electronics

MDPI

Article

Spectral Clustering Approach with K-Nearest Neighbor and Weighted Mahalanobis Distance for Data Mining

Lifeng Yin [1], Lei Lv [1], Dingyi Wang [2], Yingwei Qu [1], Huayue Chen [3,*] and Wu Deng [4,5]

[1] School of Software, Dalian Jiaotong University, Dalian 116028, China; sine981027@126.com (Y.Q.)
[2] School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China
[3] School of Computer Science, China West Normal University, Nanchong 637002, China
[4] College of Electronic Information and Automation, Civil Aviation University of China, Tianjin 300300, China
[5] State Key Laboratory of Traction Power, Southwest Jiaotong University, Chengdu 610031, China
* Correspondence: sunnyxiaoyue20@cwnu.edu.cn

Abstract: This paper proposes a spectral clustering method using k-means and weighted Mahalanobis distance (Referred to as MDLSC) to enhance the degree of correlation between data points and improve the clustering accuracy of Laplacian matrix eigenvectors. First, we used the correlation coefficient as the weight of the Mahalanobis distance to calculate the weighted Mahalanobis distance between any two data points and constructed the weighted Mahalanobis distance matrix of the data set; then, based on the weighted Mahalanobis distance matrix, we used the K-nearest neighborhood (KNN) algorithm construct similarity matrix. Secondly, the regularized Laplacian matrix was calculated according to the similarity matrix, normalized and decomposed, and the feature space for clustering was obtained. This method fully considered the degree of linear correlation between data and special spatial structure and achieved accurate clustering. Finally, various spectral clustering algorithms were used to conduct multi-angle comparative experiments on artificial and UCI data sets. The experimental results show that MDLSC has certain advantages in each clustering index and the clustering quality is better. The distribution results of the eigenvectors also show that the similarity matrix calculated by MDLSC is more reasonable, and the calculation of the eigenvectors of the Laplacian matrix maximizes the retention of the distribution characteristics of the original data, thereby improving the accuracy of the clustering algorithm.

Keywords: data mining; spectral clustering; Mahalanobis distance; Laplace matrix; K-means clustering

Citation: Yin, L.; Lv, L.; Wang, D.; Qu, Y.; Chen, H.; Deng, W. Spectral Clustering Approach with K-Nearest Neighbor and Weighted Mahalanobis Distance for Data Mining. *Electronics* **2023**, *12*, 3284. https://doi.org/10.3390/electronics12153284

Received: 8 June 2023
Revised: 18 July 2023
Accepted: 27 July 2023
Published: 31 July 2023

1. Introduction

Cluster analysis is a common analysis method; it is mainly used as a statistical analysis method that divides a set of objects into multiple class labels according to certain rules [1]. When performing cluster analysis on a data set, it is properly classified into a class according to similar data. The classification process is unsupervised, that is, data grouping is performed without prior knowledge. The K-means algorithm is a classic algorithm in traditional clustering. Although the K-means algorithm is easy to understand, it still has limitations, such as poor tolerance to sample shapes and ease of falling into local optimal solutions [2]. In order to achieve a high-quality clustering effect on data samples of arbitrary shapes, a spectral clustering algorithm [3] (referred to as SC) is proposed. Spectral clustering is used to convert the clustering problem into a graph division problem, use the data space to form an adjacency matrix, and then convert it into a similarity matrix to perform cluster analysis on the data by calculating the properties of the eigenvectors of the Laplacian matrix. The basic idea of the spectral clustering algorithm is to regard each object in the data as a node; the nodes are connected by edges and a weight (similarity) is assigned to each edge, transforming the clustering problem into a graph cutting problem. Better research on the cutting method of graphs [4,5] makes the similarity between data more accurate so that the similarity between different types of data is lower and the final

clustering effect is better. The spectral clustering algorithm can deal with the data space of any shape and has the advantages of low sensitivity, convergence to the global optimal solution, and support for high-dimensional data.

The spectral clustering algorithm has been paid more and more attention by researchers, and the application range of spectral clustering analysis is very wide [6–9], including applications in information science, economics, data mining, heterogeneous data analysis, image segmentation, and machine vision in learning; it is also used in many fields such as business analysis, marketing analysis, biology, geography, and psychological analysis [10,11]. The spectral clustering algorithm is a clustering algorithm based on spectral graph theory, which is mainly divided into two categories: iterative spectral clustering and multipath spectral clustering, represented by the SM [12] and NJW [13] algorithms, respectively. In 2000, Shi et al. used the K-nearest neighbor strategy to construct a sparse similarity matrix for image segmentation tasks [14]. In the multipath spectral clustering algorithm, in order to solve the problem that the NJW algorithm must manually set K clusters, Kong et al. used the intrinsic gap feature to realize a spectral clustering algorithm that automatically determined the K value; for the problem of pairwise constraint propagation [12], Zhao et al. proposed a semi-supervised spectral clustering algorithm combining sparse representation and constraint propagation [15]. On the basis of common constraint spectral clustering, the algorithm uses the data in the constraint set as landmark points to construct a sparse representation matrix, obtains an approximate similarity matrix, and uses constraint transfer to further improve the clustering accuracy. By imposing symmetric low-rank constraints and structured, sparse, low-rank constraints, Jia et al. allowed two constraints to be jointly optimized to achieve mutual refinement on the basis of the novel tensor low-rank norm; the efficiency was iteratively improved using an enhanced Lagrange multiplier-based method [16].

In order to achieve a better clustering effect, in 2007, Wang et al. proposed density-sensitive semi-supervised spectral clustering (DSSC) [17]. In order to avoid the influence of low-quality constraints on supervision information, Wang et al. started by improving the quality of supervision information selection [18]. Subsequently, Wang et al. adopted an active query strategy to replace the random query strategy and adopted an active query function to dynamically select constraints to enhance the robustness of the algorithm [19]. Regarding the sensitivity of spectral clustering parameters, the Hessian matrix is used instead of the Laplace matrix. The measure-based semi-supervised algorithm learns the supervised information, changes the distance measure function in the clustering algorithm, obtains a new measure suitable for data clustering [20,21], and, finally, calculates W by the new measure for clustering.

Therefore, some researchers have put forward new ideas on the measurement of data point similarity and the construction of low-rank matrices. Tao et al. started to improve the algorithm based on the characteristics of data density [22]. They defined the density-sensitive distance of the density-sensitive item based on the distance measure and improved the clustering performance of traditional spectral clustering based on Euclidean distance. Ge et al. used a density-adaptive neighborhood-building algorithm to build neighborhood information without parameters and used the shared nearest neighbor as a measure of similarity between samples [23]. In this way, the influence of the parameters when constructing the similarity map was eliminated and the local density information was reflected. Du et al. [24] created a similarity matrix by dividing subsets to calculate local covariance to improve clustering performance. Yang et al. [25] performed spectral clustering on multi-class samples. According to the density of each cluster and the average distance to the few-class samples, the number of samples was calculated for each cluster and multi-class samples were selected to improve the clustering accuracy. Liu et al. [26] proposed a spectral clustering algorithm based on low-rank representation (LRR). Elhamifa et al. [27] proposed sparse subspace clustering (SSC), which enforces the sparseness of self-representing coefficients. On the basis of previous ones, Lu et al. [28] proposed the least square regression (LSR) method to construct the correlation matrix.

The least squares regression clustering algorithm obtains the coefficient matrix through norm constraints and then obtains the similarity matrix. But, it does not consider the influence of noise features on subspace segmentation. However, Jeong et al. [29] proposed a spectral clustering algorithm called KNN-SC, which can discover precise clusters by reducing the influence of noisy points. Zhang and others [30] introduced common neighbors into the similarity measure and proposed a spectral clustering algorithm SC-CNN based on common neighbors. Nataliani et al. [31] used the eigenvectors corresponding to the eigenvalues of the similarity matrix in a data set to divide the data set into different data sets, and a Gaussian kernel function suitable for the spectral clustering algorithm was proposed to classify points. Kamal et al. [32] used a topological and attribute random walk affinity matrix (TARWAM) as a new affinity matrix to calculate the similarity between nodes. Jiang et al. [33] learned each similarity matrix and consensus graph in a mutually reinforcing manner. When the consistency graph converges, the spectral clustering algorithm is performed on it to obtain the final clustering effect. Yin et al. [34] determined the center point in the fuzzy pre-clustering stage, used the product of the neighborhood radius eps and the dispersion degree fog as a benchmark to divide the data, used the Euclidean distance to determine the similarity between two data points, and used the degree of membership to record the information of common points in each cluster. In addition, some new methods have also been proposed in recent years [35–44].

In view of the time complexity of spectral clustering algorithms and their poor adaptability to large data samples, Zhu et al. [45] deployed a spectral clustering algorithm on platforms such as Spark and used the distributed parallel characteristics of the platform to reduce its time consumption. By optimizing the spectral clustering cut model, the algorithm time complexity could be reduced [46]. In spectral clustering algorithms, many researchers use Euclidean distance or a Gaussian kernel function to describe the similarity by adjusting parameters or data weighting and seldom use other methods to calculate the similarity matrix. There are also many researchers incorporating other algorithms to reduce errors. Chen et al. [47] and others proposed a joint learning method of K-means and spectral clustering based on the multiplicative update rule. In addition, some new methods have also been proposed in recent years [38,44–55].

In a spectral clustering algorithm, the calculation of distance has a great influence on the effect of clustering. If the dimensions of the data are different, the distance calculated by Euclidean distance or the Gaussian function will greatly affect the clustering effect. In order to overcome the influence of the data dimension on the clustering effect, ref. [56] used Mahalanobis distance to solve the problem. Refs. [57,58] added weight to the Mahalanobis distance to study a spectral clustering algorithm and [59–61] improved it by changing the distance calculation. However, the calculation of the distance in these studies did not consider the influence of the strength of the linear correlation between the data on the spectral clustering effect, although the Mahalanobis distance could eliminate the influence of the dimension when calculating the distance between the data and capture the linearity between the data. But, in spectral clustering algorithms, capturing the degree of data correlation is more important than capturing data correlation. In order to solve this problem, this paper uses weighted Mahalanobis distance to calculate the distance between data points; the weight uses a correlation coefficient matrix to participate in the operation, and a weighted Mahalanobis distance spectral clustering algorithm (denoted as MDLSC) is proposed. The main contributions are as follows:

(1) In order to better capture the degree of linear correlation between the data, a correlation coefficient matrix is used as the weight to calculate the Mahalanobis distance, and then, the KNN method is used to construct the similarity matrix W.
(2) The degree matrix D is calculated according to W and then a regularized Laplacian matrix L is constructed.
(3) L is normalized and decomposed to obtain the feature space for clustering. This method fully considers the degree of linear correlation between data and achieves accurate clustering.

