entropy

Article

An Improved K-Means Algorithm Based on Evidence Distance

Ailin Zhu [1], Zexi Hua [1,*], Yu Shi [2], Yongchuan Tang [3] and Lingwei Miao [2,4]

[1] School of Information Science and Technology, Southwest Jiaotong University, Chengdu 611756, China; zhual@my.swjtu.edu.cn
[2] School of Electrical Engineering, Southwest Jiaotong University, Chengdu 611756, China; shiy@my.swjtu.edu.cn (Y.S.); miaolw@my.swjtu.edu.cn (L.M.)
[3] School of Big Data and Software Engineering, Chongqing University, Chongqing 401331, China; tangyongchuan@cqu.edu.cn
[4] Qianghua Times (Chengdu) Technology Co., Ltd., Chengdu 610095, China
* Correspondence: zxhua@swjtu.edu.cn

Abstract: The main influencing factors of the clustering effect of the k-means algorithm are the selection of the initial clustering center and the distance measurement between the sample points. The traditional k-mean algorithm uses Euclidean distance to measure the distance between sample points, thus it suffers from low differentiation of attributes between sample points and is prone to local optimal solutions. For this feature, this paper proposes an improved k-means algorithm based on evidence distance. Firstly, the attribute values of sample points are modelled as the basic probability assignment (BPA) of sample points. Then, the traditional Euclidean distance is replaced by the evidence distance for measuring the distance between sample points, and finally k-means clustering is carried out using UCI data. Experimental comparisons are made with the traditional k-means algorithm, the k-means algorithm based on the aggregation distance parameter, and the Gaussian mixture model. The experimental results show that the improved k-means algorithm based on evidence distance proposed in this paper has a better clustering effect and the convergence of the algorithm is also better.

Keywords: k-means clustering; evidence distance; cluster analysis; evidence theory

check for updates

Citation: Zhu, A.; Hua, Z.; Shi, Y.; Tang, Y.; Miao, L. An Improved K-Means Algorithm Based on Evidence Distance. *Entropy* **2021**, *23*, 1550. https://doi.org/10.3390/e23111550

Academic Editors: Zoran H. Peric, Vlado Delic and Vladimir Despotovic

Received: 27 September 2021
Accepted: 18 November 2021
Published: 21 November 2021

1. Introduction

With the rapid development of technologies such as cloud computing and the internet of things [1,2], the number of connected devices is increasing and the data generated during human–computer interaction and system operation is growing exponentially [3–5]. In response to fast-growing data, data mining technology is constantly updated and iterated [6–8]. Clustering is a method of data mining [9]. A data set is divided into multiple clusters through a certain process [10,11]. Data similarity within clusters is high, while data similarity between clusters is low [12–14]. Depending on the clustering method and characteristics, clustering algorithms can be classified as: divisional, hierarchical, density algorithms, graph theoretic clustering, grid algorithms, model algorithms, etc. [15,16].

The k-means algorithm has been widely used due to its simple algorithm idea, easy implementation, and high efficiency when processing large-scale data [17,18]. However, the traditional k-means algorithm has major limitations [19,20]. For example, when using Euclidean distance calculations, the degree of discrimination between clusters is low and the output results in unstable values [21,22]. In view of the shortcomings of the traditional k-means algorithm, the k-means algorithm can be improved from different perspectives, such as random sampling, distance optimization, and density estimation methods [23]. Better results can be obtained by improving the method of measuring distances between sample points. Researchers at home and abroad have done a lot of research on distance optimization. Tang et al. [24] proposed the d-k-means algorithm, which weighs the influence of density and distance on clustering based on traditional algorithms, and weights the data. On the basis of weights, the principle of minimum and maximum is introduced to automatically determine the initial cluster centers and the number of centers. Wang et al. [25] proposed an improved k-means algorithm based on distance and sample weights, using dimensionally weighted Euclidean distance to calculate the distance between samples. Wang et al. [26] proposed a new algorithm to help k-means jump out of a local optimum on the basis of several ideas from evolutionary computation, through the use of random and evolutionary processes. Zhao et al. [27] proposed a new variant of k-means. The clustering process is driven by an explicit objective function, which makes the k-means process simpler and converges to a better local optimal solution. Qi et al. [28] proposed an optimized k-means clustering method, named k*-means, and three optimization principles, which can reduce the risk of randomly selecting seeds and reduce the adjustable space. Chen et al. [29] proposed an efficient hybrid clustering algorithm called QALO-K, which combines k-means with an optimized quantum-inspired antlion to make the k-means algorithm converge towards the global optimum. Zhang et al. [30] proposed the DC-k-means algorithm, which added the idea of canopy. At the same time, it combines the sample density in the process of finding the initial clusters, which has a good effect when dealing with low-density areas; however, it is possible that the outliers are classified into one class in the clustering process, which affects the clustering effect.

Dempster–Shafer (DS) theory, also known as evidence theory, was first proposed by Dempster in 1967 and was refined and developed by his student Shafer in 1976. Because evidence theory can meet uncertainty and uncertain information flexibly and effectively without relying on a priori knowledge, it is widely used in many fields, such as: correlation analysis, clustering, classification, etc. Fred et al. discussed the problem of clustering data based on evidence. The n d-dimensional data are decomposed into a large number of compact clusters, and then the k-means algorithm is used to cluster them separately, and several clustering results are obtained, which constitute the association matrix. Finally, the final clustering results are obtained using the MST algorithm on the basis of the association matrix. This method can effectively identify arbitrary clusters in multidimensional data [31]. Li et al. proposed a clustering integration algorithm based on evidence theory, which focuses on the fusion process in the clustering integration algorithm. After obtaining the probability of belonging to each label using the label distribution status of the neighborhood information of the object under test, the probability values are used to form the basic partition. After that, fusion is performed using the Dempster–Shafer fusion rules to obtain the final clustering results. This algorithm avoids blind trust in the obtained labels [32]. Yu et al. proposed a three-way density-peak clustering algorithm based on evidence theory, which uses a density-peak clustering algorithm to obtain clustering centers and noise points, and then uses a mid-distance comparison scheme to merge neighboring points. Finally, the remaining points are assigned using the evidence distance fusion rule. The method effectively solves the problem of error propagation of clustering labels [33].

The main feature of k-means algorithm clustering is the high degree of similarity of data in the same class and the low degree of similarity of data in different classes. The evidence distance in evidence theory can be used to describe the degree of similarity between two bodies of evidence. In order to explore whether a new distance measure can be obtained by using evidence distance instead of Euclidean distance, an improved k-means algorithm based on evidence distance is proposed in this paper. In this paper, we use the attribute values of each sample point to form the evidence body of each sample point, and then select the class in which the cluster center with the smallest distance is added based on the evidence distance from each evidence body to the initial cluster center. Finally, it is divided into k classes to obtain the final clustering results. Through validation on the UCI data set and toy data set, and experimental comparison with the traditional k-means algorithm, and the k-means algorithm based on the aggregation distance parameter and the Gaussian mixture model, the improved k-means algorithm in this paper has better clustering effect and convergence.

The rest of the thesis is organized as follows. The second section provides a review of relevant theory. The third section introduces the algorithmic ideas and motivation of this paper and proposes a k-means algorithm based on evidence distance improvement. The fourth section describes the experimental setting and the chosen algorithm evaluation metrics. The fifth section is devoted to conducting relevant experiments on the UCI dataset and the toy dataset and comparing the experimental results with some existing algorithms. Finally, the sixth section provides the conclusion.

2. Related Theories

2.1. Traditional K-Means Algorithm

The core idea of the k-means algorithm is: After inputting the k value, randomly select k sample points in the sample point set as the initial clustering center. Then, the distances of the remaining sample points to the initial cluster centers are calculated and the sample points are grouped into the closest clusters. In the generated new clusters, new cluster centroids are reselected and the sample points are clustered and classified again until the clustering classification results no longer change [34]. In the actual application process, after multiple iterations, due to various factors, the termination conditions may not be met. Therefore, a maximum number of iterations will be set in the actual application process, and the calculation will be terminated when the maximum number of iterations is reached. The pseudo-code of the traditional k-means algorithm is summarized as Algorithm 1.

Algorithm 1 The traditional k-means algorithm.

```
Input: data set, k value
Output: divided into k clusters
select k points from the sample Euclidean from sample point $x_i$ to each cluster center
repeat
    for j=1, 2, . . . . . . , m
        calculate the Euclidean distance from sample point $x_i$ to each cluster center
        determine the cluster class mark of $x_i$ according to the closest distance
        divide the sample points into corresponding clusters
    end for
    calculate new cluster centers
until the cluster allocation result remains unchanged
```

The traditional k-means algorithm distance measures include: Euclidean metric, city block distance, Pearson correlation, absolute value correlation, absolute non-central correlation, Spearman rank correlation, and Kendall's tau. The traditional k-means algorithm mainly uses the Euclidean distance [35].

The Euclidean metric [36,37] (also known as the Euclidean distance) is a commonly adopted definition of distance and refers to the true distance between two points in m-dimensional space, or the natural length of a vector (i.e., the distance from that point to the origin). The Euclidean distance in two and three dimensions is the actual distance between two points [38].

The distance measurement formula in two-dimensional space:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{1}$$

where d is the Euclidean distance between the point (x_2, y_2) and (x_1, y_1).

The distance measurement formula in three-dimensional space:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \tag{2}$$

2.2. D-S Evidence Theory

Evidence theory was first proposed by Dempster [39] and further developed by his student Shafer [40], an imprecise reasoning theory, also known as Dempster–Shafer evidence theory. As an uncertain reasoning method, the main characteristics of evidence theory are: it satisfies lower conditions than naive Bayesian probability theory and it has the ability to express 'uncertainty' and 'not knowing' directly. At the heart of D–S evidence theory is the Dempster combination rule, which integrates the underlying reliability distributions of multiple information sources and obtains a new reliability distribution as an output [41–44].

Definition 1. *Assuming that a non-empty set Θ is composed of m mutually exclusive events, Θ is the identification frame, $\Theta = \{\theta_1, \theta_2, \ldots\ldots \theta_n\}$. The power set of Θ is represented by 2^Θ, $2^\Theta = \{\varnothing, \{\theta_1\}, \{\theta_1, \theta_2\}, \ldots\ldots \{\theta_1, \theta_2, \ldots\ldots \theta_n\}\}$ [39,40].*

Definition 2. *For any $A \in 2^\Theta$, m is the mass function. For any subset A in m, let $m(A_i) \in (0, 1)$, satisfy the following conditions [39,40]:*

$$\sum_{A \subseteq \Theta} m(A_i) = 1, m(\Phi) = 0 \tag{3}$$

Among them, $m(A_i)$ represents the basic probability of A.

Definition 3. *Body of Evidence (BOE) is a collection of all focal members and its corresponding mass functions, expressed as follows [39,40]:*

$$(B, m) = \left\{ [A, m(A)] \middle| A\epsilon 2^\theta \text{ and } m(A) > 0 \right\} \tag{4}$$

where B is a subset of the power set 2^θ.

Definition 4. *A's belief function (Bel) represents A's total trust, and A's likelihood function (Pl) represents the confidence level of not denying A. Belief function (Bel) and likelihood function (Pl) represent the upper limit function and lower limit function of A, respectively, defined as follows [39,40]:*

$$Bel(A) = \sum_{B \subseteq A} m(B) \qquad \forall A \subseteq \Theta \tag{5}$$

$$Pl(A) = 1 - Bel\left(\overline{A}\right) = \sum_{B \cap A \neq \varnothing} m(B)\ \forall A \subseteq \Theta \tag{6}$$

where $Bel(A) \leq Pl(A)$.

Definition 5. *Assuming that under the basic identification framework, the basic probability distribution functions of the two bodies of evidence (BOE) are m_1 and m_2, respectively, the formula for combining according to the Dempster rule is as follows [39]:*

$$m(C) = m_i(A) \oplus m_i(B) = \begin{cases} 0\ A \cap B = \varnothing \\ \frac{\sum_{A \cap B = C, B \subseteq \Theta} m_i(A) \times m_i(B)}{1-K}\ A \cap B \neq \varnothing \end{cases} \tag{7}$$

where $m_i(A)$, $m_i(B)$ represents two bodies of evidence and *$m(C)$ represents the consensus of two bodies of evidence; K represents the conflicting factor between the two evidence bodies and is defined as follows:*

$$K = \sum_{A \cap B = \varnothing, \forall A, B \subseteq \Theta} m_i(A) \times m_i(B) \tag{8}$$

Definition 6. *Evidence distance [45–47] is usually used to describe the degree of difference between two evidence bodies, and its calculation formula is as follows:*

$$d_{BOE}(m_1, m_2) = \left(\vec{m_1} - \vec{m_2}\right)^T \underline{\underline{D}}\left(\vec{m_1} - \vec{m_2}\right) \tag{9}$$

The actual calculation formula used is as follows:

$$d_{BOE}(m_1, m_2) = \sqrt{\frac{1}{2}\left(\vec{m_1} - \vec{m_2}\right)^T \underline{\underline{D}}\left(\vec{m_1} - \vec{m_2}\right)} \tag{10}$$

Among them d_{BOE}: the distance between the two evidence bodies; m_1 represents body of evidence 1 and m_2 represents body of evidence 2; $\vec{m_1}$,$\vec{m_2}$: the vector constituted by the basic distribution probabilities of the two evidence bodies. $\underline{\underline{D}}$ is a $2^N \times 2^N$ matrix, the row index corresponds to m_1, and the column index corresponds to m_2, indicating the similarity between the two evidence bodies. Each element of the matrix can be represented as:

$$\mathrm{d}_{ij} = |m_1 \cap m_2| \div |m_1 \cup m_2| \tag{11}$$

3. Algorithm Design

3.1. Algorithm Idea Description

The algorithmic idea is that since the selection of *k* values is not optimized in this method, trial and error is used to find the optimum number of clustering centers, i.e., *k* values. *k* sample points are randomly selected as the initial clustering centers. The attributes of the sample points can be regarded as experts for judging the sample points that belong to a certain class, so the values of the attributes of the sample points are used to form the evidence body of each sample point. After that, the distance from the evidence body to the initial clustering center is calculated using the evidence distance formula. After the initial division of sample points, the clustering centers are then re-selected using the arithmetic mean algorithm. Finally, iterative calculations are performed until the clustering centers do not change.

3.2. Algorithm Flow

Step 1: For a given data set, randomly select *k* data sample points as the initial cluster center.

Step 2: Use the attribute value of each sample point to form the evidence body of each sample point.

Step 3: Use Formula (9) to calculate the evidence distance from each sample point to each initial cluster center, select the center with the smallest distance, and add the cluster center to the class.

Step 4: Select *k* cluster centers again.

Step 5: Determine whether the clustering center has changed, if it has changed, continue the iteration, if it remains the same, output the corresponding clustering result.

The algorithm flow chart is shown in Figure 1.

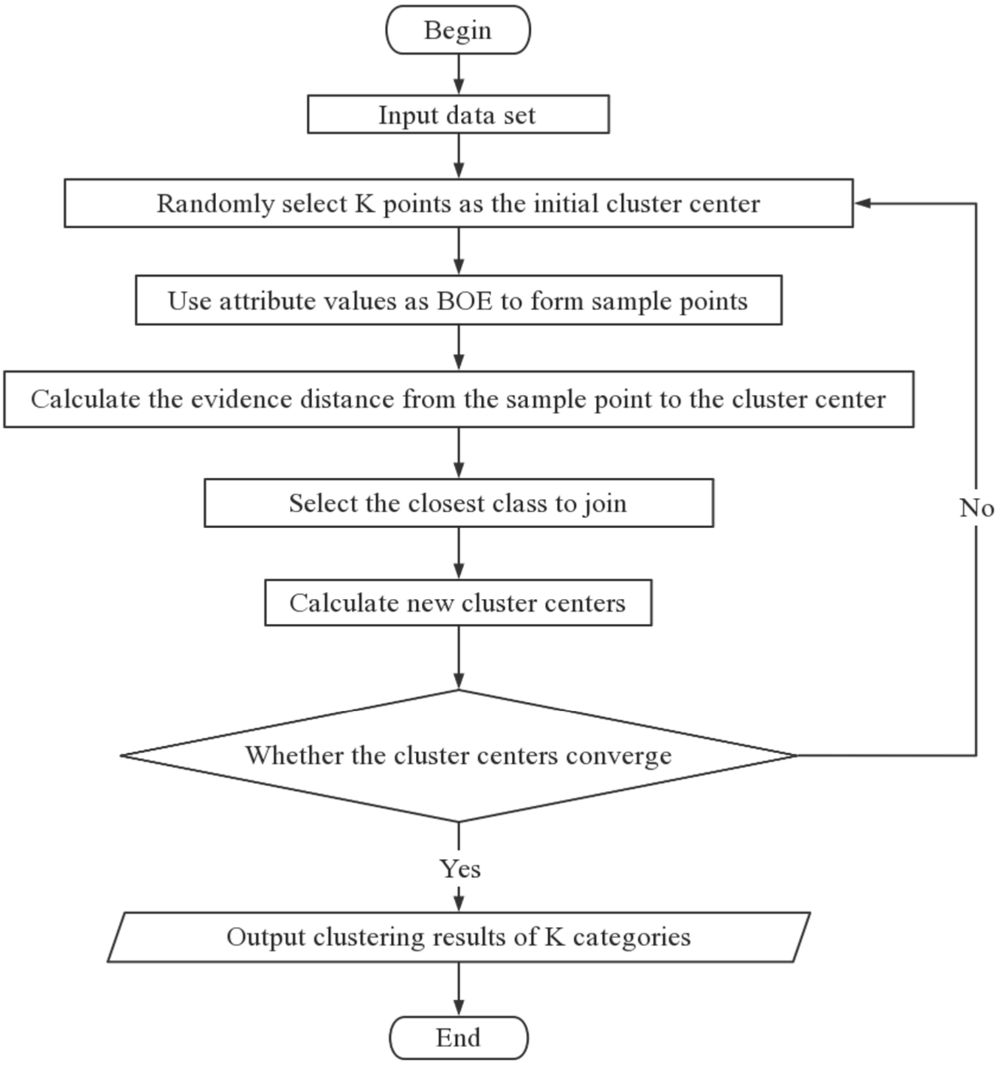

Begin
Input data set
Randomly select K points as the initial cluster center
Use attribute values as BOE to form sample points
Calculate the evidence distance from the sample point to the cluster center
Select the closest class to join
Calculate new cluster centers
Whether the cluster centers converge
No
Yes
Output clustering results of K categories
End

Figure 1. The algorithm flowchart of the improved k-means algorithm.

The pseudo code for the evidence distance-based k-means algorithm proposed in this paper is summarized in Algorithm 2.

Algorithm 2 the k-means algorithm based on evidence distance.

Input: data set, k value
Output: clustering results

```
initialize k cluster centers
use the attribute value of the sample point to construct the BOE of the sample point
While true
    num = 0;
    for $i$ = 0 to k
        $C_i = \varnothing$
    end for
    for $j$ = 1 to $m$
        for $i$ = 1 to k
            calculate evidence distance, $d = \sqrt{\frac{1}{2}\left(\vec{m_i} - \vec{m_j}\right)^T \underline{\underline{D}}\left(\vec{m_i} - \vec{m_j}\right)}$
        end for
        min = $d$
        for $i$ = 2 to k
            if $d_{ij}$ < min
                min = $d_{ij}$
                temp = $i$
            end if
        end for
        Lambda = temp
        C(Lambda) = C(Lambda) + $\{x_j\}$
    end for
    for $i$ = 1 to k
        $U_i{}'$ = Update the mean vector based on the previous cluster
        if $U_i{}'! = U_i$
            $U_i = U_i{}'$
        else
            num++
        end if
    end for
    if num = $k$
        break
    end if
end while
```

4. Experiment

4.1. Experiment Preparation

The experimental environment is: AMD A10-7300 processor, AMD Raden R7 M260DX graphics card, 8G of running memory, windows10 operating system, and programming with Python 3.7–32 bits.

4.1.1. Experimental Data Set

The data set used in this article comes from the UCI data set. The name of the data set and its attributes are shown in Table 1.

Table 1. Experimental data set.

Data Set	Number of Samples	Feature Number	Number of Categories
Iris	150	4	3
Wine	178	13	3
Breast_cancer	699	10	2
Digits	1797	5	9
Pima	768	8	2

4.1.2. Experimental Evaluation Indicators

The evaluation indicators used in this paper mainly include adjusting the Rand index, the contour coefficient, and the number of iterations. Adjustment of the Rand index and silhouette coefficient are used to evaluate the clustering performance of the algorithm, and the number of iterations is used to evaluate the convergence of the algorithm.

(1) Adjusted Rand index

In the clustering model, assuming that the actual category information is *C* and the clustering result is *K*, a denotes the number of pairs of elements that are both in the same category in *C* and *K*, and b denotes the number of pairs of elements that are both in different categories in *C* and *K*. The Rand index is defined as:

$$RI = \frac{a+b}{C_2^{n_{samples}}} \tag{12}$$

where $C_2^{n_{samples}}$ represents the total number of pairs of elements that can be composed in the data set. The range of *RI* is [0, 1] and a higher value of *RI* means that the clustering results match the real situation.

The problem with the Rand index is that for two random divisions, the value of the Rand coefficient is not a constant close to zero. Therefore, the adjusted Rand index is used, which has a higher degree of discrimination. The *ARI* is calculated as:

$$ARI = \frac{(RI - E[RI])}{max[RI] - E[RI]} \tag{13}$$

where *RI* is the Rand index and E[RI] represents the mean value. The range of values for *ARI* is [–1, 1]. A larger value for *ARI* means that the clustering results match the real situation.

(2) Silhouette Coefficient

The silhouette coefficient is a way of evaluating how well clustering works. It was first proposed by Peter J. Rousseeuw in 1986. It combines both cohesion and separation factors.

Suppose we have completed clustering by some clustering algorithm. For any one of these samples, *A* represents the average distance between the sample and the other samples in its cluster, and *B* represents the average distance between the sample and the samples in the other clusters, the silhouette coefficient of the sample is:

$$S = \frac{B - A}{max(A, B)} \tag{14}$$

where *S* denotes the silhouette coefficient of a single sample. The total silhouette coefficient of clustering is the average value of all sample silhouette coefficients. The contour coefficients range from $(-1, 1)$, with values closer to 1 indicating better clustering performance, and conversely, values closer to -1 indicating worse clustering performance.

(3) Number of iterations

Number of iterations: how many times the algorithm iterates until the algorithm converges in the actual operation. Since it is a random result, the experiments in this paper take the arithmetic average of the number of iterations after several iterations. The smaller the value, the faster the convergence of the algorithm.

$$\text{Calculation formula}: \ Number\ of\ iterations = \frac{Total\ number\ of\ iterations}{Total\ number\ of\ runs} \tag{15}$$

4.2. Experimental Procedure

(1) Import the iris data set and enter the cluster category *k* value.
(2) The traditional k-means method and the improved k-means method are used for clustering, respectively.
(3) Perform clustering 10 times, find the average value, and output the ARI, contour coefficient, and number of iterations as the final result.
(4) Compare the experimental results of the improved algorithm and the traditional algorithm.
(5) Use wine, breast cancer, and other data sets for verification.

5. Results and Analysis

5.1. Iris Data Set Test Results

After 10 clusters, the *ARI* value of each cluster is shown in Figure 2, the Silhouette Coefficient value is shown in Figure 3, and the number of iterations is shown in Figure 4. The final result is obtained by calculating the average value. The *ARI* value of the traditional method is 0.603, the profile coefficient value is 0.5371, and the number of iterations is 8.8 times. The *ARI* value of the improved method is 0.719, the silhouette coefficient value is 0.5514, and the number of iterations is 8.3 times. From Figures 2–4, it can be seen that the new method adopted in this paper is more stable than the traditional method, and the *ARI* value and the silhouette coefficient have been effectively improved. Therefore, the accuracy of this method is better than that of the traditional method, and better clustering effect can be obtained. The improved method has generally reduced the number of iterations compared with the traditional method, so the convergence of the new method is also better than that of the traditional method.

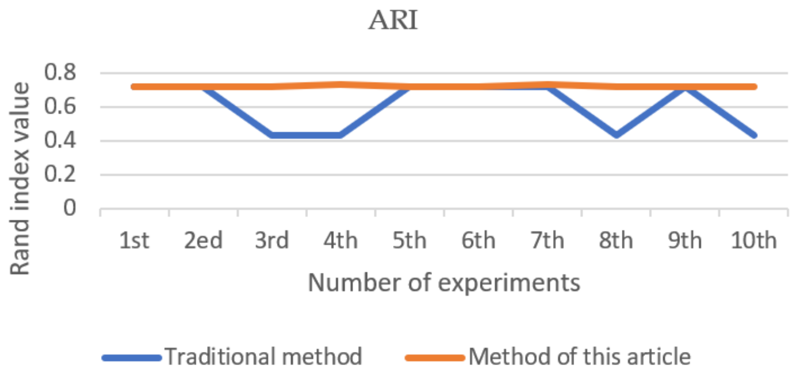


Figure 2. Adjusted rand index (iris).

5.2. Validation Results of Other Data Sets

The clustering effect and convergence of the algorithm were verified by using wine, breast cancer, digits, and pima datasets with ARI values, silhouette coefficient, and number of iterations as shown in Figures 5–7. The analysis of Figures 5 and 6 shows that the new method can obtain better clustering results and the output results of the new method are more stable in the output process. However, when there are more attribute values in the data set, the improvement of the new method is smaller. Through the analysis of Figure 7, it can be seen that, except for clustering using the breast cancer data set, the convergence of the new method is slightly worse than that of the traditional method, and the overall convergence of the new method is better than that of the traditional method.

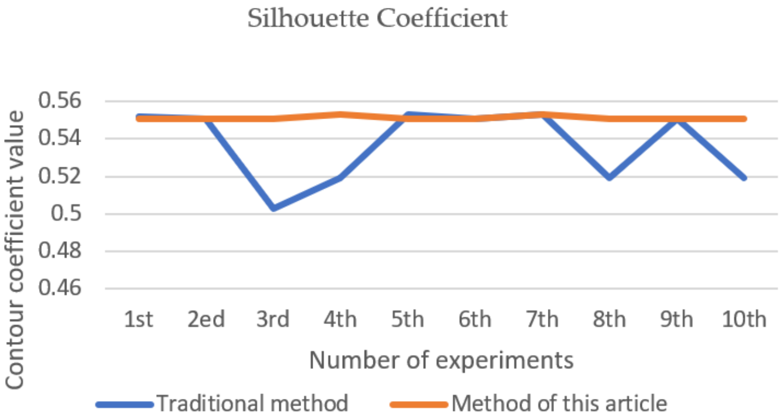


Figure 3. Silhouette Coefficient (iris).

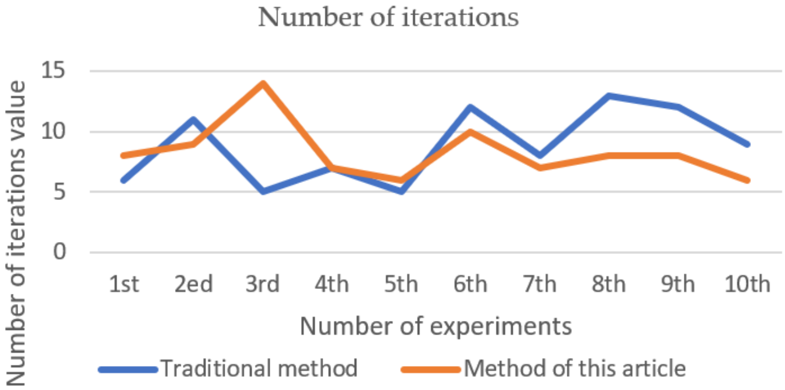


Figure 4. Number of iterations (iris).

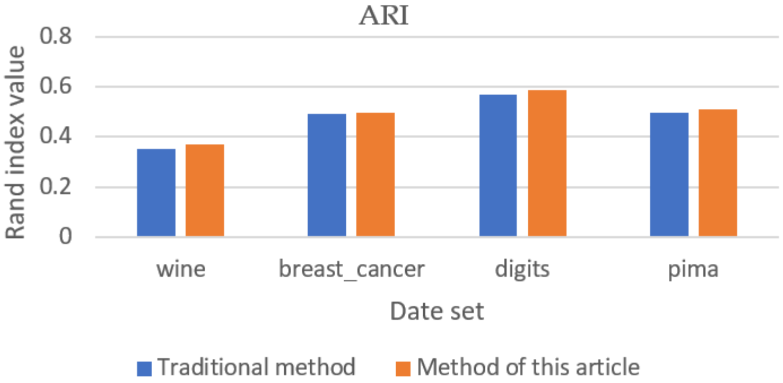


Figure 5. Adjusted rand index (validation data set).

In summary, the new method used in this experiment can obtain better clustering results than traditional methods, and in the output process, the variance between the results is smaller and the output is more stable. At the same time, the convergence of the algorithm is improved to a certain extent.

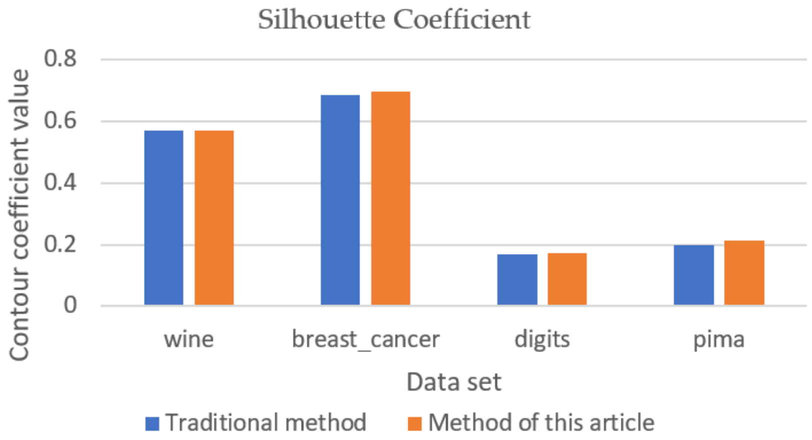


Figure 6. Silhouette coefficient (validation data set).

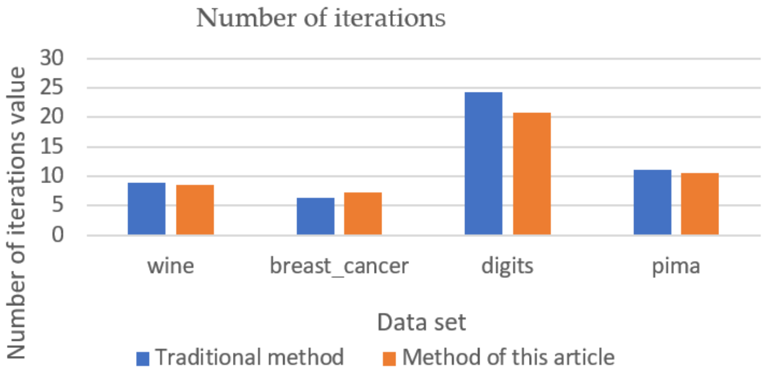
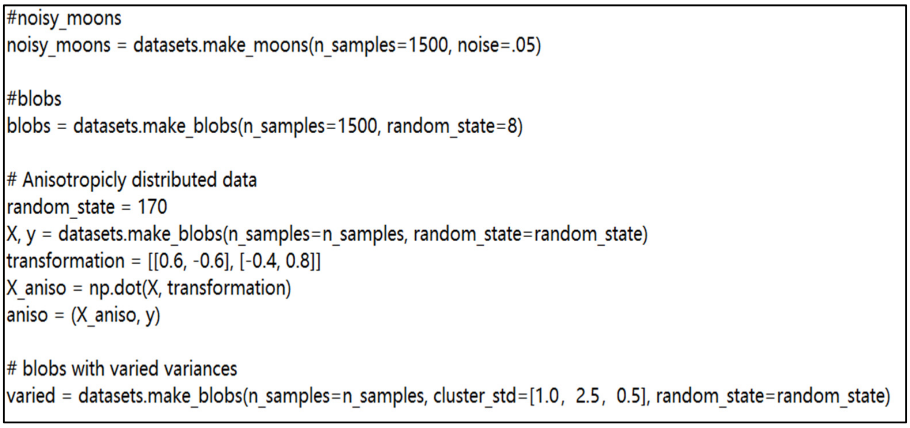


Figure 7. Number of iterations (validation data set).

5.3. *Algorithm Comparison*

In order to conduct a more in-depth verification of the performance of the evidence-distance-based improved k-means algorithm proposed in this paper, the performance of the traditional k-means algorithm (T-K-means), the k-means algorithm based on aggregated distance parameters (AD-K-means) [48], the Gaussian mixture model (GMM) [49], and the k-means algorithm based on evidence distance proposed in this paper (ED-K-means) were selected for experimental comparison. The datasets used for the experiments were the UCI dataset and four toy datasets, iris, digits, wine, noisy-moon, blobs, anisotropicly distributed data, and blobs with varied variances, in that order. The parameters for the four toy datasets are shown in Figure 8.


```
#noisy_moons
noisy_moons = datasets.make_moons(n_samples=1500, noise=.05)

#blobs
blobs = datasets.make_blobs(n_samples=1500, random_state=8)

# Anisotropicly distributed data
random_state = 170
X, y = datasets.make_blobs(n_samples=n_samples, random_state=random_state)
transformation = [[0.6, -0.6], [-0.4, 0.8]]
X_aniso = np.dot(X, transformation)
aniso = (X_aniso, y)

# blobs with varied variances
varied = datasets.make_blobs(n_samples=n_samples, cluster_std=[1.0,  2.5,  0.5], random_state=random_state)
```

Figure 8. Data set parameters.

The experimental results were evaluated in terms of adjusted Rand index (*ARI*), silhouette coefficient, number of iterations, and algorithm runtime. The experimental results are shown in Figures 9–12.

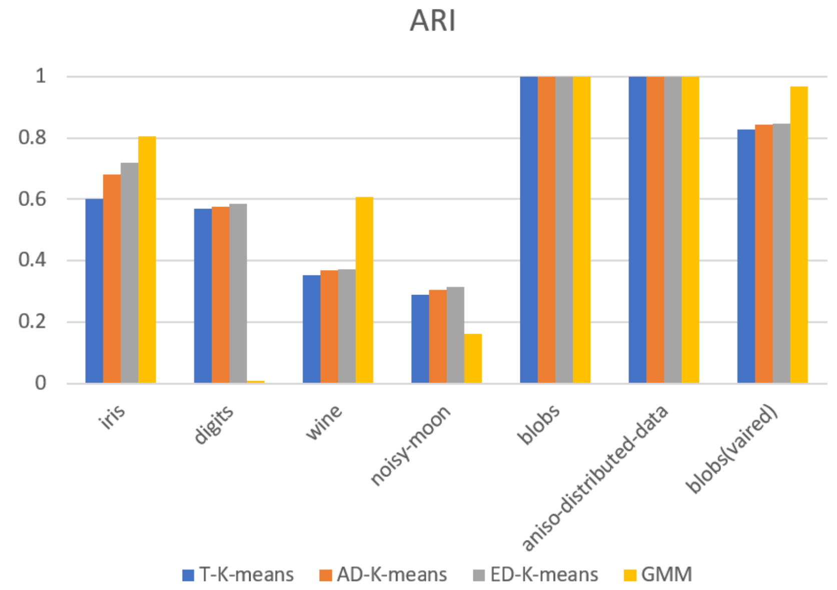


Figure 9. Adjusted rand index.

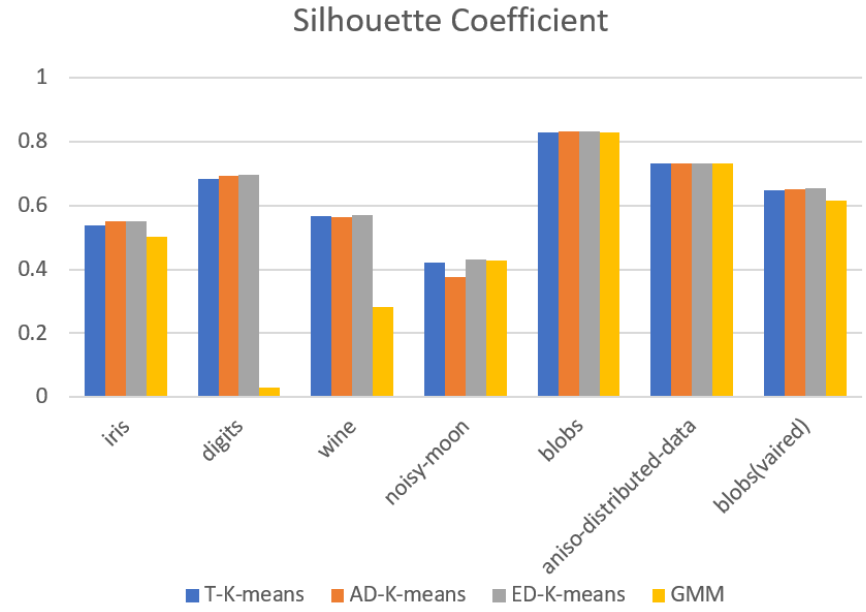


Figure 10. Silhouette coefficient.

Figure 9 shows the results of the adjusted Rand index, with larger values indicating that the clustering results are more consistent with the actual situation. The ED-K-means algorithm proposed in this paper gives higher results than the other three algorithms in both the digits and noisy-moon datasets. In the iris, wine, and blobs with varied variances datasets, the results are slightly lower than those of the GMM algorithm, but higher than those of the other two algorithms. Figure 10 shows the values of the silhouette coefficient, with larger values indicating that the clustering results are more consistent with the actual situation. The results of the ED-K-means algorithm proposed in this paper outperformed the other three algorithms on both the toy dataset and the UCI dataset. Therefore, the ED-K-means algorithm proposed in this paper can achieve better clustering results.

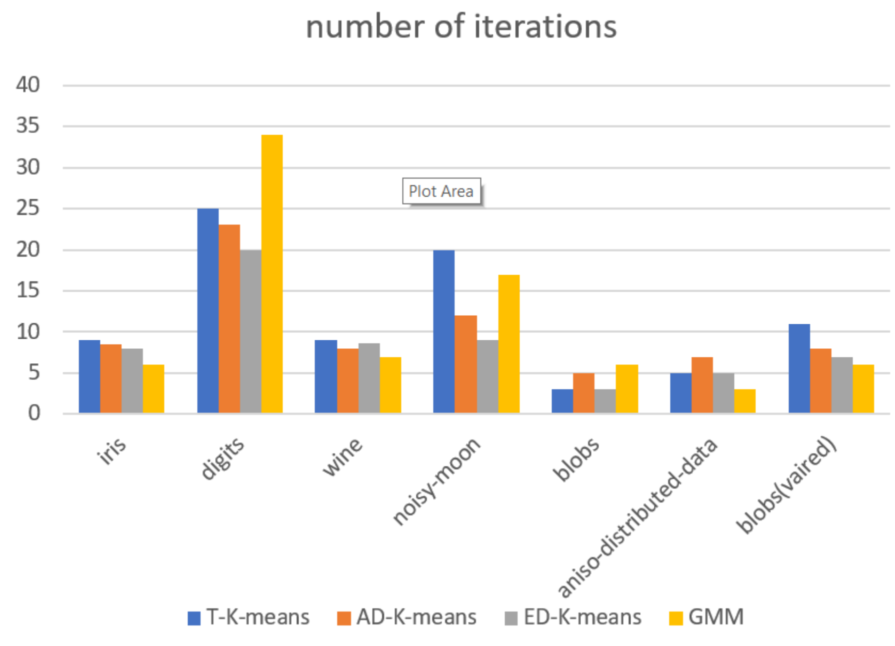


Figure 11. Number of iterations.

Figure 11 shows the value of the average number of iterations of the algorithm and Figure 12 shows the algorithm running time. Both algorithm metrics are smaller indicating better convergence of the algorithm. The analysis of the results in this figure shows that in the toy dataset, the ED-K-means algorithm proposed in this paper has the lowest number of iterations and the running time is comparable with T-K-means and slightly higher than the GMM algorithm. In the three UCI datasets, the number of iterations is less and the running time is better than that of T-K-means and GMM, and similar to that of AD-K-means. Therefore, on balance, the ED-K-means algorithm proposed in this paper has better convergence.

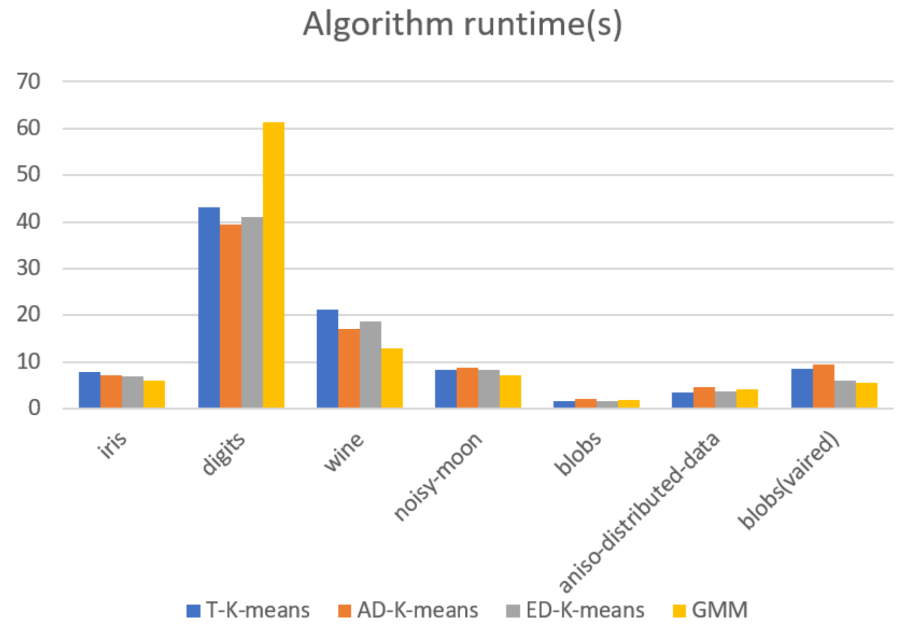


Figure 12. Algorithm runtime.

6. Conclusions

In the era of big data, data is expanding, so the clustering algorithm has a wide range of application scenarios. This paper presents an improved k-means algorithm based on evidence distance. The algorithm uses the attribute values of the sample points to form the body of evidence for the sample points. Then, the distance measure between sample points is performed using the evidence distance instead of the Euclidean distance. Finally, the k-means algorithm was used to cluster. Through experimental comparison, the improved k-means algorithm based on evidence distance proposed in this paper has good clustering effect and convergence. However, the initial clustering centers are still selected randomly when processing the data in this paper, so it can be further optimized.

Author Contributions: Conceptualization, A.Z., Z.H., and Y.T.; funding acquisition, Z.H.; methodology, A.Z., Y.S., Y.T., and L.M.; writing—original draft, A.Z., Y.S., and L.M.; writing—review and editing, Y.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially supported by the National Key Research and Development Project of China (grant no. 2020YFB1711900).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All relevant data are within the paper.

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

References

1. Barua, H.B.; Mondal, K.C. A comprehensive survey on cloud data mining (CDM) frameworks and algorithms. *ACM Comput. Surv.* **2019**, *52*, 1–62. [CrossRef]
2. Atluri, G.; Karpatne, A.; Kumar, V. Spatio-temporal data mining: A survey of problems and methods. *ACM Comput. Surv.* **2018**, *51*, 1–41. [CrossRef]
3. Fei, X.; Tian, G.; Lima, S. Research on data mining algorithm based on neural network and particle swarm optimization. *J. Intell. Fuzzy Syst.* **2018**, *35*, 2921–2926. [CrossRef]
4. Manda, P. Data mining powered by the gene ontology. Wiley Interdisciplinary Reviews. *Data Min. Knowl. Discov.* **2020**, *10*, e1359.
5. Duggirala, H.J.; Tonning, J.M.; Smith, E.; Bright, R.A.; Baker, J.D.; Ball, R.; Bell, C.; Bright-Ponte, S.J.; Botsis, T.; Bouri, K.; et al. Use of data mining at the Food and Drug Administration. *J. Am. Med. Inform. Assoc.* **2016**, *23*, 428–434. [CrossRef]
6. Zhang, C.; Hao, L.; Fan, L. Optimization and improvement of data mining algorithm based on efficient incremental kernel fuzzy clustering for large data. *Clust. Comput.* **2019**, *22*, 3001–3010. [CrossRef]
7. Yu, W. Challenges and reflections of big data mining Based on mobile internet customers. *Agro. Food Ind. Hi Tech.* **2017**, *28*, 3221–3224.
8. Feng, Z.; Zhu, Y. A Survey on Trajectory Data Mining: Techniques and Applications. *IEEE Access* **2017**, *4*, 2056–2067. [CrossRef]
9. Zhou, J.; Wang, Q.; Hung, C.C.; Yang, F. Credibilistic clustering algorithms via alternating cluster estimation. *J. Intell. Manuf.* **2017**, *28*, 727–738. [CrossRef]
10. Bulut, H.; Onan, A.; Korukoğlu, S. An improved ant-based algorithm based on heaps merging and fuzzy c-means for clustering cancer gene expression data. *Sādhanā* **2020**, *45*, 1–17. [CrossRef]
11. Mbyamm Kiki, M.J.; Zhang, J.; Kouassi, B.A. MapReduce FCM clustering set algorithm. *Clust. Comput.* **2021**, *24*, 489–500. [CrossRef]
12. Cao, L.; Liu, Y.; Wang, D.; Wang, T.; Fu, C. A Novel Density Peak Fuzzy Clustering Algorithm for Moving Vehicles Using Traffic Ra-dar. *Electronics* **2019**, *9*, 46. [CrossRef]
13. Gao, W. Improved Ant Colony Clustering Algorithm and Its Performance Study. *Comput. Intell. Neurosci.* **2016**, *2016*, 4835932. [CrossRef] [PubMed]
14. Yi, D.; Xian, F. Kernel-based fuzzy c-means clustering algorithm based on genetic algorithm. *Neurocomputing* **2016**, *188*, 233–238.
15. Kuo, R.J.; Mei, C.H.; Zulvia, F.E.; Tsai, C.Y. An application of a metaheuristic algorithm-based clustering ensemble method to APP customer segmentation. *Neurocomputing* **2016**, *205*, 116–129. [CrossRef]
16. Zhan, Q.; Jiang, Y.; Xia, K.; Xue, J.; Hu, W.; Lin, H.; Liu, Y. Epileptic EEG Detection Using a Multi-View Fuzzy Clustering Algorithm with Multi-Medoid. *IEEE Access* **2019**, *7*, 152990–152997. [CrossRef]
17. Ismkhan, H. I-k-means-plus: An iterative clustering algorithm based on an enhanced version of the k-means. *Pattern Recognition: J. Pattern. Recognit. Soc.* **2018**, *79*, 402–413. [CrossRef]
18. Sinaga, K.P.; Hussain, I.; Yang, M.S. Entropy K-Means Clustering with Feature Reduction Under Unknown Number of Clusters. *IEEE Access* **2021**, *9*, 67736–67751. [CrossRef]
19. Wang, X.; Bai, Y. The global Minmax k-means algorithm. *Springerplus* **2016**, *5*, 1665. [CrossRef] [PubMed]
20. Aggarwal, S.; Singh, P. Cuckoo, Bat and Krill Herd based k-means++ clustering algorithms. *Clust. Comput.* **2018**, *22*, 14169–14180. [CrossRef]

21. Yin, C.; Zhang, S. Parallel implementing improved k-means applied for image retrieval and anomaly detection. *Multimed. Tools. Appl.* **2017**, *76*, 16911–16927. [CrossRef]
22. Yu, S.; Chu, S.; Wang, C.; Chan, Y.; Chang, T. Two improved k-means algorithms. *Appl. Soft Comput.* **2018**, *68*, 747–755. [CrossRef]
23. Prasada, M.; Tripathia, S.; Dahalb, K. Unsupervised feature selection and cluster center initialization based arbitrary shaped clusters for intrusion detection. *Comput. Secur.* **2020**, *99*, 102062. [CrossRef]
24. Tang, Z.K.; Zhu, Z.Y.; Yang, Y.; Caihong, L.; Lian, L. D-K-means algorithm based on distance and density. *Appl. Res. Comp.* **2020**, *37*, 1719–1723.
25. Zilong, W.; Jin, L.; Yafei, S. Improved K-means algorithm based on distance and weight. *Comp. Eng. Appl.* **2020**, *56*, 87–94.
26. Wang, Y.; Luo, X.; Zhang, J.; Zhao, Z.; Zhang, J. An Improved Algorithm of K-means Based on Evolutionary Computation. *Intell. Autom. Soft Comput.* **2020**, *26*, 961–971. [CrossRef]
27. Zhao, W.L.; Deng, C.H.; Ngo, C.W. k-means: A revisit. *Neurocomputing* **2018**, *291*, 195–206. [CrossRef]
28. Qi, J.; Yu, Y.; Wang, L.; Liu, J.; Wang, Y. An effective and efficient hierarchical K-means clustering algorithm. *Int. J. Distrib. Sens. Netw.* **2017**, *13*, 1550147717728627. [CrossRef]
29. Chen, J.; Qi, X.; Chen, L.; Chen, F.; Cheng, G. Quantum-inspired ant lion optimized hybrid k-means for cluster analysis and intrusion detection. *Knowl. Based. Syst.* **2020**, *203*, 106167. [CrossRef]
30. Zhang, G.; Zhang, C.; Zhang, H. Improved K-means algorithm based on density canopy. *Knowl. Based. Syst.* **2018**, *145*, 289–297. [CrossRef]
31. Fred, A.L.; Jain, A.K. Data clustering using evidence accumulation. In Proceedings of the 2002 International Conference on Pattern Recognition, Quebec City, QC, Canada, 11–15 August 2002.
32. Li, F.; Qian, Y.; Wang, J.; Liang, J. Multigranulation information fusion: A Dempster-Shafer evidence theory-based clustering ensemble method. *Inf. Sci.* **2017**, *378*, 389–409. [CrossRef]
33. Yu, H.; Chen, L.; Yao, J. A three-way density peak clustering method based on evidence theory. *Knowl.-Based Syst.* **2020**, *211*, 106532. [CrossRef]
34. Fränti, P.; Sieranoja, S. K-means properties on six clustering benchmark datasets. *Appl. Intell.* **2018**, *48*, 4743–4759. [CrossRef]
35. Giannella, C.R. Instability results for Euclidean distance, nearest neighbor search on high dimensional Gaussian data. *Inf. Process. Lett.* **2021**, *169*, 106115. [CrossRef]
36. Drusvyatskiy, D.; Lee, H.L.; Ottaviani, G.; Thomas, R.R. The Euclidean distance degree of orthogonally invariant matrix varieties. *Isr. J. Math.* **2017**, *221*, 291–316. [CrossRef]
37. Morin, L.; Gilormini, P.; Derrien, K. Generalized Euclidean distances for elasticity tensors. *J. Elast.* **2020**, *138*, 221–232. [CrossRef]
38. Subba Rao, T. *Classification, Parameter Estimation and State Estimation-an Engineering Approach Using MATLAB*; John Wiley & Sons, Ltd.: West Sussex, UK, 2011; Volume 32, p. 194.
39. Dempster, A.P. Upper and Lower Probabilities Induced by a Multivalued Mapping. In *Classic Works Dempster–Shafer Theory Belief Functions*; Springer: Berlin/Heidelberg, Germany, 1966; Volume 38, pp. 57–72.
40. Shafer, G. *A Mathematical Theory of Evidence*; Princeton University Press: Princeton, NJ, USA, 1976; Volume 42.
41. Tang, Y.; Wu, D.; Liu, Z. A new approach for generation of generalized basic probability assignment in the evidence theory. *Pattern Anal. Appl.* **2021**, *24*, 1007–1023. [CrossRef]
42. Gong, Y.; Su, X.; Qian, H.; Yang, N. Research on fault diagnosis methods for the reactor coolant system of nuclear power plant based on D-S evidence theory. *Ann. Nucl. Energy* **2018**, *112*, 395–399. [CrossRef]
43. Deng, X.; Liu, Q.; Deng, Y.; Mahadevan, S. An improved method to construct basic probability assignment based on the confusion matrix for classification problem. *Inf. Sci.* **2016**, *340*, 250–261. [CrossRef]
44. Yuan, K.; Deng, Y. Conflict evidence management in fault diagnosis. *Int. J. Mach. Learn. Cybern.* **2019**, *10*, 121–130. [CrossRef]
45. Li, M.; Hu, Y.; Zhang, Q.; Deng, Y. A novel distance function of D numbers and its application in product engineering. *Eng. Appl. Artif. Intell.* **2016**, *47*, 61–67. [CrossRef]
46. Mo, H.; Lu, X.; Deng, Y. A generalized evidence distance. *J. Syst. Eng. Electron.* **2016**, *27*, 470–476. [CrossRef]
47. Wang, J.; Xiao, F.; Deng, X.; Fei, L.; Deng, Y. Weighted evidence combination based on distance of evidence and entropy function. *Int. J. Distrib. Sens. Netw.* **2016**, *12*, 3218784. [CrossRef]
48. Qiaoling, W.; Fei, Q.; Youhao, J. Improved K-means algorithm based on aggregation distance parameter. *Int. J. Comput. Appl.* **2019**, *39*, 2586–2590.
49. Khan, M.H.; Farid, M.S.; Grzegorzek, M. Spatiotemporal features of human motion for gait recognition. *Signal Image Video Process.* **2018**, *13*, 369–377. [CrossRef]