply the improved method to higher dimensions ($d > 15$). For the extension, data reduction methods, such as principal component analysis, independent component analysis, factor analysis, and ISOMAP, were used to reduce data dimensions. In addition, various methods

of reducing data dimensions and dimensions were used. It allowed us to evaluate the possible data impact of compression on the clustering accuracy. The results showed that the new method extension positively affects the data clustering results. Compared to other popular data clustering methods, the newly constructed RCBMIDE method works well and achieves the best accuracy in many cases. When the accuracy is not the best, the data clustering results are close to those obtained by the best models. Furthermore, an experimental evaluation was made of the data of lower dimensions. This comparison aims to compare this with the CBMIDE method without the extension. The results showed that the new modification method has better results than the usual CBMIDE method in all cases. Therefore, we can say that the hypothesis about method modification's impact on better clustering results was proved. It is also worth noting that the results showed that the data dimensionality reduction methods used in all cases are different, so it is not easy to decide which method works best. The trustworthiness metric allowed us to compare how well the dimensionality reduction was performed and the effectiveness of the clustering after that. The results showed that the UMAP and TriMap methods, which are methods of the same family, usually have the best results for lower dimensional data. It can be said that this paper proved the hypothesis about the extension of the developed method to large-dimensional data applications in data clustering. Further research will be conducted based on this work. The future direction of these studies and additional studies will be carried out using a more significant number of data clustering methods, more data sets, and dimensionality reduction methods. Further studies also aim to evaluate the influence of individual method parameters on the results of data clustering, not only by isolating the best models but by evaluating these models in more detail. Furthermore, this study is the beginning of further studies in which the main focus will be deep clustering. The development of the deep clustering method will be based on the density estimation of the modified inversion formula.

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. and 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.

Data Availability Statement: Not applicable.

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

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

Appendix A

Table A1. RCBMIDE method extension (dimensionality reduction) for each higher-dimensions dataset.

Dataset	RCBMIDE
arrhythmia	TriMap (5)
breast	UMAP (3)
Coil20	LLE (3)
dermatology	NMF (2)
german	MDS (3)
heart-statlog	CosinePCA (4)
iono	LLE (2)
segment	TriMap6th (5)
spambase	PCA (10)
wdbc	MDS (5)

The value in the brackets represents the number of dimensions.

Table A2. RCBMIDE method extension (dimensionality reduction) for each lower-dimensions dataset.

Dataset	RCBMIDE
1balance-scale	UMAP (2)
atom	UMAP (2)
cpu	TriMap (5)
diabetes	TriMap (5)
ecoli	UMAP (4)
glass	LLE (5)
Haberman	TSVD (2)
iris	UMAP (6)
pmf	LLE (2)
thyroid	TSVD (4)
Wine	PCA (4)

The value in the brackets represents the number of dimensions.

References

1. 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]
2. 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. [PubMed]
3. Farouk, Y.; Rady, S. Early diagnosis of alzheimer's disease using unsupervised clustering. *Int. J. Intell. Comput. Inf. Sci.* **2020**, *20*, 112–124. [CrossRef]
4. 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]
5. Nie, W.; Cheng, H.; Su, Y. Modeling temporal information of mitotic for mitotic event detection. *IEEE Trans. Big Data* **2017**, *3*, 458–469. [CrossRef]
6. Abualigah, L.; Gandomi, A.H.; Elaziz, M.A.; Hamad, H.A.; Omari, M.; Alshinwan, M.; Khasawneh, A.M. Advances in meta-heuristic optimization algorithms in big data text clustering. *Electronics* **2021**, *10*, 101.
7. Lukauskas, M.; Pilinkienė, V.; Bruneckienė, J.; Stundžienė, A.; Grybauskas, A.; Ruzgas, T. Economic Activity Forecasting Based on the Sentiment Analysis of News. *Mathematics* **2022**, *10*, 3461. [CrossRef]
8. Trentin, E.; Lusnig, L.; Cavalli, F. Parzen neural networks: Fundamentals, properties, and an application to forensic anthropology. *Neural Netw.* **2018**, *97*, 137–151. [CrossRef] [PubMed]
9. Lukauskas, M.; Ruzgas, T. A New Clustering Method Based on the Inversion Formula. *Mathematics* **2022**, *10*, 2559. [CrossRef]
10. Ding, C.; He, X. K-means clustering via principal component analysis. In Proceedings of the 21st International Conference on Machine Learning, Banf, AL, Canada, 4–8 July 2004; p. 29.
11. Yang, L.; Liu, J.; Lu, Q.; Riggs, A.D.; Wu, X. SAIC: An iterative clustering approach for analysis of single cell RNA-seq data. *BMC Genom.* **2017**, *18*, 689.
12. Kakushadze, Z.; Yu, W. * K-means and cluster models for cancer signatures. *Biomol. Detect. Quantif.* **2017**, *13*, 7–31. [CrossRef] [PubMed]
13. Shin, J.; Berg, D.A.; Zhu, Y.; Shin, J.Y.; Song, J.; Bonaguidi, M.A.; Enikolopov, G.; Nauen, D.W.; Christian, K.M.; Ming, G.-L. Single-cell RNA-seq with waterfall reveals molecular cascades underlying adult neurogenesis. *Cell Stem Cell* **2015**, *17*, 360–372. [CrossRef] [PubMed]
14. Feng, C.; Liu, S.; Zhang, H.; Guan, R.; Li, D.; Zhou, F.; Liang, Y.; Feng, X. Dimension reduction and clustering models for single-cell RNA sequencing data: A comparative study. *Int. J. Mol. Sci.* **2020**, *21*, 2181. [CrossRef] [PubMed]
15. Melit Devassy, B.; George, S.; Nussbaum, P. Unsupervised clustering of hyperspectral paper data using t-SNE. *J. Imaging* **2020**, *6*, 29. [CrossRef] [PubMed]
16. Bollon, J.; Assale, M.; Cina, A.; Marangoni, S.; Calabrese, M.; Salvemini, C.B.; Christille, J.M.; Gustincich, S.; Cavalli, A. Investigating How Reproducibility and Geometrical Representation in UMAP Dimensionality Reduction Impact the Stratification of Breast Cancer Tumors. *Appl. Sci.* **2022**, *12*, 4247. [CrossRef]
17. Li, H.; Liu, J.; Liu, R.W.; Xiong, N.; Wu, K.; Kim, T.-h. A dimensionality reduction-based multi-step clustering method for robust vessel trajectory analysis. *Sensors* **2017**, *17*, 1792. [CrossRef]
18. Wenskovitch, J.; Crandell, I.; Ramakrishnan, N.; House, L.; North, C. Towards a systematic combination of dimension reduction and clustering in visual analytics. *IEEE Trans. Vis. Comput. Graph.* **2017**, *24*, 131–141. [CrossRef]
19. Tang, B.; Shepherd, M.; Milios, E.; Heywood, M.I. Comparing and combining dimension reduction techniques for efficient text clustering. In Proceedings of the SIAM International Conference on Data Mining, Newport Beach, CA, USA, 23 April 2005; pp. 17–26.
20. Wang, X.-D.; Chen, R.-C.; Zeng, Z.-Q.; Hong, C.-Q.; Yan, F. Robust dimension reduction for clustering with local adaptive learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *30*, 657–669. [CrossRef]

21. Markos, A.; D'Enza, A.I.; van de Velden, M. Beyond tandem analysis: Joint dimension reduction and clustering in R. *J. Stat. Softw.* **2019**, *91*, 1–24. [CrossRef]
22. Wenskovitch, J.; Dowling, M.; North, C. With respect to what? simultaneous interaction with dimension reduction and clustering projections. In Proceedings of the 25th International Conference on Intelligent User Interfaces, Cagliari, Italy, 17–20 March 2020; pp. 177–188.
23. Ruzgas, T.; Lukauskas, M.; Čepkauskas, G. Nonparametric Multivariate Density Estimation: Case Study of Cauchy Mixture Model. *Mathematics* **2021**, *9*, 2717. [CrossRef]
24. Kavaliauskas, M.; Rudzkis, R.; Ruzgas, T. The projection-based multivariate density estimation. *Acta Comment. Univ. Tartu. Math.* **2004**, *8*, 135–141. [CrossRef]
25. 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]
26. Xu, Q.; Yuan, S.; Huang, T. Multidimensional uniform initialization Gaussian mixture model for spar crack quantification under uncertainty. *Sensors* **2021**, *21*, 1283. [CrossRef] [PubMed]
27. Fraley, C. Algorithms for model-based Gaussian hierarchical clustering. *SIAM J. Sci. Comput.* **1998**, *20*, 270–281. [CrossRef]
28. Dempster, A.P.; Laird, N.M.; Rubin, D.B. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. Ser. B (Methodol.)* **1977**, *39*, 1–22.
29. Everitt, B. *Finite Mixture Distributions*; Springer Science & Business Media: New York, NY, USA, 2013.
30. Redner, R.A.; Walker, H.F. Mixture densities, maximum likelihood and the EM algorithm. *SIAM Rev.* **1984**, *26*, 195–239. [CrossRef]
31. Wold, S.; Esbensen, K.; Geladi, P. Principal component analysis. *Chemom. Intell. Lab. Syst.* **1987**, *2*, 37–52. [CrossRef]
32. Comon, P. Independent component analysis, a new concept? *Signal Process.* **1994**, *36*, 287–314. [CrossRef]
33. Jöreskog, K.G. Factor analysis as an error-in-variables model. In *Principals of Modern Psychological Measurement*; Routledge: Abingdon-on-Thames, UK, 1983; pp. 185–196.
34. Van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2580–2605.
35. Van Der Maaten, L. Accelerating t-SNE using tree-based algorithms. *J. Mach. Learn. Res.* **2014**, *15*, 3221–3245.
36. Li, W.; Cerise, J.E.; Yang, Y.; Han, H. Application of t-SNE to human genetic data. *J. Bioinform. Comput. Biol.* **2017**, *15*, 1750017. [CrossRef] [PubMed]
37. Kobak, D.; Berens, P. The art of using t-SNE for single-cell transcriptomics. *Nat. Commun.* **2019**, *10*, 1–14. [CrossRef] [PubMed]
38. Amid, E.; Warmuth, M.K. TriMap: Large-scale dimensionality reduction using triplets. *arXiv* **2019**, arXiv:1910.00204.
39. Ghojogh, B.; Ghodsi, A.; Karray, F.; Crowley, M. Locally linear embedding and its variants: Tutorial and survey. *arXiv* **2020**, arXiv:2011.10925.
40. Venna, J.; Kaski, S. Neighborhood Preservation in Non-linear Projection Methods: An Experimental Study. In Proceedings of the Artificial Neural Networks—ICANN, Berlin/Heidelberg, Germany, 21–25 August 2001; pp. 485–491.
41. Caliński, T.; Harabasz, J. A dendrite method for cluster analysis. *Commun. Stat. -Theory Methods* **1974**, *3*, 1–27. [CrossRef]
42. Davies, D.L.; Bouldin, D.W. A cluster separation measure. *IEEE Trans. Pattern Anal. Mach. Intell.* **1979**, *PAMI-1*, 224–227. [CrossRef]
43. Ahmadinejad, N.; Liu, L. J-Score: A Robust Measure of Clustering Accuracy. *arXiv* **2021**, arXiv:2109.01306.
44. Zhong, S.; Ghosh, J. Generative model-based document clustering: A comparative study. *Knowl. Inf. Syst.* **2005**, *8*, 374–384. [CrossRef]
45. Lawrence, H.; Phipps, A. Comparing partitions. *J. Classif.* **1985**, *2*, 193–218.
46. 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]
47. Fowlkes, E.B.; Mallows, C.L. A method for comparing two hierarchical clusterings. *J. Am. Stat. Assoc.* **1983**, *78*, 553–569. [CrossRef]