(4) Multi-angle comparative experiments are conducted on artificial and UCI data sets to verify the superiority of MDLSC.

2. Related Information

2.1. Mahalanobis Distance

Mahalanobis distance [62] was proposed by P. C. Mahalanobis and represents the covariance distance of data; it is used to evaluate the distance between data like Euclidean distance, Manhattan distance, Hamming distance, etc. However, it can also deal with the problem of non-independent and identical distribution between dimensions in high-dimensional, linearly distributed data; it is also a correction of Euclidean distance, which corrects the problem whereby the scales of each dimension in Euclidean distance are inconsistent and related.

The Mahalanobis distance formula for a single multidimensional variable X is as follows:

$$D_M(X) = \sqrt{(X-\mu)^T \Lambda^{-1} (X-\mu)} \tag{1}$$

where μ is the mean of variable X; Λ is the covariance matrix of variable X.

The Mahalanobis distance of two different sample points X and Y is shown in Formula (2):

$$D_M(X,Y) = \sqrt{(X-Y)^T \Gamma^{-1} (X-Y)} \tag{2}$$

Γ is the covariance matrix of multi-dimensional variables X and Y, which is used to describe the similarity of X and Y, as shown in Formula (3):

$$\Gamma = \begin{bmatrix} Cov(X,X) & Cov(X,Y) \\ Cov(Y,X) & Cov(Y,Y) \end{bmatrix} \tag{3}$$

where Cov () is the covariance function. It is not affected by dimension, and the Mahalanobis distance between two points is not related to the measurement unit of the original data.

2.2. Similarity Matrix and Laplace Matrix

An undirected graph G can be described by a set of points V and set of edges E, namely, G (V, E), where V is the set of all points in the data set $\{v_1, v_2, \ldots, v_n\}$ and E is a collection of edges. W_{ij} is defined as the weight of the edge between points v_i and v_j. For v_i and v_j with an edge connection, $W_{ij} > 0$; for v_i and v_j without an edge connection, $W_{ij} = 0$ and $W_{ij} = W_{ji}$.

In a spectral clustering algorithm, the weight cannot be given directly. When the weight value is given quantitatively, the adjacency matrix is obtained by the similarity S_{ij}, which is measured by the distance of each data point, and then the similarity matrix is calculated by the KNN algorithm. The main idea is as follows: use the KNN algorithm to traverse all sample points, take the nearest k points of each sample as the nearest neighbors, $S_{ij} > 0$ only between the k points closest to the sample, and δ is the neighborhood width (scale parameter).

However, this method will cause the reconstructed adjacency matrix to be asymmetric, and the algorithm requires a symmetric adjacency matrix. In order to solve this problem, one of the following two methods is generally adopted:

The first KNN method (denoted as method_A) is to keep S_{ij} as long as a point is in the k-nearest neighbor of another point. The second KNN method (denoted as method_B) is that the two points must be k-nearest neighbors to each other to keep S_{ij}. Both of the above methods can use Equation (4) to calculate s_{ij}, i.e. W_{ij}:

$$W_{ij} = W_{ji} = \begin{cases} 0 \\ \exp(-\frac{d^2(v_i,v_j)}{2\delta^2}) \end{cases} \tag{4}$$

where v is the data sample point, d (v_i, v_j) is the distance between the two sample points, generally taking the Euclidean distance, and δ is the scale parameter.

The accuracy of the spectral clustering algorithm is affected by the quality of the similarity matrix. Generally speaking, the higher the quality of the similarity matrix, the better the clustering result. At present, spectral clustering algorithms are mostly based on KNN and use Euclidean distance to obtain a similarity matrix. The degree matrix D is a diagonal matrix and the diagonal element D_{ii} is the sum of the elements of each row of the similarity matrix W, as shown in Equation (5):

$$D_{ii} = \sum_{j}^{n} W_{ij} \tag{5}$$

The Spectral clustering algorithms usually use the largest k eigenvectors of the similarity matrix for clustering, but the similarity matrix cannot guarantee that the eigenvectors corresponding to the selected k eigenvalues are heterogeneous vectors. Therefore, there is a problem of selecting a piece of eigenvectors multiple times when selecting eigenvectors through W, which leads to poor representativeness of the selected eigenvectors.The Laplacian matrix L (Laplacian matrix) is a positive semi-definite matrix, the minimum eigenvalue of L is 0 and the corresponding eigenvector is 1. When selecting the eigenvectors corresponding to the first k eigenvalues of L, the matrix can ensure that each component contains only one eigenvector; thus, the L matrix is introduced into the spectral clustering. The L matrix is generally divided into a canonical Laplacian matrix and a non-canonical Laplacian matrix. The non-canonical Laplacian matrix is the result of the subtraction of the degree matrix D and the similarity matrix W, as shown in Equation (6):

$$L(i,j) = D(i,j) - W(i,j) \tag{6}$$

The regularized Laplacian matrix is further divided into the random walk Laplacian matrix and the symmetric Laplacian matrix, as shown in Equations (7) and (8), respectively:

$$L = D^{-1}L \tag{7}$$

$$L = D^{-\frac{1}{2}}LD^{-\frac{1}{2}} \tag{8}$$

2.3. Spectral Clustering Algorithm

The spectral clustering algorithm is a graph-based clustering algorithm. The idea of this algorithm originates from the theory of spectral graph partitioning [63]. An undirected weighted graph is generated through the similarity between data, and data points can be regarded as graphs. The similarity between vertices and data points is the weight of the edge between the two points, which transforms the clustering problem into a weighted undirected graph division problem. The spectral division of the undirected weighted graph divides the graph into several subgraphs, which corresponds to the clustering process of the clustering algorithm. For spectral graph division, the selection of graph division criteria will directly affect the division results. Commonly used graph division criteria include normative cut set, minimum cut set, average cut set, proportional cut set, and other criteria [64]. Compared with the graph partitioning problem, the spectral clustering algorithm considers the continuous relaxed form of the problem and transforms the graph partitioning problem into a spectral decomposition problem of finding the similarity matrix [65]. The basic idea of this method is to solve the Laplacian matrix on the basis of the similarity matrix of the data set, perform eigendecomposition, and complete the spectral clustering algorithm in the obtained eigenvector space. Its main steps are as follows:

(1) Construct a matrix W that can describe the similarity relationship between data through data samples. Construct Laplacian matrix L from degree matrix D and similarity matrix W.
(2) Calculate and sort the eigenvalues and eigenvectors of Laplacian matrix L.
(3) Take the eigenvectors corresponding to the first k eigenvalues after sorting, and arrange each vector in the column direction to form a new solution space.

(4) On the new solution space, use the classical clustering algorithm (fuzzy clustering, K-means, etc.) for clustering and map the clustering result back to the original solution space.

3. Improved Spectral Clustering Algorithm

3.1. Mahalanobis Distance Spectral Clustering (M_SC) Algorithm

The Mahalanobis distance between data points is calculated by Equation (2) and the KNN algorithm is used to take the nearest k points of each data sample as neighbors to form similarity matrix W. According to the similarity matrix, Laplacian matrix L based on the Mahalanobis distance is obtained by calculation, the eigenvalues and eigenvectors are solved for matrix L, and the eigenvectors corresponding to the first k eigenvalues are taken to perform K-means clustering. The specific steps of the Mahalanobis distance algorithm are as follows:

(1) Calculate the Mahalanobis distance between each point and all other points, use the KNN algorithm to initialize it, and form similarity matrix W.
(2) Calculate the regular Laplace matrix $L = D^{-1/2}(D - W)\, D^{-1/2}$; D is the degree matrix, that is, the sum of the elements of each row of similarity matrix W.
(3) Calculate the eigenvectors corresponding to the first k eigenvalues of matrix L and place them in columns to form matrix U.
(4) Unitize the row vectors in matrix U sequentially and then perform K-means clustering to obtain K clustering results.

3.2. Weighted Mahalanobis Distance Spectral Clustering (MDLSC) Algorithm

Mahalanobis distance takes into account the correlation between features and measures the linear relationship between features by using a covariance matrix. This allows for more accurate measurement of the similarity between data points when computing the similarity matrix. During the calculation of the Mahalanobis distance, the calculation of the mean value and covariance matrix of the data will standardize the data so that the distance calculation is not affected by the data scale. This makes the Mahalanobis distance somewhat robust to scale changes in the data.

However, the Mahalanobis distance also has certain shortcomings. It does not consider the importance of different features for calculating the distance; thus, this paper proposes a weight-based Mahalanobis distance spectral clustering (denoted as *MDLSC*) algorithm.

3.2.1. Weighted Mahalanobis Distance

This section elaborates on three perspectives: the weight derivation process of the weighted Mahalanobis distance (denoted as WMD), the significance of selecting the correlation coefficient as the Mahalanobis distance weight for spectral clustering, and the steps for using the weighted Mahalanobis distance to construct the similarity matrix of the data set.

The Weight Derivation Process of WMD

For the given data set D, assuming there are n sample points and p features in D and that the data set D is a matrix of $n \times p$, let X and Y be two sample points in the data set D, respectively, $X = (x_1, x_2, \ldots\ldots, x_p)^T$ and $Y = (y_1, y_2, \ldots\ldots, y_p)^T$, where p is the number of features of data samples X and Y, thus introducing the weighted Mahalanobis distance algorithm. The weighted Mahalanobis distance of samples X and Y is as shown in Equation (9):

$$D_{WM}(X, Y) = \sqrt{(X - Y)^T B \Gamma^{-1} B (X - Y)} \tag{9}$$

Among them, B is the weight matrix, which is determined by principal component analysis, and Γ is the covariance matrix. The specific derivation process of the weight matrix B is as follows:

(1) When performing principal component analysis, it is first necessary to standardize the data set D. Calculate the covariance matrix Γ of the standardized data D'; Γ is also

the correlation coefficient matrix R, that is, Γ = R, and further obtain the characteristic root of Γ: λ1 ≥ λ2 ≥… ≥ λp.

(2) When performing principal component analysis on data, it is first necessary to standardize the data set D. Calculate the covariance matrix Γ of the standardized data D'; Γ is also the correlation coefficient matrix R, that is, Γ = R, and further obtain the the characteristic root of Γ: $\lambda_1 \geq \lambda_2 \geq \ldots\ldots \geq \lambda_p$.

(3) Let Γ correspond to the unitized eigenvectors $U_1, U_2, \ldots\ldots, U_p$ and the matrix $U = (U_1, U_2, \ldots\ldots, U_p)$, formed by the eigenvectors. Suppose the diagonal matrix formed by characteristic roots $\{\lambda_1, \lambda_2, \ldots\ldots\lambda_p\}$ is Λ = diag $(\lambda_1, \lambda_2, \ldots\ldots\lambda_p)$; then, $\Lambda = U^{\mathrm{T}}\,\Gamma U$ is established. Data set D' is transformed into $F = U^{\mathrm{T}}\,D'$. Sample point $X = (x_1, x_2, \ldots\ldots, x_p)^T$ is converted into $F_X = U^{\mathrm{T}}\,X$, and sample point $Y = (y_1, y_2, \ldots\ldots, y_p)^T$ is converted into $F_Y = U^{\mathrm{T}}\,Y$.

(4) Record $F = (F_1, F_2, \ldots, F_p)^T$, $F_i = U_i{}^T D'$, $i = 1, 2, \ldots, p$ and define $\eta_i = \lambda_i/\mathrm{tr}(\Lambda)$ as the variance contribution rate of F_i (i = 1,2,…,p) because F_i and F_j are independent of each other (i and j are integers between 1 and p); then, the F_i variance contribution rate can be used as the weight. The weight matrix recorded as the data set is W = diag(η_1, η_2,……, η_p); then, the correlation coefficient matrix of D' after transformation is $\Gamma = U^T \Gamma U$. For any two sample points, the weight matrix is the same as the weight matrix of the data set and then the weighted Mahalanobis distance from F_X to F_Y is expressed as shown in Equation (10):

$$\begin{aligned} D_{WM}(F_X, F_Y) &= \sqrt{(F_X - F_Y)W\overline{\Gamma}^{-1}W(F_X - F_Y)} \\ &= \sqrt{(U^T X - U^T Y)^T W\overline{\Gamma}^{-1}W(U^T X - U^T Y)} \\ &= \sqrt{(U^T (X - Y))^T W\overline{\Gamma}^{-1}WU^T (X - Y)} \\ &= \sqrt{(X - Y)^T UW\overline{\Gamma}^{-1}WU^T (X - Y)} \\ &= \sqrt{(X - Y)^T UW(U^T \Gamma U)^{-1}WU^T (X - Y)} \\ &= \sqrt{(X - Y)^T UWU^T \Gamma^{-1} UWU^T (X - Y)} \end{aligned} \tag{10}$$

(5) According to Equation (12), $UWU^T = B$ in Equation (13). Since W = diag(η_1, η_2, …, η_p), Λ = diag $(\lambda_1, \lambda_2, \ldots\ldots\lambda_p)$, $\eta_i = \lambda_i/\mathrm{tr}(\Lambda)$; thus, $W = \Lambda/\mathrm{tr}(\Lambda)$. Deduce $B = U(\Lambda/\mathrm{tr}(\Lambda))U^T$ and, because $\Lambda = U^{\mathrm{T}}\Gamma U$, $\Gamma = U\Lambda U^T$; then, $B = \Gamma/\mathrm{tr}(\Lambda) = R/\mathrm{tr}(\Lambda)$. The weighted Mahalanobis distance formula for two samples X and Y is shown in Equation (11):

$$D_{WM}(X, Y) = \sqrt{\frac{1}{tr(\Lambda)^2}(X - Y)^T R\Gamma^{-1}R(X - Y)} \tag{11}$$

(6) Since Γ = R, $R\Gamma^{-1} = RR^{-1} = I$; I is the identity matrix, and the final weighted Mahalanobis distance formula of samples X and Y is shown in Equation (12).

$$D_{WM}(X, Y) = \sqrt{(X - Y)^T \frac{1}{tr(\Lambda)^2}R(X - Y)} \tag{12}$$

The weight of the weighted Mahalanobis distance is $R/(\mathrm{tr}(\Lambda))^2$. Since $(\mathrm{tr}(\Lambda))^2$ is a constant, the calculation of this constant can be ignored for zooming in and zooming out at the same multiple and will not affect the final effect. The weight can be regarded as determined by the correlation coefficient R. The final weighted Mahalanobis distance formula is shown in Equation (13):

$$D_{WM}(X, Y) = \sqrt{(X - Y)^T R(X - Y)} \tag{13}$$

The Superiority of WMD

The weight of the Mahalanobis distance in Equation (2) uses covariance and the weight of the weighted Mahalanobis distance in Equation (14) uses the correlation coefficient

matrix. These two formulas are to highlight the importance of features, but there are differences between the two, as follows:

(1) Covariance matrix: The covariance matrix is used to measure the degree of correlation and variation between different features. In the covariance matrix, the variance of each feature corresponds to its own importance, while the covariance measures the correlation between two features. If a feature has a higher variance, it has more variation in the data; thus, it can be considered to contribute more to the overall distance. Therefore, when using the covariance matrix as weights, the importance of features with a larger variance will be more prominent.

(2) Correlation coefficient matrix: The correlation coefficient matrix measures the linear correlation between features, which removes the variance of the features themselves and focuses more on the relationship between features. The value range of the correlation coefficient is between -1 and 1, and the closer the absolute value is to 1, the stronger the linear relationship between the two features. If a feature has a strong linear relationship with other features, its corresponding value in the correlation coefficient matrix will be relatively large. When using the correlation coefficient matrix as a weight, the importance of features with strong linear relationships will be more prominent.

In summary, the covariance matrix highlights the importance of features with larger variances, while the correlation coefficient matrix highlights the importance of features with stronger linear relationships. The choice of which to use as weights depends on your specific definition and need to highlight the importance of features.

Mahalanobis distance with a correlation coefficient matrix as a weight has some advantages in spectral clustering:

(1) Considering the linear relationship between features: The correlation coefficient matrix captures the linear correlation between features. This is beneficial for spectral clustering since spectral clustering algorithms exploit the characteristic relationships of the data to build a similarity matrix. As a weight, the correlation coefficient matrix can better reflect the linear relationship between data points and provide a more accurate similarity measure.

(2) The variance influence of the feature itself is removed: the correlation coefficient matrix eliminates the variance influence of the feature itself so that more attention is paid to the relationship between features when calculating the distance, without being affected by the variance of a single feature. This can reduce the dominance of large feature variance on the distance calculation and thus better capture the relative importance between features.

Mahalanobis distance with a covariance matrix as a weight can also be used in some cases, but, usually, a correlation coefficient matrix is more suitable for the task of spectral clustering. Therefore, when performing spectral clustering, Mahalanobis distance using a correlation coefficient matrix as a weight may produce better clustering results.

Calculation Steps of WMD

Given data set D, and assuming that there are n samples and p variables (features) and $D = (d_1, d_2, \ldots, d_p)$, $d_i = (d_{i1}, d_{i2}, \ldots, d_{in})^T$ is an n-dimensional column vector, the steps to calculate the weighted Mahalanobis distance are as follows:

(1) Calculate the mean of each variable d_i. i is an integer between 1 and p, the specific formula is shown in Equation (14):

$$\overline{d}_i = \frac{1}{n}\sum_{j=1}^{n} d_{ji} \tag{14}$$

(2) Calculate the standard deviation of each variable d_i. i is an integer between 1 and p, the specific formula is shown in Equation (15):

$$std(d_i) = \frac{\sqrt{\sum_{j=1}^{n} (d_{ji} - \overline{d}_i)^2}}{n-1} \tag{15}$$

(3) Calculate the covariance matrix: Calculate the covariance among all variables in data set D. The covariance matrix is a $p \times p$ matrix where the (i, j)th element represents the covariance between the i-th variable and the j-th variable. Each element of the covariance matrix can be calculated using Equation (16):

$$Cov(d_i, d_j) = \frac{\sum_{k=1}^{n} (d_{ki} - \overline{d}_i)(d_{kj} - \overline{d}_j)}{n-1} \tag{16}$$

Among them, $Cov(d_i, d_j)$ represents the covariance between d_i and d_j, and $\overline{d}_i$ and $\overline{d}_j$ represent the mean values of d_i and d_j, respectively. Σ represents the summation symbol and n represents the number of samples.

(4) Compute the correlation matrix: Calculate the correlation coefficient matrix using the covariance matrix. The correlation coefficient matrix is a $p \times p$ matrix where the (i, j)th element represents the correlation coefficient between the i variable and the j variable. Each element of the correlation coefficient matrix can be calculated using Equation (17):

$$Corr(d_i, d_j) = \frac{Cov(d_i, d_j)}{std(d_i) \times std(d_j)} \tag{17}$$

$Corr(d_i, d_j)$ represents the correlation coefficient between d_i and d_j, $Cov(d_i, d_j)$ represents the covariance between d_i and d_j, $std(d_i)$, and $std(d_j)$ represents the standard deviation of d_i and d_j.

(5) The correlation coefficient matrix obtained through step (4) is R, and the weighted Mahalanobis distance between any two sample points can be calculated by Equation (14).

3.2.2. Construction of the Similarity Matrix

The data set D is the same as above. D is transformed into an undirected graph $G = (V, E)$, $V = \{v_1, v_2, \ldots, v_n\}$, V represents the set of nodes of graph G, and E represents the set of edges of graph G. Each node v_i (i is an integer from 1 to n) has a one-to-one correspondence with data objects and each edge represents the relationship between data objects. The steps to construct the similarity matrix are as follows:

(1) Calculate the weighted Mahalanobis distance between any two sample points, and the data constitute an $n \times n$ matrix (denoted as WM_matrix), which is a symmetric matrix.

(2) Use the K-NN algorithm to traverse all data points and take the nearest k points of the distance (this distance is calculated by the Mahalanobis distance calculated in step (1)) of each data node v_i as the nearest neighbor points, denoted as v_i_KNN. v_i_KNN is a set with k data nodes. Only the k points closest to the data node v_i have a similarity greater than 0. Let W_{ij} represent the similarity between the i-th data node v_i and the j-th data node v_j. When v_j belongs to v_i_KNN, then $W_{ij} > 0$, and W_{ij} is the weighted Mahalanobis distance from v_i to v_j; otherwise. $W_{ij} = 0$. Experiments indicate that the effect of k being 3 and 5 is the same and the best, respectively. In order to reduce the amount of calculation, k is 3 in this paper. In order to ensure that the constructed similarity matrix W is symmetrical, it is constructed using method_B of KNN (see Section 2.2).

3.2.3. Construction of the Regularized Laplacian Matrix

The steps to construct a regularized Laplacian matrix are as follows:

(1) Construct the degree matrix D. According to similarity matrix W (calculated in Section 3.2.2), degree matrix D is constructed. Degree matrix D is a diagonal matrix and its diagonal element $D(i, i)$ represents the degree of node v_i (that is, the number of edges connected to node v_i). Graph G constructed in this paper is an undirected graph. $D(i, i)$ can be calculated according to Equation (6).
(2) Construct a standard Laplacian matrix L. According to the obtained similarity matrix W and degree matrix D, use Equation (7) to construct a standard Laplacian matrix L, i.e., $L = D - W$. The Laplacian matrix can provide important information about the graph structure and help cluster the data. The eigenvalues and eigenvectors of the Laplacian matrix can be obtained through spectral decomposition, thus realizing the core steps of the spectral clustering algorithm.
(3) Construct a regularized Laplacian matrix. Symmetrically normalize L obtained by Equation (2), using Equation (18) to obtain a regularized Laplacian matrix L.

$$L = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}} \tag{18}$$

In Equation (19), D is a symmetric degree matrix and $D^{-\frac{1}{2}}$ means computing the inverse square root of the degree matrix D. For each element D_{ii} in degree matrix D, take the square root of its reciprocal, shown in Formula (19).

$$D_{ii}{}^{-\frac{1}{2}} = \frac{1}{\sqrt{D_{ii}}} \tag{19}$$

By multiplying by the inverse square root of degree matrix D, the weights of similarity matrix W can be normalized such that nodes with larger degrees and nodes with smaller degrees have similar importance in the Laplacian matrix. This helps remove imbalances, allowing spectral clustering to better handle the differently sized clusters present in the data.

3.2.4. Steps for the MDLSC Algorithm

The previous sections introduced the calculation process for the weighted Mahalanobis distance and the construction process for the similarity and symmetric normalized Laplacian matrices. This section gives the steps for the MDLSC algorithm based on these contents. The details are as follows:

(1) According to the steps in Section 3.2.1, calculate the weighted Mahalanobis distance of any two sample points and construct the weighted Mahalanobis distance matrix of the entire data set D.
(2) According to the steps in Section 3.2.2, construct the similarity matrix W.
(3) According to the steps in Section 3.2.3, construct the symmetric normalized Laplacian matrix L.
(4) Perform spectral decomposition on L to obtain the eigenvectors corresponding to the first k smallest eigenvalues and place them according to column vectors to form matrix P. In order to make each new dimension reveal a cluster, the dimension k of the new space formed by P is usually set to the number of the last clustered clusters.
(5) Unitize the row vector of matrix P and then perform K-means clustering to obtain the clustering result.
(6) Calculate the clustering accuracy rate ACC, standard mutual confidence rate NMI, and purity and F1-Score.

3.2.5. Pseudocode for the MDLSC Algorithm

The pseudocode of the MDLSC Algorithm 1 is as follows:

Algorithm 1: MDLSC

Input: Data D, Number of clusters K, Number of neighbors k
Output: Clustering results K

```
X = read(D)//read data into X
for x_i to x_n
d_i = get-mahalanobis( )//Calculate the weighted Mahalanobis distance of each//data point to other data points and store to di-matrix
KNN (di-matrix) = W_k //Use the KNN algorithm to mark the nearest k data points in the di-matrix to get W_k, here k = 3.
W = (W_k +W_T)/2, W_k is transposed with W_T //Calculate the Similarity matrix
D = diagonal sum of W //Calculate the Degree matrix
L = D-W //Calculate a standard Laplacian matrix L
L = D^{-1/2} L D^{-1/2} //Calculate the symmetric normalized Laplacian matrix
Calculate the first k smallest eigenvectors p_1, p_2, ..., p_k of L//the value of k is the same as the number of clusters
form p_1, p_2, ..., p_k into matrix P, P ∈ R^{n×k}
normalize P so that the norm of each row of p is 1.//data normalization
for (i = 1; i < = n; i++)
Taking each row vector y_i in P as a data point, K-mean(y_i)
Update data point labels
return K clustering results
```

3.2.6. Algorithm Time Complexity Analysis

The calculation amount of a traditional spectral clustering algorithm mainly includes three aspects: the generation of the similarity matrix, the calculation of the Laplace matrix, and the cluster analysis of the eigenvectors. For the MDLSC algorithm, optimized in terms of similarity matrix and Laplace matrix regularization, for the weighted Mahalanobis distance similarity matrix, due to the improved calculation method, the time complexity is increased from $O(n^2)$ of Euclidean distance to $O(n^3)$ (n is the number of data points in the data set), increasing the time consumption but considering the structural similarity between data. For the calculation of the Laplacian matrix, this paper adopts a symmetric Laplacian matrix. The regularization of a Laplacian matrix is usually completed by matrix multiplication, and its complexity is high and reaches $O(n^3)$. For the formed feature matrix, the K-means algorithm is usually used for cluster analysis; its time complexity is $O(Kmnt)$, t represents the number of iterations, m represents the number of data, n represents the dimension of data features, and K represents the number of clusters. Therefore, the time complexity of the MDLSC algorithm is $O(n^3) + O(\mathrm{n}^3) + O(Knmt)$. The time complexity of the MDLSC algorithm is higher than that of the classical spectral clustering algorithm due to changes in the construction of the similarity matrix and the calculation of the regularized Laplacian matrix, but the accuracy of the clustering is effectively improved.

4. Experimental Results and Analysis

4.1. Data Set Description

In order to verify the effectiveness of the proposed MDLSC algorithm, The experimental environment uses Intel(R) Core(TM) i5-10300H cpu with a frequency of 2.50 GHz, 16G memory, 64-bit operating system and the PyCarm development environment based on Python 3.8. The experiment adopted the classical clustering artificial and UCI data sets. The artificial data sets included aggregation, flame, Jain, path-based, and spiral data sets. The UCI data sets included glass, wine, iris, and vehicle data sets. A brief introduction to the artificial experimental data set is listed in Table 1, and a brief introduction to the UCI data set is listed in Table 2.

Table 1. Artificial data sets.

Data Sets	Sample	Feature	Number of Clusters
Aggregation	789	2	7
Flame	241	2	2
Jain	374	2	2
Path-based	301	2	3
Spiral	313	2	3

Table 2. UCI data sets.

Data Sets	Sample	Feature	Number of Clusters
Glass	215	9	6
Wine	178	13	3
Iris	150	4	3
Vehicle	752	18	4

4.2. Evaluation Metrics

In this paper, the clustering accuracy rate ACC [66], standard mutual trust rate NMI [67], purity [68], Rand [69] index, and F1-Score [70] were used to evaluate the quality of the clustering algorithm. The calculation of ACC is shown in Formula (20):

$$\mathrm{ACC}(p,r) = \frac{\sum_{i=1}^{n} \delta(r_i, map(p_i))}{n} \quad \delta(a,b) = \begin{cases} 1\,, a = b \\ 0\,, other \end{cases} \tag{20}$$

$map(p_i)$ is the best mapping function, which can map the sample clustering label p_i to its equivalent sample true label, and there is a one-to-one mapping relationship between the label obtained by clustering and the sample true label.

NMI is the mutual trust ratio, which represents the normalized information of the data. The definition of NMI between two random variables R and S is shown in Formula (21).

$$\mathrm{NMI}(R,S) = \frac{I(R,S)}{\sqrt{(H(R)H(S))}} \tag{21}$$

where I (R, S) is the mutual information of labels R (true label) and S (predicted label), and H () represents information entropy.

Purity represents the purity of the data and its expression is shown in Equation (22):

$$\mathrm{Purity}(\mathrm{Y},\mathrm{C}) = \frac{1}{n}\sum_{i} \max_{j} |y^i \cap c^j| \tag{22}$$

n represents the total number of samples, and samples are divided into k clusters. Let $|c^i|$ and $|y^i|$ obtained by the clustering algorithm represent the number of elements in c^i and y^i, respectively. Here, the value range of purity (Y, C) is [0, 1], and the larger the value, the better the clustering effect.

The Rand index is a commonly used evaluation index for clustering results that is used to measure the degree of consistency between the clustering results and the external standard classes of the data. The accuracy is equal to the ratio of the number of correct matching pairs to the total matching pairs, that is, RI = correct matching pairs/total matching pairs, as shown in Equation (23):

$$\mathrm{RI} = \frac{r+s}{q+s+r+t} \tag{23}$$

Let C denote the actual category information and K denote the clustering result. r represents the logarithm of elements of the same category in C and K, s represents the logarithm of elements of different categories in C and K, and $r + s$ is the number of correctly

divided elements. q denotes the logarithm of elements in C that belong to the same class but not in K, and t denotes the logarithm of elements that do not belong to the same class in C but belong to the same class in K. $q + s + r + t$ represents the number of element logarithms that can be composed in the data set. The value range of the Rand (RI) evaluation index is [0, 1]. The larger the RI value, the greater the similarity of the data in the cluster and the higher the consistency of the two divisions.

Precision refers to the proportion of the samples that are indeed positive among all the samples that are judged to be positive and reflects the error rate of the prediction results. The calculation is shown in Equation (24):

$$P = \frac{TP}{TP + FP} \tag{24}$$

Recall refers to the proportion of positive samples among all the actual positive samples, which reflects the missed detection rate of the prediction results. The calculation formula is shown in Equation (25):

$$R = \frac{TP}{TP + FN} \tag{25}$$

F1 is the harmonic mean of precision and recall, and the formula is shown in Equation (26). Its value is between 0 and 1, and the closer it is to 1, the better the clustering effect:

$$F1 = \frac{2 \times P \times R}{P + R} \tag{26}$$

TP is the number of true positives and FP is the number of false positives; TN is the number of true negatives and FN is the number of false negatives. The larger the value of the evaluation indicators ACC, NMI, purity, and F1-Score, the closer the prediction result is to the real label, the better the clustering performance and the value range of the evaluation indicators is [0, 1].

4.3. Experimental Results and Analysis

In this paper, the classical clustering algorithm was selected to be compared with the proposed MDLSC algorithm and the Mahalanobis distance spectral clustering (M-SC) algorithm on artificial data sets. The classic clustering algorithms are the Gaussian kernel spectral clustering (kernel SC) algorithm, NJW clustering algorithm, K-means algorithm, and LSR algorithm. For the UCI data set, the NJW clustering algorithm, DSSC algorithm, literature [70] algorithm, and MDLSC algorithm were used for comparison.

4.3.1. Experiment on Artificial Data Sets

The five algorithms were tested on artificial data sets, and the evaluation index values obtained are shown in Table 3.

Table 3. Evaluation index value of five algorithms on artificial data sets.

Data Sets	Evaluation Metrics	Kernel SC	LSR	K-Means	NJW	MDLSC	M_SC
Aggregation	ACC	0.9035	0.8753	0.8238	0.888	0.928	0.9112
	NMI	0.9088	0.8621	0.8421	0.9392	0.9492	0.8594
	Purity	0.8327	0.8461	0.8052	0.8327	0.8329	0.8314
	F1-Score	0.9136	0.914	0.8238	0.9328	0.9651	0.9112
Flame	ACC	0.8708	0.9916	0.7541	0.9958	0.9916	0.9833
	NMI	0.9259	0.9354	0.4534	0.9633	0.9875	0.899
	Purity	0.875	1	0.85	0.9958	1	0.9916
	F1-Score	0.8837	0.9916	0.8676	0.9909	0.9916	0.9833

Table 3. *Cont.*

Data Sets	Evaluation Metrics	Kernel SC	LSR	K-Means	NJW	MDLSC	M_SC
Jain	ACC	0.5124	0.512	0.2769	0.4879	0.5425	0.512
	NMI	0.1108	0.1108	0.0468	0.1108	0.1492	0.1234
	Purity	0.1469	0.1637	0.0738	0.1469	0.1936	0.1892
	F1-Score	0.512	0.5123	0.4876	0.5123	0.6517	0.512
Path-based	ACC	0.8533	0.923	0.8333	0.88	0.9133	0.9066
	NMI	0.7148	0.7204	0.6622	0.7941	0.7951	0.8115
	Purity	0.6521	0.6254	0.4965	0.4552	0.6721	0.6366
	F1-Score	0.7392	0.7373	0.73	0.7373	0.7833	0.7833
Spiral	ACC	0.5946	0.5519	0.381	0.5743	0.6538	0.6634
	NMI	0.6358	0.6684	0.6176	0.6703	0.6995	0.7039
	Purity	0.802	0.8557	0.8218	0.8429	0.8461	0.8743
	F1-Score	0.5738	0.5734	0.5217	0.5707	0.6714	0.6743

In order to observe the advantages and disadvantages of the clustering quality of each algorithm more clearly, the histogram of each indicator was generated according to each indicator data in Table 3, as shown in Figures 1–4, respectively.

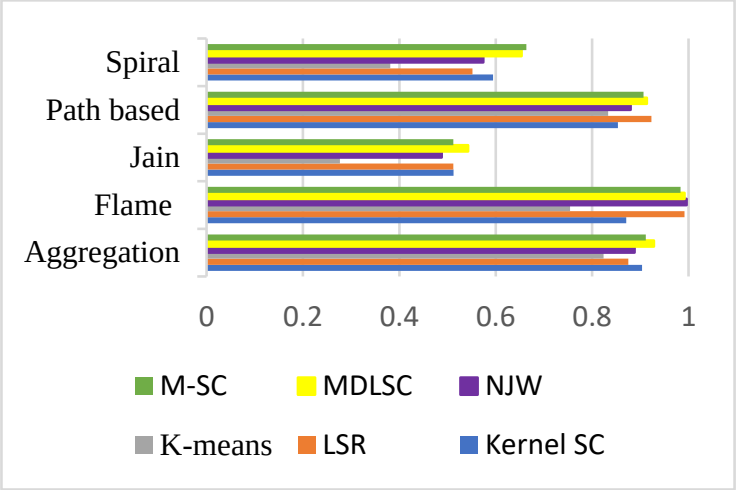

Figure 1. ACC Index Comparison Chart.

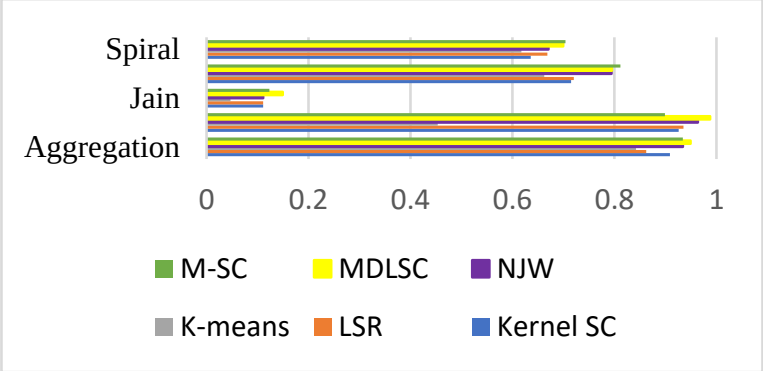

Figure 2. NMI Index Comparison Chart.

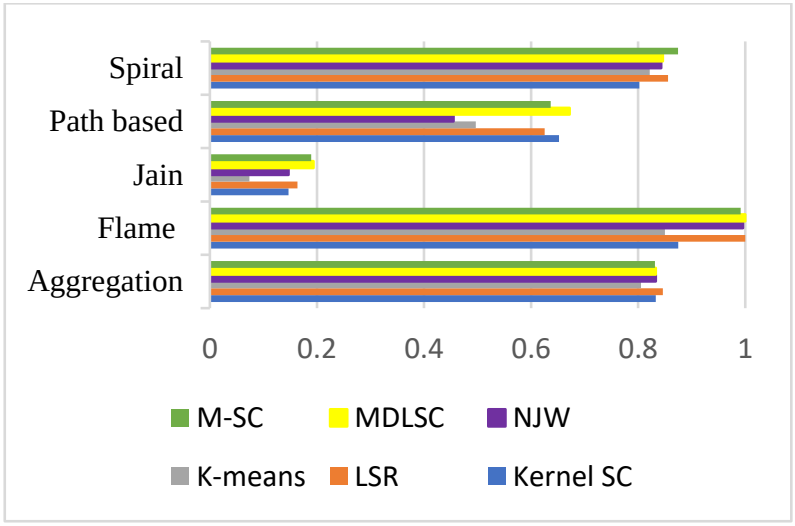

Figure 3. Purity Index Comparison Chart.

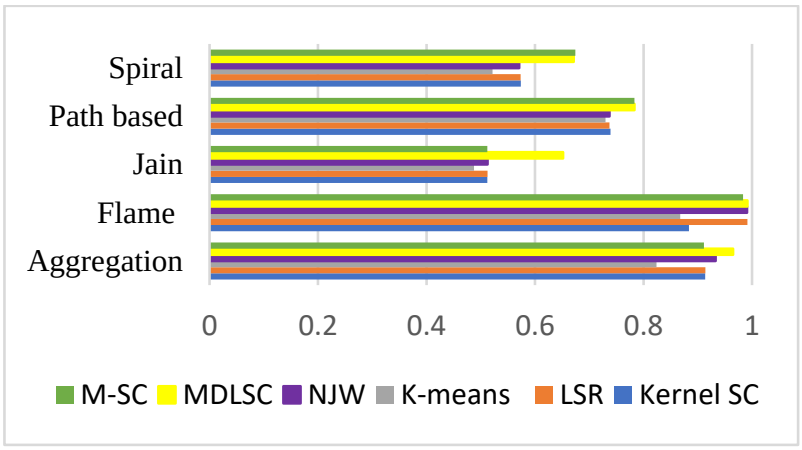

Figure 4. F1-Score Index Comparison Chart.

As can be seen from Table 3, the ACC of the two Mahalanobis distance-based spectral clustering algorithms proposed in this paper was better than the K-means algorithm and was also higher than some classical spectral clustering algorithms. It can be seen from Figure 1 that the MDLSC algorithm proposed in this paper was higher than other comparison algorithms on the ACC index of different artificial data sets. The M-SC was slightly lower than the MDLSC algorithm, which reflects the superiority of the weighted Mahalanobis distance algorithm.

Specifically, the ACC of the MDLSC algorithm on the Aggregation data set was 10% higher than that of the K-means algorithm; compared with the classical spectral clustering algorithm, it was improved by 2~5%. In terms of mutual trust rate, the NMI of the MDLSC algorithm was 10% higher than that of the K-means algorithm and 8% higher than that of the LSR algorithm. The purity of the MDLSC algorithm was 1% lower than that of the LSR algorithm, but the F-Score value of the MDLSC algorithm was higher than other comparison algorithms. The purity of the MDLSC algorithm was 1% lower than that of the LSR algorithm, but the F1-Score value of the MDLSC algorithm was higher than for other comparison algorithms. The purity was slightly lower than the LSR algorithm, considering that the calculation of the Mahalanobis distance was more complicated than the Euclidean distance calculation process, and the complex value was calculated after the eigenvector was normalized. But, on the whole, the MDLSC algorithm proposed was better than other algorithms.

For the flame and Jain data sets, the ACC of the MDLSC algorithm was 24~27% higher than that of the K-means algorithm; the ACC of the flame data set exceeded the kernel SC algorithm by 12%, which was 3~6% higher than for other algorithms. For the mutual trust rate NMI, the NMI of the MDLSC algorithm was improved by 10~54% compared with the K-means algorithm. The F1-Score value of the MDLSC algorithm was 14~17% higher than for other comparison algorithms.

From Table 3 (path-based data set), it can be concluded that the MDLSC algorithm was slightly lower than the LSR for the ACC value, but the NMI of the MDLSC algorithm for the mutual trust rate NMI was 7% higher than for the LSR algorithm and 13% higher than for the K-means algorithm. The F1-Score value was 4~5% higher than for other comparison algorithms. For the spiral data set, the ACC of the MDLSC algorithm was 27% higher than that of the K-means algorithm, 10% higher than that of the classical spectral clustering algorithm LSR, and 6~8% higher than that of other spectral clustering algorithms. For the F1-Score, the value of the MDLSC algorithm was higher than that of all comparison algorithms.

The MDLSC algorithm and M-SC are compared in Table 3. For different data sets, the MDLSC algorithm was mostly slightly higher than the M-SC for each index. Since the final feature matrix needed to be normalized, the similarity matrix calculated by the two had a small difference in value, and the improvement of the clustering index was not very great. However, the weighted Mahalanobis distance was better than the unweighted Mahalanobis distance spectral clustering algorithm in the judgment of individual data points, and the clustering effect of the MDLSC algorithm was stronger than that of the M-SC algorithm.

From Figures 1–4, comparing the same indicators on different artificial data sets, it can be seen that kernel SC, K-means, and LSR performed slightly worse in terms of the clustering effect. However, NJW and the MDLSC and M-SC algorithms proposed in this paper had good clustering performance, indicating that the improvement of the similarity measurement method played a role in the clustering performance. The algorithm in this paper had high index values for different data sets, and other algorithms only had high values for specific data sets; thus, MDLSC had better robustness in terms of overall stability. Overall, the MDLSC algorithm proposed in this paper outperformed other comparison algorithms on the data set.

Figures 5–7 show the distribution of eigenvectors of the flame, Jain, and spiral data sets selected by the algorithm in this paper using the weighted Mahalanobis distance. The distribution of each eigenvector, orthogonal relationship, and data discrimination was clear. Figures 8–10 show the distribution of eigenvectors obtained by the NJW algorithm. The eigenvectors were prone to inclusion and merging, and some of the orthogonal relationships were not clear; Therefore, the experimental results further show that the similarity matrix calculated by the Mahalanobis distance measure in the algorithm in this paper was more reasonable, and the eigenvectors of the Laplace matrix were calculated to preserve the distribution characteristics of the original data to the greatest extent, thereby improving the accuracy of the clustering algorithm. Comparing the above experimental results and the clustering results of the algorithm on the artificial data set, it can be observed that the clustering effect of the MDLSC algorithm on the experimental data set was better than that for other clustering algorithms.

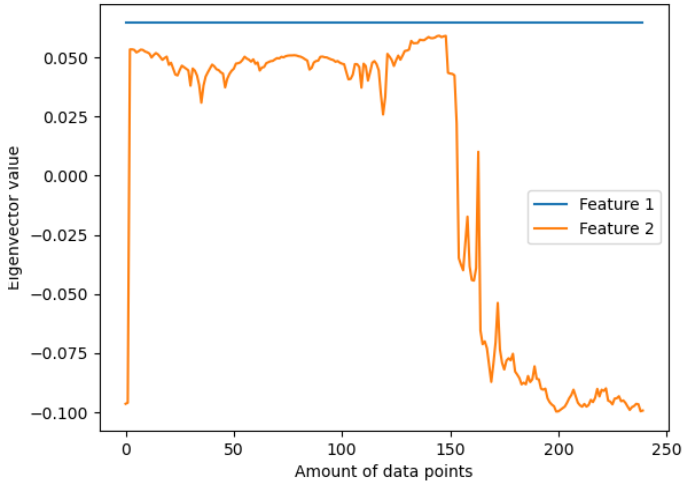

Figure 5. Distribution Map of Eigenvectors of MDLSC Algorithm in Flame Data Set.

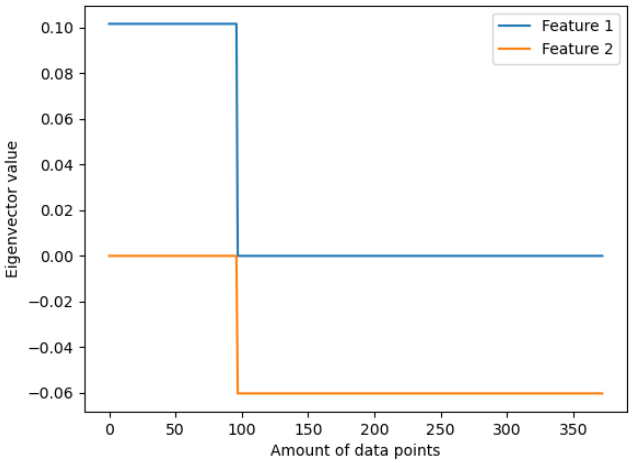

Figure 6. Distribution of Eigenvectors of MDLSC Algorithm in Jain Data Set.

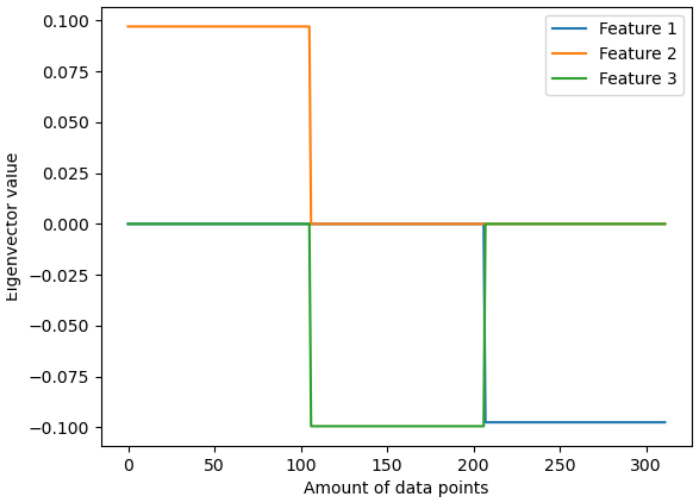

Figure 7. Distribution of MDLSC Algorithm Feature Vectors in Spiral Data Set.

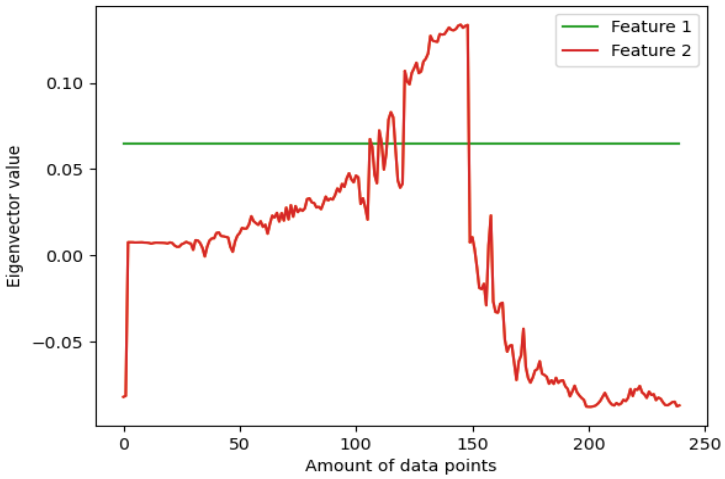

Figure 8. Distribution Map of Eigenvectors of MDLSC Algorithm in Flame Data Set.

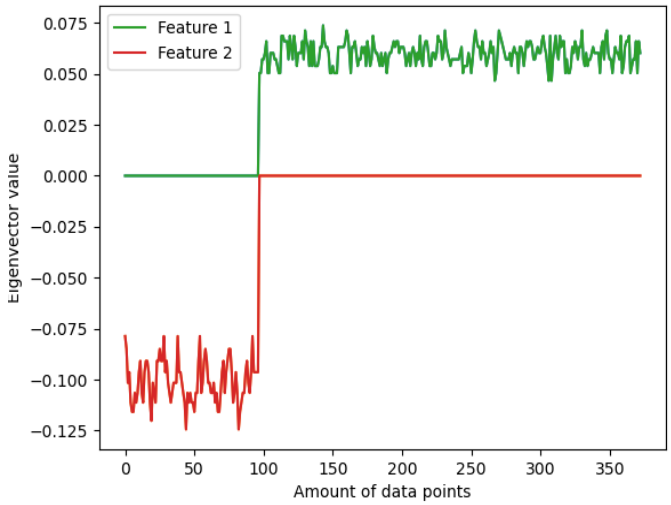

Figure 9. Distribution Map of Eigenvectors of NJW Algorithm in Jain Data Set.

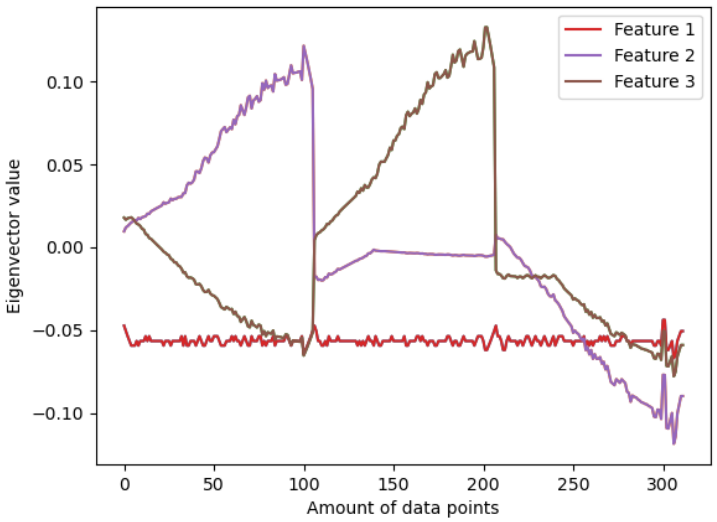

Figure 10. Distribution of NJW Algorithm Feature Vectors in Spiral Data Set.

4.3.2. Experiment on UCI Data Set

In order to further verify the effectiveness of the algorithm in this paper, four data sets on UCI were selected for verification. Table 2 lists the basic information of these four data sets. Figure 11 lists the RI evaluation index comparison chart of the algorithms and Figure 12 lists the F1-Score evaluation index comparison chart of the four algorithms.

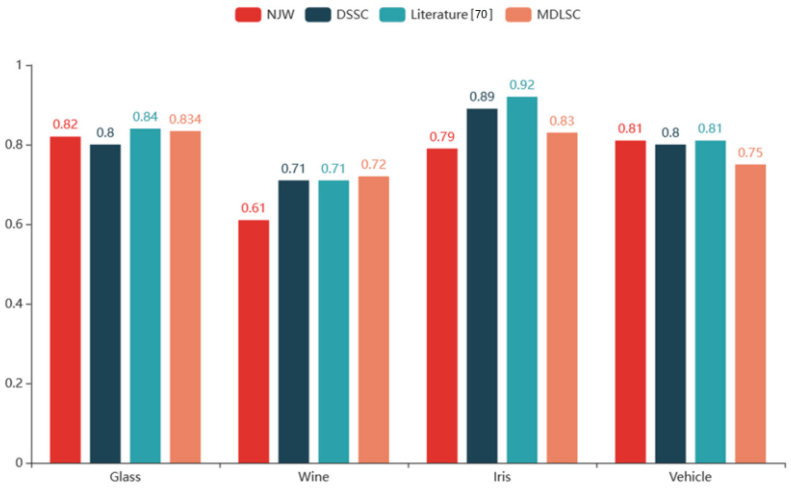

Figure 11. Comparison of RI Evaluation Indicators for UCI Data Sets [70].

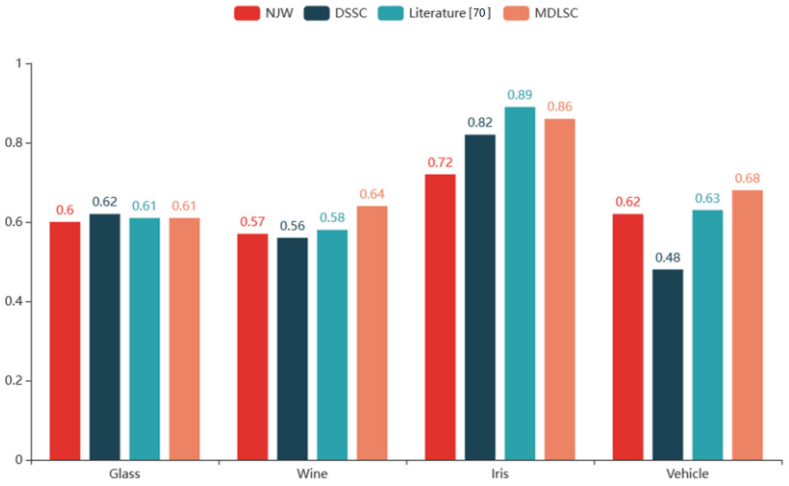

Figure 12. Comparison of F1 Scores for UCI Data Sets [70].

It can be seen from Figures 11 and 12 that for the glass data set, the DSSC algorithm had a slightly higher F1 score than the algorithm in this paper. But, the RI evaluation value was lower than the MDLSC algorithm in this paper. However, for the wine data set, the algorithm in this paper was higher than other literature algorithms in terms of F1 score and RI evaluation index. For the iris data set, although the RI index was not as good as other algorithms, the accuracy of the F1 score obtained was higher than that of the NJW and DSSC algorithms and was slightly lower than that of the literature algorithm [48]. For the vehicle data set, the RI index was lower than other algorithms, but the F1 score value was higher than other algorithms. From the comprehensive comparison of the two indicators of the UCI data set, because the algorithm in this paper improved the calculation method of similarity and fully described the similarity relationship between data points, the algorithm in this paper was relatively stable under the clustering index and was better than other algorithms.

4.4. Experimental Results and Analysis

Through the comparison and verification of the experimental results of the above artificial and UCI data sets, the weighted Mahalanobis distance spectrum clustering algorithm proposed in this paper achieved good clustering results. For the artificial data set, experiments were carried out by comparing the clustering index, clustering effect, and influence of eigenvectors. Firstly, the numerical analysis and comparison of the clustering algorithm indexes ACC, NMI, purity, and F1 score obtained from the experiment showed that the clustering quality was better than the traditional spectral clustering algorithm. In addition, comparing the numerical distribution of five algorithms under the clustering index for different data sets showed the robustness of the algorithm in this paper. The K-means algorithm, NJW algorithm, and MDLSC algorithm were compared using the clustering effect diagram. From a data center point of view, the clustering effect of the algorithm in this paper was better than that of the classical partitioned K-means algorithm and NJW algorithm. Secondly, by comparing the distribution of the eigenvectors of the NJW and MDLSC, it is concluded that the orthogonal relationship of the eigenvectors of the algorithm in this paper is clear and better than that of the traditional spectral clustering algorithm in the process of clustering selection. For the UCI data set, the RI and F1 score of the algorithm in this paper and the improved algorithm in the literature were displayed through histograms. A comprehensive comparison of the data showed that the indicators were high and low. But, the algorithm in this paper was more stable. A comprehensive comparison of the algorithm in this paper fully reflected the spatial characteristics of the data and the strong robustness and stable handling of the clustering process and showed that it is superior to other algorithms in terms of clustering effect.

In traditional spectral clustering algorithms, Euclidean distance or the Gaussian kernel function is usually used to calculate the similarity between data points, but these calculation methods cannot capture the linear correlations between data points. Mahalanobis distance calculates the distance through covariance, which can capture the correlations between data points, such as positive correlation, negative correlation, and irrelevance, but, in solving practical problems, it is necessary to reflect the degree of linear correlation between data, and normalization of the covariance is used, that is, the correlation coefficient as the weight of the Mahalanobis distance can capture the strength of the linear correlation between data points to achieve a more accurate measure of the similarity between data points.

However, it should be noted that the spectral clustering method using the Mahalanobis distance weighted by the correlation coefficient also has certain limitations and applicable conditions. Firstly, the correlation coefficient can only reflect the degree of linear correlation and may not be suitable for data with nonlinear correlations. Secondly, since the MDLSC algorithm involves calculations such as covariance matrix, correlation coefficient matrix, regularized Laplacian matrix, and matrix eigendecomposition, MDLSC may face the problem of computational complexity when processing large-scale data sets.

Therefore, when choosing to use the MDLSC algorithm, it must be evaluated according to specific problems and data characteristics. If the data have clear correlations, and the linear correlation can better describe the similarity between the data, then MDLSC may be an applicable choice in, for example, image processing, social network analysis, natural language processing, and other domain problems. However, for non-linear correlations or large-scale data sets, it may be necessary to consider other clustering methods or employ approximation algorithms to improve efficiency.

5. Conclusions and Future Work

This paper proposes a weighted Mahalanobis distance spectral clustering algorithm to solve the inaccuracy of similarity measurements in traditional spectral clustering algorithms and ignore the relationships between data points. Since the construction of the similarity matrix is very important to the clustering effect of the spectral clustering algorithm, the accuracy of the algorithm can be improved by improving the similarity measurement. Weighted Mahalanobis distance is used to construct the similarity matrix and then the

eigenvectors are obtained by calculating the regularized Laplacian matrix, which overcomes the problems whereby (1) a traditional spectral clustering algorithm only describes the similarity by distance and (2) the feature space is not stable enough. The clustering accuracy is improved, and the clustering effect is good. Various experiments were designed for artificial and UCI data sets, and the results show that the algorithm is better in terms of clustering quality, clustering effect, and the distribution of feature space.

However, since the calculation of Mahalanobis distance is more troublesome than the original Euclidean distance calculation, and with the advent of the era of big data, the time complexity of running the algorithm on a single computer is relatively large. How to reduce the computational complexity and arrange it on big data platforms is the next research focus of the algorithm. The next step is to use the distributed computing framework Apache Spark to design and implement a distributed weighted Mahalanobis distance spectral clustering algorithm and distribute the computing and storage tasks of the algorithm to multiple computing nodes for parallel processing. The scalability and computational efficiency of the algorithm will be improved.

Author Contributions: Conceptualization, L.Y. and L.L.; methodology, L.Y. and L.L.; software, L.Y., Y.Q. and L.L.; validation, L.Y., D.W., W.D. and L.L.; formal analysis, L.L.; resources, D.W., L.Y. and H.C.; data curation, L.L.; writing—original draft preparation, L.L.; writing—review and editing, H.C., W.D. and Y.Q.; visualization, Y.Q.; funding acquisition, H.C. and D.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Natural Science Foundation of Sichuan Province under grant 2022NSFSC0536 and the Project of Wenzhou Key Laboratory Foundation, China under grant 2021HZSY0071.

Data Availability Statement: Not applicable.

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

References

1. Wang, J.; Wang, S.; Deng, Z. Several Problems in Cluster Analysis Research. *Control Decis.* **2012**, *27*, 8.
2. Lei, X.; Xie, K.; Lin, F.; Xia, Z. An Efficient Clustering Algorithm Based on K-means Local Optimality. *J. Softw.* **2008**, *19*, 1683–1692. [CrossRef]
3. Liu, Z.; Wu, P.; Wu, Y. Research on three spectral clustering algorithms and their applications. *Comput. Applicat. Res.* **2017**, *34*, 1026–1031.
4. Miguel, C. On the diameter of the commuting graph of the matrix ring over a centrally finite division ring. *Linear Algebra Its Applicat.* **2016**, *509*, 276–285. [CrossRef]
5. Zhou, X.; Ma, H.; Gu, J.; Chen, H.; Deng, W. Parameter adaptation-based ant colony optimization with dynamic hybrid mechanism. *Eng. Appl. Artif. Intell.* **2022**, *114*, 105139. [CrossRef]
6. Zhang, J.M.; Sheny, X. Review on spectral methods for clustering. In Proceedings of the 2015 34th Chinese Control Conference (CCC), Hangzhou, China, 28–30 July 2015; pp. 3791–3796.
7. Che, W.F.; Feng, G.C. Spectral clustering: A semi-supervised approach. *Neuro Comput.* **2012**, *77*, 119–228.
8. Zhao, Y.C.; Zhang, S.C. Generalized Dimension-Reduction Frame work for Recent-Biased Time Series Analysis. *IEEE Trans. Knowl. Data Eng.* **2006**, *18*, 231–244. [CrossRef]
9. Langone, R.; Mall, R.; Alzate, C.; Suykens, J.A.K. *Kernel Spectral Clustering and Applications*; Unsupervised Learning Algorithms; Springer International Publishing: Berlin/Heidelberg, Germany, 2016.
10. Shi, B.; Guo, Y.; Hu, Y.; Yu, J. Multi-scale spectral clustering algorithm. *Comput. Eng. Applicat.* **2011**, *47*, 128–132.
11. Fisher, D.H. Knowledge acquisition via incremental conceptual clustering. *Machine Learn.* **1987**, *2*, 139–172. [CrossRef]
12. Shi, J.; Malik, J.M. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 888–905.
13. Ng, A.Y.; Jordan, M.I.; Weiss, Y. On spectral clustering: Analysis and an algorithm. In Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic, Vancouver, BC, Canada, 3–8 December 2001.
14. Kong, W.Z.; Sun, Z.H.; Yang, C.; Sun, C. Automatic spectral clustering based on eigengap and orthogonal eigenvectors. *Chin. J. Electron.* **2010**, *38*, 1880–1885.
15. Zhao, X.X.; Zhou, Z.P. A Semi-Supervised Spectral Clustering Algorithm Combining Sparse Representation and Constraint Transfer. *J. Intell. Syst.* **2018**, *13*, 855–862.
16. Jia, Y.; Liu, H.; Hou, J.; Kwong, S.; Zhang, Q. Multi-view spectral clustering tailored tensor low-rank representation. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 4784–4797. [CrossRef]
17. Wang, L.; Feng, B.L.; Cheng, J.L. Density -sensitive spectral clustering. *Acta Electron. Sin.* **2007**, *35*, 1577–1581.

18. Wang, N.; Li, X. Active Semi-Supervised Spectral Clustering Algorithm Based on Supervised Information Characteristics. *J. Electron.* **2010**, *38*, 172–176.
19. Wang, X.Y.; Ding, S.F.; Jia, W.K. Active constraint spectral clustering based on Hessian matrix. *Soft Comput.* **2020**, *24*, 2381–2390. [CrossRef]
20. Klein, D.; Kamvar, S.D.; Manning, C.D. *From Instance-Level Constraints to Space-Level Constraints: Making the Most of Prior Knowledge in Data Clustering*; Stanford University: Stanford, CA, USA, 2002.
21. Wu, L.; Wen, G.; Tong, T.; Tan, M.L.; Du, T.-T. Spectral clustering algorithm combining local PCA and k-nearest neighbors. *Comput. Eng. Design* **2019**, *40*, 2204–2210.
22. Tao, X.; Wang, R.T.; Chang, R.; Huang, B.; Zhu, Q. Spectral Clustering Algorithm Based on Low Density Segmentation Density Sensitive Distance. *Chin. J. Automat.* **2020**, *46*, 1479–1495.
23. Ge, J.; Yang, G. Density Adaptive Neighborhood Spectral Clustering Algorithm Based on Shared Nearest Neighbors. *Comput. Eng.* **2021**, *47*, 116–123. [CrossRef]
24. Du, T.; Wen, G.; Wu, L.; Tong, T.; Tan, M. Spectral clustering algorithm based on local covariance matrix. *Comput. Eng. Applicat.* **2019**, *148–154*, 176.
25. Yang, X. Research on Imbalanced Data Undersampling Method Based on Spectral Clustering. *Comput. Digit. Eng.* **2021**, *49*, 2305–2309+2330.
26. Lu, G.; Lin, Z.; Yan, S.; Sun, J.; Yu, Y.; Ma, Y. Robust recovery of subspace structures by low-rank representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *35*, 171–184. [CrossRef] [PubMed]
27. Elhamifar, E.; Vidal, R. Sparse subspace clustering. In Proceedings of the 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Miami, FL, USA, 20–25 June 2009; pp. 2790–2797.
28. Sun, S. *Subspace Clustering and Its Application*; Xi'an University of Architecture and Technology: Xi'an, China, 2016.
29. Kim, J.H.; Choi, J.H.; Park, Y.H.; Leung, C.K.S.; Nasridinov, A. KNN-SC: Novel spectral clustering algorithm using k-nearest neighbors. *IEEE Access* **2021**, *9*, 152616–152627. [CrossRef]
30. Zhang, X.; Li, J.; Yu, H. Local density adaptive similarity measurement for spectral clustering. *Patt. Recognit. Lett.* **2011**, *32*, 352–358. [CrossRef]
31. Nataliani, Y.; Yang, M.S. Powered Gaussian kernel spectral clustering. *Neural Comput. Applicat.* **2019**, *31*, 557–572. [CrossRef]
32. Berahmand, K.; Mohammadi, M.; Faroughi, A.; Mohammadiani, R.P. A novel method of spectral clustering in attributed networks by constructing parameter-free affinity matrix. *Cluster Comput.* **2022**, *25*, 869–888. [CrossRef]
33. Jiang, Z.; Liu, X. Adaptive KNN and graph-based auto-weighted multi-view consensus spectral learning. *Informat. Sci.* **2022**, *609*, 1132–1146. [CrossRef]
34. Yin, L.; Li, M.; Chen, H.; Deng, W. An Improved Hierarchical Clustering Algorithm Based on the Idea of Population Reproduction and Fusion. *Electronics* **2022**, *11*, 2735. [CrossRef]
35. Ren, Z.; Zhen, X.; Jiang, Z.; Gao, Z.; Li, Y.; Shi, W. Underactuated control and analysis of single blade installation using a jackup installation vessel and active tugger line force control. *Mar. Struct.* **2023**, *88*, 103338. [CrossRef]
36. Song, Y.; Zhao, G.; Zhang, B.; Chen, H.; Deng, W.Q.; Deng, Q. An enhanced distributed differential evolution algorithm for portfolio optimization problems. *Eng. Appl. Artif. Intell.* **2023**, *121*, 106004. [CrossRef]
37. Sun, Q.; Zhang, M.; Zhou, L.; Garme, K.; Burman, M. A machine learning-based method for prediction of ship performance in ice: Part I. ice resistance. *Mar. Struct.* **2022**, *83*, 103181. [CrossRef]
38. Li, X.; Zhao, H.; Deng, W. BFOD: Blockchain-based privacy protection and security sharing scheme of flight operation data. *IEEE Internet Things J.* **2023**. [CrossRef]
39. Yu, Y.; Tang, K.; Liu, Y. A fine-tuning based approach for daily activity recognition between smart homes. *Appl. Sci.* **2023**, *13*, 5706. [CrossRef]
40. Yu, C.; Gong, B.; Song, M.; Zhao, E.; Chang, C.I. Multiview Calibrated Prototype Learning for Few-shot Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5544713. [CrossRef]
41. Chen, H.; Wang, T.; Chen, T.; Deng, W. Hyperspectral Image Classification Based on Fusing S3-PCA, 2D-SSA and Random Patch Network. *Remote Sens.* **2023**, *15*, 3402. [CrossRef]
42. Huang, C.; Zhou, X.; Ran, X.; Wang, J.; Chen, H.; Deng, W. Adaptive cylinder vector particle swarm optimization with differential evolution for UAV path planning. *Eng. Appl. Artif. Intell.* **2023**, *121*, 105942. [CrossRef]
43. Xie, C.; Zhou, L.; Ding, S.; Liu, R.; Zheng, S. Experimental and numerical investigation on self-propulsion performance of polar merchant ship in brash ice channel. *Ocean Eng.* **2023**, *269*, 113424. [CrossRef]
44. Cai, J.; Ding, S.; Zhang, Q.; Liu, R.; Zeng, D.; Zhou, L. Broken ice circumferential crack estimation via image techniques. *Ocean Eng.* **2022**, *259*, 111735. [CrossRef]
45. Zhu, G.; Huang, S.; Yuan, C.; Huang, Y.H. SCoS: Design and Implementation of Parallel Spectral Clustering Algorithm Based on Spark. *Chin. J. Comput.* **2018**, *41*, 868–885.
46. Bai, L.; Zhao, X.; Kong, Y.; Zhang, Z.; Shao, J.; Qian, Y. Review of Spectral Clustering Algorithms Research. *Comput. Eng. Applicat.* **2021**, *57*, 15–26.
47. Chen, D.; Liu, J. Joint Learning of k-means and Spectral Clustering Based on Multiplicative Update Rule. *J. Nanjing Univ. (Nat. Sci. Ed.)* **2021**, *57*, 177–188.

48. Li, M.; Zhang, J.; Song, J.; Li, Z.; Lu, S. A clinical-oriented non severe depression diagnosis method based on cognitive behavior of emotional conflict. *IEEE Trans. Comput. Soc. Syst.* **2022**, *10*, 131–141. [CrossRef]
49. Duan, Z.; Song, P.; Yang, C.; Deng, L.; Jiang, Y.; Deng, F.; Jiang, X.; Chen, