mathematics

MDPI

Article

An Improved K-Means Algorithm Based on Contour Similarity

Jing Zhao [1,*], Yanke Bao [2], Dongsheng Li [1] and Xinguo Guan [1]

[1] Key Laboratory of Industrial Automation and Machine Vision of Qiannan, School of Mathematics and Statistics, Qiannan Normal University for Nationalities, Duyun 558000, China; qyt_lds1110@126.com (D.L.); 18208645535@163.com (X.G.)
[2] College of Science, Liaoning Technical University, Fuxin 123000, China; baoyanke_9257@163.com
* Correspondence: zhaoyuhuan0927@outlook.com

Abstract: The traditional k-means algorithm is widely used in large-scale data clustering because of its easy implementation and efficient process, but it also suffers from the disadvantages of local optimality and poor robustness. In this study, a Csk-means algorithm based on contour similarity is proposed to overcome the drawbacks of the traditional k-means algorithm. For the traditional k-means algorithm, which results in local optimality due to the influence of outliers or noisy data and random selection of the initial clustering centers, the Csk-means algorithm overcomes both drawbacks by combining data lattice transformation and dissimilar interpolation. In particular, the Csk-means algorithm employs Fisher optimal partitioning of the similarity vectors between samples for the process of determining the number of clusters. To improve the robustness of the k-means algorithm to the shape of the clusters, the Csk-means algorithm utilizes contour similarity to compute the similarity between samples during the clustering process. Experimental results show that the Csk-means algorithm provides better clustering results than the traditional k-means algorithm and other comparative algorithms.

Keywords: k-means algorithm; degree of similarity; contour similarity; improved algorithm

MSC: 68W99

Citation: Zhao, J.; Bao, Y.; Li, D.; Guan, X. An Improved K-Means Algorithm Based on Contour Similarity. *Mathematics* **2024**, *12*, 2211. https://doi.org/10.3390/math12142211

Academic Editor: Ashkan Panahi

Received: 10 June 2024
Revised: 30 June 2024
Accepted: 6 July 2024
Published: 15 July 2024

1. Introduction

Different researchers proposed the k-means algorithm in the 1950s and 1960s [1]. Those researchers include Lloyd [2], MacQueen [3], Jancey [4], and Steinhaus [5]. Due to the simplicity of implementation and low computational complexity of the k-means algorithm, it has become one of the top ten most popular algorithms in problems requiring clustering and data mining [6,7]. The application areas are data mining [8], cluster analysis [9], data clustering [10], pattern recognition [11], financial risk control [12], data science [13], intelligent marketing [14], and data compression [15,16].

From the recent works [17,18], the following problems with the traditional k-means algorithm remain a hot research topic:

(1) Clustering results are susceptible to noisy data and outliers;
(2) Random selection of initial centers tends to make clustering results locally optimal;
(3) Realistic clustering application scenarios where the number of clusters is predetermined in advance are difficult;
(4) Furthermore, the reliance on the Euclidean distance in the Classic K-means algorithm limits its capacity to discern non-linear cluster shapes or clusters with irregular formations.

In this case, the measure of similarity between samples is the core of the k-means clustering algorithm [19,20]. Therefore, this research attempts to improve the traditional k-means algorithm through a series of data processing methods and innovative similarity metrics to solve the above problems.

The contributions of this research work are as follows: 1. We have solved the aforementioned problems (1), (2), and (3) using data lattice transformation, isometric interpolation, and Fisher optimal segmentation, respectively; 2. The traditional k-means algorithm has been improved to solve the aforementioned problem (4) using contour similarity; and 3. Numerical experiments on clustering have been designed to verify the feasibility of the work performed. The work performed in this research can provide several valuable and implementable methods in handling data outliers and missing values, predetermining of the number of clusters, selecting initial centers in the clustering process, and measuring similarity between samples.

The subsequent sections of this paper are structured as follows: Section 2 categorically discusses the research on improving the k-means algorithm with different sample similarity metrics. Section 3 describes in detail the similarity basics used in this paper. Section 4 presents algorithmic improvement ideas and processes. Section 5 conducts numerical experiments to verify the performance of the proposed algorithm in this paper. Section 6 discusses the limitations and implications of the model proposed in this paper. Finally, Section 7 concludes the paper.

2. Literature Survey

The focus of this study is to improve the robustness of the classical k-means algorithm in recognizing alternative clustering result shapes. To this end, this section summarizes some of the studies relevant to the work of this manuscript and divides these studies into three categories.

The first type is based on improving typical similarity distance determination methods, including Euclidean metrics, Manhattan distance, Minkowski distance, city block, cosine, correlation distance, and so on. Among them, Singh et al. [21] compared the performance results of the k-means algorithm using Euclidean distance, Manhattan distance, and Minkowski distance in the classification process, and the conclusion showed that the distortion degree of the k-means algorithm using Minkowski distance in describing the similarity of the samples is the smallest, but the best classification results were obtained using Euclidean distance, and the k-means algorithm using Manhattan distance had the worst classification results. In Singh et al.'s study, they expressed the need for careful selection of similarity metrics. A. Chakraborty et al. [22] showed the performance results of the k-means algorithm in using city block distance, cosine distance, and correlation distance in classifying the Iris dataset, and the k-means algorithm using city block distance and cosine distance achieved 98% accuracy. F. A. Sebayang et al. [23] used Canberra distance, city block distance, and Euclidean distance in the process of using the k-means algorithm; the results show that the k-means algorithm using different distances on different datasets shows different degrees of superiority of results. This kind of research aims to improve the robustness of the k-means algorithm in different application scenarios of clustering shapes and achieve better real-world results, but ignores the impact of disadvantages (1)–(3) caused by the k-means algorithm, and at the same time for this kind of improvement the research applicant should also have a certain knowledge of the data distribution.

The second type is an improvement of the k-means algorithm based on weighted distances, typically weighted Euclidean distances. Among them, Tang et al. [24] introduced the minimum and maximum principles based on weighted data to balance the effects of distance and density on clustering, as a way to automatically determine the clustering center and the number of centers. Wang et al. [25] used dimensionally weighted Euclidean distance to measure the degree of similarity between samples when improving the k-means algorithm, and to ensure that the algorithm does not fall into the local optimum, they also proposed a new algorithm that jumps out of the local optimum based on the idea of evolutionary algorithms by using stochastic and evolutionary processes [26]. In their discussion on the effect of initial distance centers and outliers on clustering results, Zhang et al. [27] proposed an adaptive k-means variant based on weighted Gaussian distances, which gave better results in dealing with clustering results and the effect of outliers on

clustering results. This kind of research revolves around solving the local optimum of the k-means algorithm and improving the robustness of the clustering shape, which automatically discovers the initial clustering center through the process of determining the weights, and then improves the accuracy of clustering by the weighted distances, which achieves a certain degree of practical application, but it ignores the effects of outliers and noisy data on the clustering results, and it fails to determine the number of clusters in advance very well.

The third type is the study of non-conventional sample similarity, dissimilarity measures, or variations of the k-means algorithm in combination with other methods. Among them, Zhu et al. [17] proposed to improve the k-means algorithm by using evidence distance; they first replaced the original value of sample data with the probability assignment method, then used evidence distance for clustering, and compared the experimental results with the classical k-means algorithm, aggregated distance parameter k-means algorithm, and Gaussian mixture model, and the clustering experimental results are better and the algorithm convergence is also good. Chen and Yang [28] proposed a diffusion k-means algorithm based on diffusion distance to maximize the proximity of sample data from the same cluster class, which has better advantages in dealing with nonlinear datasets and mixed-dimensional data with non-Euclidean geometrical features. Dinh et al. [29] proposed K-CMM fusion interpolation to the clustering process based on the mean and kernel approach, which uses the information-theoretic dissimilarity in measuring the distance metric and squared Euclidean distance. Research in this category has failed to address problems (1), (2), and (3) by redefining the similarity metric and using it in conjunction with the k-means algorithm for clustering, which greatly improves the robustness of the algorithm in detecting the shape of the clusters, but still fails to address disadvantages (1), (2), and (3).

Through recent studies [17,18], it has been found that the traditional k-means algorithm still suffers from four problems that have not been solved systematically, and existing studies have proposed solutions around some of the four problems. Therefore, our goal is to solve the four problems that exist in traditional k-means using different methods, and ultimately integrate them to form a systematic solution to help weaken the influence of outliers and missing values on the clustering results, avoid the choice of initial centers falling into the trap of local optimum, and enhance the robustness of the algorithm to identify the shape of the clustering results.

3. Pan-Factor Space Theory and Contour Similarity

Pan-factor space is a mathematical theory of concept discovery and representation based on theories, factors, and their operations. Therefore, we give the following definition:

Definition 1. *The domain represented by* $U = \{u_i\}_{i=1}^{n}$ *is a non-empty countable set that is composed of all objects studied in a problem study, where* u_i *is the i-th study subject.*

In the process of studying a problem utilizing mathematical tools, it is necessary to resort to some indicators or variables, which in the pan-factor space we call factors, a factor generally denoted as f.

Definition 2. *For* $\forall u_i \in U, i = 1, 2, \cdots, n$, *the existence of a special value* d_i, *satisfied* $d_i = f(u_i)$. *The set* $I_f = \{d_i \mid \forall u_i \in U, d_i = f(u_i)\}$ *is called the phase space of the factor* f.

The set consisting of all factors defined on U is denoted as $\mathcal{F}$. The factor f is a surjection from the domain U to the phase space I_f and there is a possibility that the feature d_i is vacant (d_i is a missing value) during the application, and the missing value is special in the theory of pan-factor spaces.

For the factor f as a special mapping, it exists as a pre-image $\overleftarrow{f}$, defined mathematically by the following:

Definition 3. *The* $\overleftarrow{f}$ *is a mapping from the phase space* I_f *to the power set of* U*, satisfied:*

$$\forall d \in I_f, \overleftarrow{f}(d) = [d]_f \in U, [d]_f \in U/f \subset 2^U$$

where $[d]_f = \left\{ u_i \middle| f(u_i) = d \in I_f, \forall u_i \in U \right\} \subseteq U, i = 1, 2, \cdots, n$*. The* U/f *is the quotient set. Then we called* $\overleftarrow{f}$ *as Recall.*

The purpose of Recall is to form a perception of class in the discourse. Obviously:

$$\begin{aligned} &1.\ \forall d \in I_f, f\left(\overleftarrow{f}(d)\right) = d. \\ &2.\forall [d]_f \in U, \overleftarrow{f}\left(f\left([d]_f\right)\right) = [d]_f. \\ &3.\forall x, y \in I_f, x \neq y, \overleftarrow{f}(x) \neq \overleftarrow{f}(y). \end{aligned}$$

For domains with a finite number of samples, assuming $\mathrm{I}_f = \{d_1, d_2, \cdots, d_s\}$, then the quotient set is

$$U/f = \left\{ [d_1]_f, [d_2]_f, \cdots, [d_s]_f \right\}$$

U/f is a partition of U.

Based on the definitions of factor f and Recall $\overleftarrow{f}$, there are two special factors in the convention:

1. *zero-factor* ***o*****.** The phase space of ***o*** is $I_o = \{NoN\}$, the quotient set of ***o*** is $U/o = U$.
2. *full-factor* ***e*****.** The phase space of ***e*** is $I_e = U$, the quotient set of ***e*** is $U/e = \left\{ \{u\}_{\forall u \in U} \right\}$.

Thus, the following axiom can be obtained:

Axiom 1. *(Discovering Axioms)* $\overleftarrow{o}(NoN) = U, \overleftarrow{e}(x) = \varnothing$

Further, based on the foregoing, we define the logical operations of the factors as follows:

Definition 4. *There are two factors* f *and* g*, if* f *equation to* g*, iff* $I_f = I_g$ *and* $\forall u \in U, \overleftarrow{f}(f(u)) = \overleftarrow{g}(g(u))$*, denoted as* $f = g$*.*

Definition 5. *There are two factors* f *and* g*, for* $\forall u \in U, f(u) = x \in I_f, g(u) = y \in I_g$*. If* $\forall x \in I_f, \exists y \in I_g$*, s.t.* $\overleftarrow{f}(x) \subset \overleftarrow{g}(y)$*, then we called* g *less than* f*, denoted as* $g < f$*.*

According to the Definitions 4 and 5, we can combine $f = g$ with $g < f$ denoted as $g \leq f$.

Definition 6. *For* $\forall u \in U, f(u) = x \in I_f, g(u) = y \in I_g$*,* $h(u) = (x, y) \in I_f \times I_g$*. If* $\forall (x, y) \in I_f \times I_g$*, satisfied:*

$$\overleftarrow{h}(x, y) = \overleftarrow{f}(x) \cap \overleftarrow{g}(y) \tag{1}$$

Then we called h is the Analysis-Factor of f *and* g*, denoted as* $h = f \wedge g$*.*

Definition 7. *For* $\forall u \in U, f(u) = x \in I_f, g(u) = y \in I_g$*,* $l(u) = (x, y) \in I_f \times I_g$*. If* $\forall (x, y) \in I_f \times I_g$*, satisfied:*

$$\overleftarrow{l}(x, y) = \overleftarrow{f}(x) \cup \overleftarrow{g}(y) \tag{2}$$

Then we called the l *is the Composite-Factor of* f *and* g*, denoted as* $l = f \vee g$*.*

Theorem 1. *The algebra system* $(\mathcal{F}, \leq)$ *is a lattice.*

Proof. Firstly, we prove the $\mathcal{F}$ with "$\leq$" is a partially ordered set. According to the Definitions 4 and 5, obviously:$f \leq f$ (Reflexive). If $f \leq g$, $g \leq f \Leftrightarrow f = g$ (Antisymmetric relation). If $f \leq g$, $g \leq h \Leftrightarrow f \leq h$ (Transitivity). Then the $\mathcal{F}$ with "$\leq$" is partially ordered set.

Secondly, for $\forall f, g \in \mathcal{F}$, prove the existence of l.u.b.$\{f, g\}$ and g.u.b.$\{f, g\}$. According to the rules of calculation of the quotient set:

$$U/f \wedge g = U/f \circ U/g, U/f \vee g = U/f + U/g \tag{3}$$

and the existence theorem for upper and lower definite bounds on quotient sets, and according to Definition 5, it is possible to know $f \wedge g$ is the supremum, that is l.u.b.$\{f, g\} = f \wedge g$. $f \vee g$ is the infimum, that is g.u.b.$\{f, g\} = f \vee g$.

So, the algebra system $(\mathcal{F}, \leq)$ is a lattice. □

From Theorem 1 *Analysis-Factor* $\wedge$ and *Composite-Factor* $\vee$ are the natural arithmetic. Further, we can obtain the first absorption law, the law of order, the idempotent law, the law of commutation, the associative law, and the second absorption law of factors f and g on $\wedge$ and $\vee$ arithmetic hold. It can further be demonstrated that the algebra system $(\mathcal{F}, \leq)$ is bounded lattice.

Definition 8. *Let's refer to the combination of the four elements* $\left(U, \mathcal{F}, \mathcal{D}, 2^U\right)$ *as a pan-factor space, where* $\mathcal{F}$ *is a set that consists of all factors defined on* U. $\mathcal{D}$ *is the data space that be constructed by all factors, denoted as:*

$$\mathcal{D} = \prod_{\forall f_i \in \mathcal{F}} I_{f_i}$$

In practice, the number of factors and the number of objects of study in the domain are limited. Denote the set of finite factors taken out of $\mathcal{F}$ as $\mathcal{F}_n = \{f_i | f_i \in \mathcal{F}\}_{i=1}^n$. Denote $\mathcal{F}_n^*$ as the set consisting of all factors obtained from $\mathcal{F}_n$ and the factors contained in $\mathcal{F}_n$ on operations $\wedge$ and $\vee$. We can prove that $(\mathcal{F}_n^*; \wedge, \vee)$ is a sublattice of $(\mathcal{F}; \wedge, \vee)$.

Based on the foregoing, we give the following definition:

Definition 9. $\{o; \mathcal{F}_n\}$ *is called the finite factor scalar frame or lattice coordinate frame. The corresponding Cartesian product* $\mathcal{D}^* = I_1^* \times I_2^* \times \cdots \times I_n^*$ *is the lattice coordinate system, where* $I_j^*, j = 1, 2, \cdots, n$, *is the normalized phase space of* I_{f_i} *by* NoN.

According to Definition 9, if $I_{f_i} = \left\{x_1, x_2, \cdots, x_{n_j}\right\}$, then the normalized phase space I_j^* is $I_j^* = \left\{0, 1, 2, \cdots, n_j\right\}$.

The difference between the lattice coordinate system and the affine coordinate system is that the affine coordinate system cannot uniformly handle data of different metric scales.

Definition 10. *For* $f_1, f_2, \cdots, f_n \in \mathcal{F}_n, \forall u \in U$, *the*

$$(f_1, f_2, \cdots, f_n)(u) = (x_1, x_2, \cdots, x_n) \in \mathcal{D}^* \tag{4}$$

are called the lattice coordinates of the object u (abbr. **L.c***).*

The **L.c** has the following properties:

(1) The coordinates $f_j(u_i) = x_{ij}$ of the object $u_i \in U$ on the factor $f_j \in u_i$ are natural numbers and when $x_{ij} = 0$ represent no actual statistical values, i.e., missing values.

(2) For $\forall f_k(u_i) = x_{ik}, f_k(u_j) = x_{jk} \in I_k^*$, the magnitudes of x_{ik} and x_{ik} represent only the order of precedence and have no quantitative meaning. In general, $x_{ik} + x_{ik}$ is

meaningless, but $x_{ik} - x_{ik}$ represents the potential difference between two objects u_i and u_j on factor f_k.

(3) For $\forall f_j(u_i) = x_{ij} \in I_j^*, f_k(u_i) = x_{ik} \in I_k^*$, x_{ij} and x_{ik} are not directly comparable.

Therefore, before similarity analyses are performed, the data corresponding to the original samples need to be converted to the corresponding "lattice point" data in **L.c**. This step of data preprocessing is known as the lattice transformation. The detailed conversion process is described in detail in Section 4.1.

Going from multidimensional data to low dimensional data is a common means of data analysis, but going from low to high dimensions represents the ability to obtain more information, with commonly used means such as kernel transformations. Such a means is lacking for the process of similarity metrics.

In the process of describing the similarity of samples, there are no more than four perspectives: first, describing the degree of mutual influence between samples, such as the inner product. The second is to compare the sizes, shapes, volumes, or areas of the samples and then determine the degree of similarity between the two samples, e.g., the 2-paradigm number. The third is to describe the similarity between samples from the distance between them within a coordinate system, such as the Minkowski distance. Fourthly, the similarity of the samples is described from the perspective of the sameness of the objects with which the samples interact, e.g., correlation coefficient. In this paper, according to the idea and principle of simple shape in topology, under the framework of lattice coordinates, we construct the factor contour indexes describing the basic shape of the sample data, and on this basis, we construct the variation of the degree of similarity between the factor contours of the samples and their algorithms.

Definition 11. *For* $x_{1\times n} = (x_1, x_2, \cdots, x_n) \in \mathcal{D}^*$, *define:*

$$q_{1\times n} = x_{1\times n} P_{n\times n}$$

is the factor contour transformation of data $x_{1\times n}$ *(abbr.* $q^{(x)}$*), where* $q_{1\times n}$ *is the factor contour that represents the shape of data* $x_{1\times n}$ *in* **L.c.** $P_{n\times n}$ *is the contour operator matrix with the following matrix form:*

$$P_{n\times n} = \begin{pmatrix} -1 & 0 & \cdots & 0 & 1 \\ 1 & -1 & \cdots & 0 & 0 \\ 0 & 1 & -1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & -1 \end{pmatrix}$$

Obviously, for $\forall x_{1\times n} = (x_1, x_2, \cdots, x_n) \in \mathcal{D}^*$, *the factor contour transformation* $q^{(x)} = (x_2 - x_1, x_3 - x_2, \cdots, x_n - x_1, x_1 - x_n)$ *is ascending dimensions transform of data. The set* $\mathcal{L} = \left\{ q^{(x)} \middle| q^{(x)} = x_{1\times n} P_{n\times n}, \forall x_{1\times n} \in \mathcal{D}^* \right\}$ *be called Factor Contour Space.*

Definition 12. *Let* $q^{(x)}, q^{(y)} \in \mathcal{L}, \lambda = (\lambda_1, \lambda_2, \cdots, \lambda_n) \in R_+^n$, *if*

$$q^{(y)} = \lambda \circledast q^{(x)} = (\lambda_1 x_1 - \lambda_2 x_2, \lambda_1 x_2 - \lambda_2 x_3, \cdots, \lambda_1 x_n - \lambda_2 x_1) \tag{5}$$

then $q^{(y)}$ *is the zoom of* $q^{(x)}$, *the vector* λ *is the zoom-operator.*

Definition 13. *Let* $q^{(y)} = \lambda \circledast q^{(x)}, \lambda = (\lambda_1, \lambda_2, \cdots, \lambda_n) \in R_+^n$, *define:*
If $\forall \lambda_k = 1, k = 1, 2, \cdots, n$, *then* $q^{(y)} = q^{(x)}$.
If $\forall \lambda_k = m > 0, k = 1, 2, \cdots, n$, *then* $q^{(y)} \parallel q^{(x)}$.
If $\forall \lambda_k > 1, k = 1, 2, \cdots, n$, *then* $q^{(y)} \succ q^{(x)}$.
If $\forall \lambda_k < 1, k = 1, 2, \cdots, n$, *then* $q^{(y)} \prec q^{(x)}$.

If not, then $q^{(y)} \gtrless q^{(x)}$.

The ordinal structure of the factor contour space will not be discussed in depth.

Definition 14. *Let* $x, y, z \in \mathcal{D}^*; q^{(x)}, q^{(y)}, q^{(z)} \in \mathcal{L}$, *for the functional*

$$\rho : \mathcal{L} \times \mathcal{L} \to [0, 1]$$

If the ρ satisfies the following conditions:

(1) Boundedness $0 \leq \rho\left(q^{(x)}, q^{(y)}\right) \leq 1$;

(2) Regularity $\rho\left(q^{(x)}, q^{(x)}\right) = 1$;

(3) Symmetry $\rho\left(q^{(x)}, q^{(y)}\right) = \rho\left(q^{(y)}, q^{(x)}\right)$;

(4) Rank Preservation if $q^{(y)} \succ q^{(x)} \succ q^{(z)}$, then $\rho\left(q^{(y)}, q^{(x)}\right) > \rho\left(q^{(y)}, q^{(z)}\right)$.

Then ρ is said to be a similarity measure on $\mathcal{L}$. $\rho\left(q^{(x)}, q^{(y)}\right)$ is the contour similarity between $q^{(x)}$ and $q^{(y)}$, denoted $\rho^{(xy)}$.

In Definition 14, the closer the value of $\rho^{(xy)}$ is to 1, the more similar the two factor contours $q^{(x)}$ and $q^{(y)}$. We can construct a variety of profile similarities that conform to Definition 12 and Definition 14. In Section 4.2, we give an algorithm for the computation of factor contour similarity.

4. Algorithmic Ideas and Improvements

The design thought process of the improved Csk-means algorithm is shown below (Figure 1).

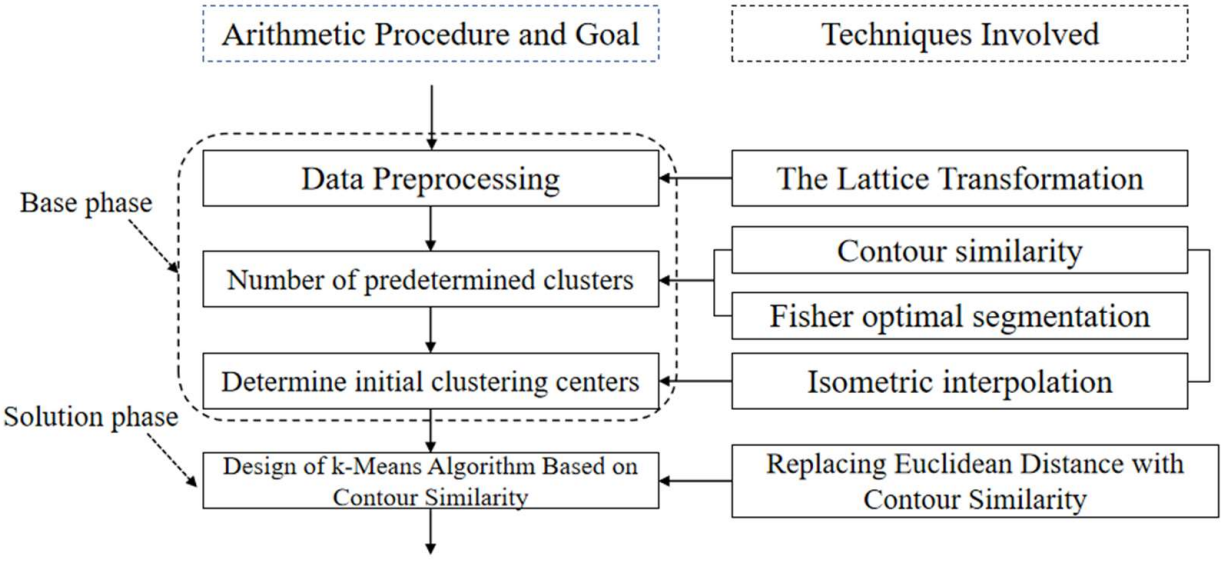

Figure 1. The Csk-means algorithm design process.

In particular, the basic data format obtained herein is illustrated by the following images (Figure 2).

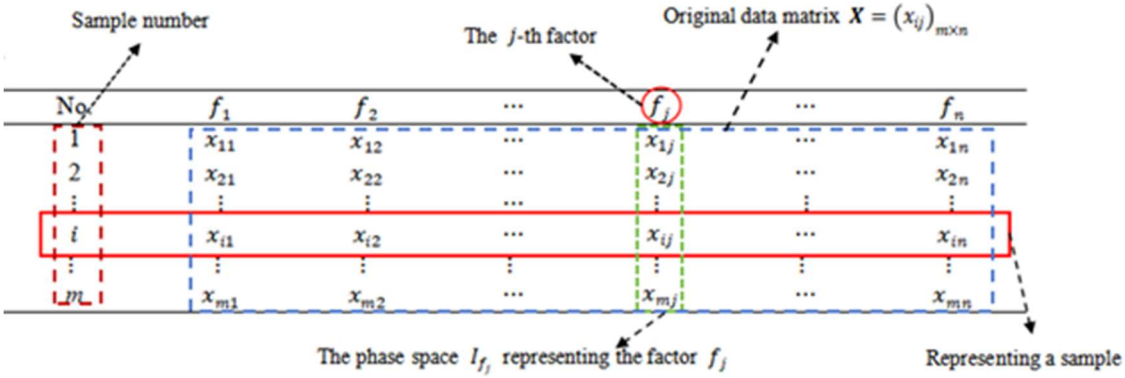

Figure 2. Basic data format image presentation. The dark red dashed box represents the number of the sample data. The red rounded box represents the *j*th conditional factor described in the previous section. The red box represents the phase distribution of the *i*th sample under each factor; and the blue dashed box represents the original data matrix consisting of all the samples.

4.1. The Lattice Transformation Process

The data lattice transformation is a data preprocessing process on the original data matrix $X = (x_{ij})_{m\times n}$, aiming at establishing a unified reference point for each factor, transforming the missing values of the data into the origin of the lattice coordinate system, and integrating the original data. That is, there exists a one-to-one mapping:

$$I_{f_j} \to N_j = \{0, 1, 2, \cdots\}, j = 1, 2, \cdots, n,$$

In this paper, the steps to perform the data lattice transformation are as follows:

(1) For the original data matrix $X = (x_{ij})_{m\times n}$, extract the number m of the matrix's rows and the number n of the matrix's columns.

(2) Find the reference point vector

$$\mathrm{INF} = \left(\inf\left(I_{f_1}\right), \inf\left(I_{f_2}\right), \cdots, \inf\left(I_{f_n}\right)\right)$$

where $\inf(I_j), j = 1, 2, \cdots, n$ represents the Infimum of the phase space I_{f_j} of the factor f_j. If the factor f_j is a continuous variable and $\inf(I_j)$ is unknown, let

$$\inf(I_j) = \min_{\forall i}\{x_{ij}\} - \varepsilon_j, \varepsilon_j \in (0, 1), i = 1, 2, \cdots, m.j = 1, 2, \cdots, n.$$

where ε_j is the slack variable.

(3) For $\forall x_{ij} \in I_{f_j}$, let

$$y_{ij} = \begin{cases} 0, if\ x_{ij} = NoN \\ (x_{ij} - \inf(I_j)) \times 10^{k_j} + 1, if \neq NoN' \end{cases}, i = 1, 2, \cdots, m.j = 1, 2, \cdots .n.$$

where NoN represents that x_{ij} is the missing value, k_j is the scale-accurate parameters (it can be viewed as the number of decimal places in the decimal portion of the data).

After the previous three steps, the original data matrix $X = (x_{ij})_{m\times n}$ is lattice transformed into a matrix $Y = (y_{ij})_{m\times n}$.

4.2. The Similarity Calculation Process of Contour Similarity

For the data matrix $Y = (y_{ij})_{m\times n}$ that has been lattice transformed, the procedure for calculating the degree of similarity between samples based on contour similarity is as follows:

(1) Calculate the scale vector

$$S = (\sup(M_1), \sup(M_2), \cdots, \sup(M_n))$$

where $\sup(M_i) = \max_{\forall j}\{y_{ij}\} + \delta_j, i = 1, 2, \cdots, n.\ j = 1, 2, \cdots, m$. If the factor f_j is a continuous variable, then $\delta_j > 0$, otherwise $\delta_j = 0$.

(2) Generate the contour data matrix

$$Z_{m\times n} = Y(R - E)$$

where

$$R_{n\times n} = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & 0 & 1 \\ 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & & \vdots & \\ 0 & 0 & 0 & \cdots & 1 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 & 0 \end{pmatrix}_{n\times n}$$

E is a unit matrix with dimension $n \times n$.

(3) Measure the potential of each sample

$$W = |DY|(\mathrm{diag}(S))^{-1}$$

where $|DY|$ represents taking the absolute value for each element of the matrix DY,

$$D_{\frac{(m-1)m}{2}\times m} = \begin{pmatrix} -1 & 1 & 0 & \cdots & 0 & 0 & 0 \\ -1 & 0 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ -1 & 0 & 0 & \cdots & 0 & 0 & 1 \\ 0 & -1 & 1 & \cdots & 0 & 0 & 0 \\ 0 & -1 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ 0 & -1 & 0 & \cdots & 0 & 0 & 1 \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 1 & 0 \\ 0 & 0 & 0 & \cdots & -1 & 0 & 1 \\ 0 & 0 & 0 & \cdots & 0 & -1 & 1 \end{pmatrix} \begin{matrix} \\ \\ \\ m-1\ rows \\ \\ \\ m-2\ rows \\ \\ \\ \\ 2\ rows \\ \\ 1\ row \end{matrix}$$

$\mathrm{diag}(S)$ represents the expansion of the scale vector S into a diagonal matrix.

(4) Calculate the pose metric for each sample.

$$H = |DZ|(\mathrm{diag}(S(R + E)))^{-1}$$

(5) Calculate the factor contour distance matrix for the sample.

$$P = \mathrm{sum}\left((W * H)^T\right)/n$$

where $W * H$ represents the Hadamard product of the matrix W and H, $\mathrm{sum}\left((W * H)^{\mathrm{T}}\right)$ stands for summing the columns of the matrix $(W * H)^{\mathrm{T}}$.

(6) Calculate the contour similarity matrix between individual samples.

$$M = \mathrm{sum}\left(((1 - W) * (1 - H))^T\right)/n$$

The concept of contour similarity was first proposed in [30] and a simple calculation case is given; the specific application is performed in [31]. The contour similarity calculation step proposed here is an improvement and enhancement, which is more convenient to calculate.

In particular, for the process of predetermining the number of clusters, which aims to predetermine the number of classifications, the following strategy is adopted to build a sample similarity neighbor vector: Find the two most similar samples a and b using contour similarity, record their contour similarity as the initial value of the similarity neighbor vector, use one of the sample points (assumed to be b) as the datum point, find the most similar sample point c among the remaining samples using contour similarity, and append the contour similarity of these two to the similarity neighbor vector. Update the reference point b and repeat the process until all samples are traversed. The number of clusters is found using Fisher optimal partitioning based on the sample similarity neighbor vector.

4.3. Design of k-Means Algorithm Based on Contour Similarity

The classical k-means algorithm uses Euclidean distance to determine the similarity between samples, assumes that the sample needs to be divided into k categories, and its basic algorithmic flow is as follows:

(1) Randomly select k samples in the sample data set as the initial center of mass;
(2) The distance from each sample to these k centers of mass is calculated using the Euclidean distance;
(3) Divide each sample into the nearest center of mass to form a collection of classes;
(4) Update the center of mass of the k classes;
(5) Repeat steps 1–4 until the number of iterations is reached or the center of mass no longer changes between the two preceding and following times;
(6) Output the k classes of the division.

The idea of algorithm improvement in this paper lies in the use of contour similarity instead of Euclidean distance for the judgment of similarity between samples. We call this improved algorithm CSk-means. It should be noted that the contour similarity calculation between the samples needs to be carried out on the original data matrix formed by the lattice transformation is the data matrix $Y = \left(y_{ij}\right)_{m\times n}$. The pseudo-code of the CSk-means is summarized as Algorithm 1.

Algorithm 1: The CSk-means.

Input: The lattice transformed matrix $Y = (y_1, y_2, \cdots, y_m)^T$ (where $y_i = (y_{i1}, y_{i2}, \cdots, y_{in}), i = 1, 2, \cdots, m$ represents a vector of sample data), k value, Maximum Iterations
Output: The k classes that have been classified.
1. Select k sample points as the initial clustering centers.
2. **repeat**
3. **for** $i = 1, 2, \cdots, m$
4. Calculate the contour similarity from each sample point y_i to each class center.
5. Class labeling is decided based on maximum contour similarity.
6. Divide the sample points into the appropriate clusters.
7. **end for**
8. Updating the clustering center.
9. **until** the maximum number of iterations is reached or the clustering center no longer changes.

Algorithm 1 is also affected by the initial clustering centers; for this reason, the initial clustering center selection in this paper takes the following strategy:

Use contour similarity to find the two least similar sample points, and use these two sample points as the reference points. If the number of clusters needed is two, then use these two points as the initial set of clustering centers; if the number of clusters needed is three, then use contour similarity to find the points that are similar to both of these two sample points, and insert the sample points into the initial set of clustering centers; if the

number of clusters needed is four, then use these three sample points as the reference points, use contour similarity to find the sample points that are similar to all three sample points, and insert the sample points found into the initial set of clustering centers. If the number of classes to be clustered is four, use the three sample points as the reference point, use the contour similarity to find the sample point that is similar to all three sample points, and insert this sample point into the initial clustering center set. Repeat this process until the initial set of clustering centers is found that meets the required number of clusters. We call this process isometric interpolation.

5. Experimental Validation of the Csk-means Algorithm

To evaluate the performance of the proposed CSk-means algorithm, this study downloaded six datasets used as testing data from the UCI Machine Learning Repository (https://archive.ics.uci.edu/, accessed on 21 April 2024). The experimental environment is a 12th Gen Intel(R) Core(TM) i5-12500 3.00 GHz processor, NVIDIA GeForce RTX 3080 graphics card, 16.0 GB of running memory, Windows 11 operating system, and programming with MATLAB 2023b-64bits. Meanwhile, the whole experiment is conducted based on the MATLAB 2023b platform. The comparison algorithms are the traditional k-means algorithm, hierarchical clustering model, spectral clustering, and k-means++. The code for the Csk-means algorithm that needs to be validated is custom code.

Table 1 describes the basic statistical characterization information of the above six datasets. In this paper, these datasets are used for validating the performance of the proposed algorithm, and in this process, there will be no division between the training set and the testing set. The main reasons for the selection of the reported datasets were: (1) the ability to characterize the validity and reliability of the algorithm using validated metrics; and (2) the need for the improved clustering algorithm to focus on real clustering scenarios, taking into account both varying degrees of complexity as well as varying degrees of distribution, which is reflected in the categorical and sample sizes of the dataset as well as in the characteristics of the data distribution.

Table 1. The information of six datasets was downloaded from the UCI Machine Learning Repository.

Datasets	No. of Samples	No. of Features	No. of Clusters
Iris [32]	150	4	3
Wine [33]	178	13	3
Ecoli [34]	336	8	8
Seeds [35]	420	7	3
Yeast [36]	1484	8	10
Abalone [37]	4177	8	28

The metrics for evaluating the performance of the model were chosen as Precision, Recall, Accuracy, Running Time, Number of Iterations, and Normalized Mutual Information (NMI).

Table 2 demonstrates the iteration time, number of iterations, and the degree of accuracy of the clustering results for the clustering process of the Csk-means algorithm and the classical k-means algorithm on the six datasets. Table 2 demonstrates the iteration time, number of iterations, and the degree of accuracy of the clustering results for the clustering process of the Csk-means algorithm and the classical k-means algorithm on the five datasets. The results show that the Csk-means algorithm takes more time than the k-means algorithm for the clustering process since the gridded transformation of the Csk-means algorithm takes more time. However, the number of iterations and the clustering results are better than the classical k-means algorithm.

Table 2. The number of iterations, the running time (s), and the accuracy for classifying the six datasets.

	CSk-Means	k-Means	CSk-Means	k-Means	CSk-Means	k-Means
Datasets	**Time Spent**		**Number of Iterations**		**Accuracy**	
Iris	0.0076	0.0064	3	4	0.92	0.89
Wine	0.0754	0.0182	7	5	0.94	0.70
Ecoli	0.0822	0.0418	10	23	0.76	0.58
Seeds	0.4331	0.2781	6	7	0.90	0.89
yeast	0.1946	0.0358	10	40	0.42	0.40
Abalone	0.2286	0.2092	30	54	0.18	0.22

Table 3 demonstrates the F-measure, Precision, and Recall metric values for the clustering results of the CSk-means algorithm and classical k-means algorithm on the six datasets. It can be found that the CSk-means algorithm is more stable than the classical k-means algorithm. The overall performance of the clustering of the CSk-means algorithm is better than that of the classical k-means algorithm. It is noteworthy that the datasets Yeast and Abalone are less accurate, and we analyze in depth the reasons for this phenomenon: the two datasets contain a large number of classes and the degree of differentiation between samples in the different classes is not clear enough, and the data distribution morphology shows a strong consistency. Methods used in this process include, but are not limited to, principal component analysis, data distribution, and other clustering models.

Table 3. The F-measure (F), Precision (P), and Recall (R) for classifying the six datasets.

	CSk-Means	k-Means	CSk-Means	k-Means	CSk-Means	k-Means
Datasets	**F**		**P**		**R**	
Iris	0.94	0.92	0.94	0.88	0.94	0.98
Wine	0.94	0.78	0.95	0.83	0.97	0.74
Ecoli	0.76	0.57	0.76	0.90	0.76	0.42
Seeds	0.91	0.91	0.92	0.91	0.91	0.92
yeast	0.40	0.35	0.35	0.34	0.50	0.36
Abalone	0.30	0.24	0.33	0.29	0.28	0.21

Figure 3 and Table 4 show the accuracy, F-measure, Precision, Recall, and NMI values of the clustering results of the Csk-means algorithm, hierarchical clustering, and spectral clustering on the six datasets. In comparison, the Csk-means algorithm shows better stability and more accurate clustering results.

Table 4. Evaluation results of Csk-means algorithm, hierarchical clustering, and spectral clustering on six datasets.

	Csk-Means				Hierarchical Clustering				Spectral Clustering				k-Means++			
Datasets	**Accuracy**	**F**	**P**	**R**	**Accuracy**	**F**	**P**	**R**	**Accuracy**	**F**	**P**	**R**	**Accuracy**	**F**	**P**	**R**
iris	0.92	0.94	0.94	0.94	0.35	0.51	0.34	1.00	0.91	0.94	0.88	1.00	0.91	0.92	0.88	0.98
wine	0.94	0.96	0.95	0.97	0.43	0.60	0.61	0.58	0.32	0.49	0.33	0.93	0.74	0.80	0.86	0.76
Ecoli	0.76	0.76	0.76	0.76	0.45	0.61	0.44	0.98	0.56	0.55	0.85	0.41	0.60	0.57	0.91	0.43
Seeds	0.90	0.91	0.92	0.91	0.92	0.94	0.93	0.95	0.90	0.92	0.91	0.94	0.91	0.93	0.93	0.94
yeast	0.42	0.40	0.35	0.50	0.32	0.48	0.31	0.97	0.29	0.01	0.05	0.00	0.41	0.36	0.35	0.36
Abalone	0.18	0.30	0.33	0.28	0.22	0.33	0.29	0.36	0.13	0.22	0.31	0.17	0.23	0.25	0.30	0.21

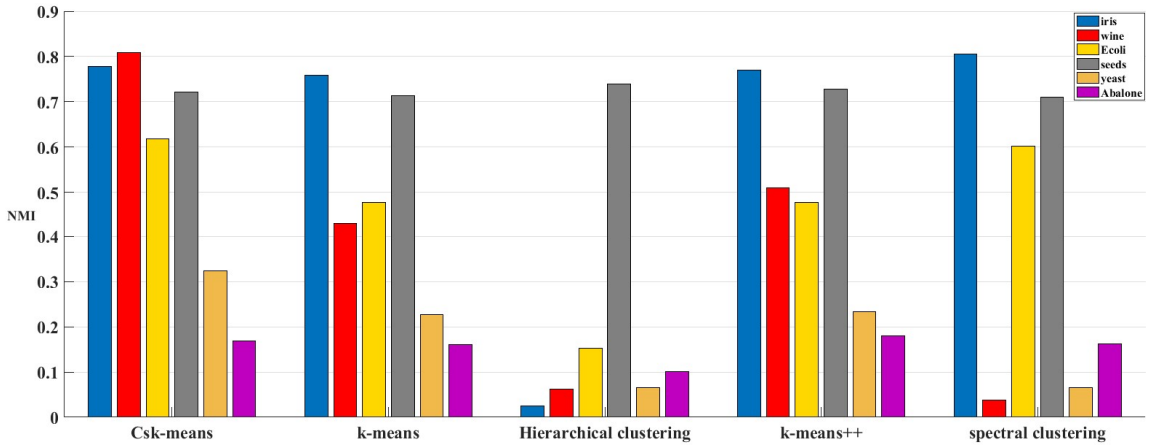

Figure 3. Comparison of NMI of the Csk-means algorithm with other clustering algorithms on 6 datasets.

6. Discussion

To systematically solve the four problems involved in the traditional k-means algorithm proposed in the recent research literature, we propose the Csk-means algorithm. Experimental results demonstrate the effectiveness and potential of the model. Based on the process and related techniques involved in the proposed Csk-means algorithm, in this section, we discuss the corresponding key findings, significance, and limitations of each of the techniques involved in the Csk-means algorithm in the order of the process.

To eliminate the negative effects of outliers and missing values on the clustering results, we redefine the preprocessing of the data, involving the method of lattice transformation. The lattice transform essentially constructs a new system of lattice coordinates, transforming the original continuous data to discrete lattice coordinates and transforming the missing values to the origin of the lattice coordinates. The lattice transform can be interpreted as an integer or discrete transform. The lattice transform may provide a new strategy and method for discretization of data, handling of missing values, and outliers. The advantage of the lattice transform is that it can reduce the adverse effects of outliers and missing values on the model results to a certain extent, which can be supported by some of the experimental results of the study. However, its non-ignorable limitation is that the lattice transform is a homomorphic migration transform, and the adverse effect of outliers still exists to a certain extent. In addition, the degree of discretization of the lattice transform is insufficient when the data is discretized, which may lead to poor generalization of the mined knowledge if the lattice transform is applied in the data preprocessing of rule learning.

According to the idea of hierarchical clustering, in the process of predetermining the number of clusters, the strategy we adopted is to use the contour similarity to build a vector of contour similarity between samples and use Fisher optimal segmentation to realize the predetermination of the number of clusters. From the results, the number of clusters predetermined by this strategy is consistent with the actual number. This strategy also provides a new idea and method for predetermining the number of classes in clustering applications. However, its limitation is obvious: the predetermination of the number of classes in a large sample dataset will take more time, which is not conducive to the layout of the online algorithm.

For the selection of initial clustering centers, we adopt the method of isometric interpolation, which is a new method for determining the initial clustering centers. Isometric interpolation implies the idea of dissimilarity between classes, which is essentially a balanced iterative search. Isometric interpolation can prevent the clustering model from falling into the trap of local optimality and also provides a new clustering strategy, which will be the focus of our next work. However, its limitation is obvious: in the process of determining the initial clustering centers, the selected initial clustering centers need to be evaluated for their balance with the centers of other classes. The adoption of the balancing strategy will

deeply affect the selection of clustering centers, which in turn leads to fluctuations in the clustering results. Although the clustering centers can be determined by a trial-and-error strategy, the time cost will increase.

We improved the traditional k-means algorithm by using contour similarity instead of Euclidean distance to enhance the ability of the k-means algorithm to recognize different clustered shapes. From the experimental results, such an improvement is effective. Contour similarity is a new measure of similarity between samples, which is essentially an upscaling transformation. Compared with other similarity measures, contour similarity does not result in information loss due to data dimensionality reduction, which makes the description of similarity inaccurate. We believe that contour similarity enriches the methodological theory of similarity measures. However, its limitations are: when using contour similarity, one should have at least three samples; at the same time, its calculation process is relatively complicated, which will lead to an increase in time cost, which can be known from the results of the designed experimental comparison.

Although the improved Csk-means algorithm can lead to a certain degree of improvement in the accuracy of the clustering results, we also need to see its limitations and implications, especially the increase in the time cost of the Csk-means algorithm during the clustering process. The change in the data lattice, the selection of the initial clustering centers, and the computation of the similarity between the samples involved in the Csk-means algorithm increase the time cost. If the calculation of contour similarity to inter-sample similarity is further optimized, and the effect of outliers on the clustering results can be ignored, then the lattice transformations of data can be eliminated from the Csk-means algorithm, thus the Csk-means algorithm reduces the time cost.

In addition, the choice of comparison algorithms in our designed comparison experiments contains the traditional k-means algorithm, hierarchical clustering, and spectral clustering, which aims at verifying the feasibility of the proposed method. However, the comparison with other clustering models, such as SOM, was neglected. Therefore, in future studies, we will include these models to provide a more comprehensive analysis of the practical applications of clustering models.

7. Conclusions

In the clustering process using the traditional k-means algorithm, to predetermine the number of clusters, weaken the adverse effects of outliers and noise on the clustering results, avoid falling into the local optimum due to the random selection of the initial center, and improve the robustness of the algorithm in detecting the shape of the clusters, this paper proposes the Csk-means algorithm based on the k-means algorithm. The experimental results show that predetermining the number of clusters is still a thorny problem, the adverse effect of the selection of the initial center on the clustering results still exists, but the adverse effect of the outliers and the noise data on the clustering results has been weakened to a certain extent, and the Csk-means algorithm effectively improves the accuracy of the clustering results.

Author Contributions: J.Z.: Writing—original draft, Writing—review and editing, Methodology. Y.B.: Editing, Methodology; D.L. and X.G.: Investigation. J.Z., Y.B., D.L. and X.G. report financial support was provided by the Department of Education of Guizhou Province and the Department of Science and Technology of Guizhou. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Project for Growing Youth Talents of the Department of education of Guizhou Province (Qianjiaoji [2022] No.378,377,380,386; Qianjiaoji [2024] No.234,236,238); the Foundation Project for Talents of Qiannan Science and Technology Cooperation Platform Supported by the Department of Science and Technology, Guizhou ([2019]QNSYXM-05); the Guizhou Provincial Department of Education 2024 Humanities and Social Sciences Research Program for Colleges and Universities (2024RW101).

Data Availability Statement: These data were derived from the following resources available in the public domain: https://archive.ics.uci.edu/, accessed on 21 April 2024.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Pérez-Ortega, J.; Almanza-Ortega, N.N.; Vega-Villalobos, A.; Pazos-Rangel, R.; Zavala-Díaz, C.; Martínez-Rebollar, A. The K-means algorithm evolution. *Introd. Data Sci. Mach. Learn.* **2019**, 69–90. [CrossRef]
2. Lloyd, S. Least squares quantization in PCM. *IEEE Trans. Inf. Theory* **1982**, *28*, 129–137. [CrossRef]
3. MacQueen, J. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*; University of California Press: Berkeley, CA, USA, 1967; Volume 1, No. 14; pp. 281–297.
4. Jancey, R.C. Multidimensional group analysis. *Aust. J. Bot.* **1966**, *14*, 127–130. [CrossRef]
5. Steinhaus, H. Sur la division des corps matériels en parties. *Bull. Acad. Polon. Sci.* **1956**, *1*, 801.
6. Kapoor, A.; Singhal, A. A comparative study of K-Means, K-Means++ and Fuzzy C-Means clustering algorithms. In Proceedings of the 2017 3rd International Conference on Computational Intelligence & Communication Technology (CICT), Ghaziabad, India, 9–10 February 2017; pp. 1–6. [CrossRef]
7. Ezugwu, A.E.-S.; Agbaje, M.B.; Aljojo, N.; Els, R.; Chiroma, H.; Elaziz, M.A. A Comparative Performance Study of Hybrid Firefly Algorithms for Automatic Data Clustering. *IEEE Access* **2020**, *8*, 121089–121118. [CrossRef]
8. Annas, M.; Wahab, S.N. Data Mining Methods: K-Means Clustering Algorithms. *Int. J. Cyber IT Serv. Manag.* **2023**, *3*, 40–47. [CrossRef]
9. Hu, H.; Liu, J.; Zhang, X.; Fang, M. An Effective and Adaptable K-means Algorithm for Big Data Cluster Analysis. *Pattern Recognit.* **2023**, *139*, 109404. [CrossRef]
10. Mussabayev, R.; Mladenovic, N.; Jarboui, B.; Mussabayev, R. How to use K-means for big data clustering? *Pattern Recognit.* **2023**, *137*, 109269. [CrossRef]
11. Theodoridis, S.; Koutroumbas, K. *Pattern Recognition*, 3rd ed.; Academic Press: Cambridge, MA, USA, 2006.
12. Guedes, P.C.; Müller, F.M.; Righi, M.B. Risk measures-based cluster methods for finance. *Risk Manag.* **2023**, *25*, 4. [CrossRef]
13. Yudhistira, A.; Andika, R. Pengelompokan Data Nilai Siswa Menggunakan Metode K-Means Clustering. *J. Artif. Intell. Technol. Inf.* **2023**, *1*, 20–28. [CrossRef]
14. Navarro, M.M.; Young, M.N.; Prasetyo, Y.T.; Taylar, J.V. Stock market optimization amidst the COVID-19 pandemic: Technical analysis, K-means algorithm, and mean-variance model (TAKMV) approach. *Heliyon* **2023**, *9*, 2–3. [CrossRef]
15. Foster, J.; Gray, R.; Dunham, M. Finite-state vector quantization for waveform coding. *IEEE Trans. Inf. Theory* **1985**, *31*, 348–359. [CrossRef]
16. Liaw, Y.-C.; Lo, W.; Lai, J.Z. Image restoration of compressed image using classified vector quantization. *Pattern Recognit.* **2002**, *35*, 329–340. [CrossRef]
17. Zhu, A.; Hua, Z.; Shi, Y.; Tang, Y.; Miao, L. An improved K-means algorithm based on evidence distance. *Entropy* **2021**, *23*, 1550. [CrossRef] [PubMed]
18. Ikotun, A.M.; Ezugwu, A.E.; Abualigah, L.; Abuhaija, B.; Heming, J. K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. *Inf. Sci.* **2023**, *622*, 178–210. [CrossRef]
19. Jain, A.K. Data clustering: 50 years beyond K-means. *Pattern Recognit. Lett.* **2010**, *31*, 651–666. [CrossRef]
20. Li, Y.; Wu, H. A clustering method based on K-means algorithm. *Phys. Procedia* **2012**, *25*, 1104–1109. [CrossRef]
21. Singh, A.; Yadav, A.; Rana, A. K-means with Three different Distance Metrics. *Int. J. Comput. Appl.* **2013**, *67*, 1–3. [CrossRef]
22. Chakraborty, A.; Faujdar, N.; Punhani, A.; Saraswat, S. Comparative Study of K-Means Clustering Using Iris Data Set for Various Distances. In Proceedings of the 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 29–31 January 2020.
23. Sebayang, F.A.; Lydia, M.S.; Nasution, B.B. Optimization on Purity K-Means Using Variant Distance Measure. In Proceedings of the 2020 3rd International Conference on Mechanical, Electronics, Computer, and Industrial Technology (MECnIT), Medan, Indonesia,, 25–27 June 2020; pp. 143–147.
24. Tang, Z.K.; Zhu, Z.Y.; Yang, Y.; Caihong, L.; Lian, L. DK-means algorithm based on distance and density. *Appl. Res. Comput.* **2020**, *37*, 1719–1723.
25. Wang, Z.L.; Li, J.; Song, Y.F. Improved K-means algorithm based on distance and weight. *Comput. 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. Zhang, Y.; Zhang, D.; Shi, H. K-means clustering based on self-adaptive weight. In Proceedings of the 2012 2nd International Conference on Computer Science and Network Technology, Changchun, China, 29–31 December 2012; IEEE: Piscataway, NJ, USA, 2012.
28. Chen, A.; Yang, Y. Diffusion K-means clustering on manifolds: Provable exact recovery via semidefinite relaxations. *Appl. Comput. Harmon. Anal.* **2021**, *52*, 303–347. [CrossRef]

29. Dinh, D.-T.; Huynh, V.-N.; Sriboonchitta, S. Clustering mixed numerical and categorical data with missing values. *Inf. Sci.* **2021**, *571*, 418–442. [CrossRef]
30. Bao, Y. Contour similarity and metric of samples of finite dimensional state vector. *J. Liaoning Tech. Univ.* **2011**, *30*, 603–660.
31. Zhao, F.; Sun, M.; Bao, Y. Similarity Measure of Geometric Contours about Multi-Sale Data and Its Application. *Math. Pract. Theory* **2013**, *43*, 178–182.
32. Fisher, R.A. Iris. UCI Machine Learning Repository. 1988. Available online: https://archive.ics.uci.edu/dataset/53/iris (accessed on 21 April 2024).
33. Aeberhard, S.; Forina, M. Wine. UCI Machine Learning Repository. 1991. Available online: https://archive.ics.uci.edu/dataset/109/wine (accessed on 21 April 2024).
34. Nakai, K. Ecoli. UCI Machine Learning Repository. 1996. Available online: https://archive.ics.uci.edu/dataset/39/ecoli (accessed on 21 April 2024).
35. Charytanowicz, M.; Niewczas, J.; Kulczycki, P.; Kowalski, P.; Lukasik, S. Seeds. UCI Machine Learning Repository. 2012. Available online: https://archive.ics.uci.edu/dataset/236/seeds (accessed on 21 April 2024).
36. Nakai, K. Yeast. UCI Machine Learning Repository. 1996. Available online: https://archive.ics.uci.edu/dataset/110/yeast (accessed on 21 April 2024).
37. Nash, W.; Sellers, T.; Talbot, S.; Cawthorn, A.; Ford, W. Abalone. UCI Machine Learning Repository. 1995. Available online: https://archive.ics.uci.edu/dataset/1/abalone (accessed on 21 April 2024).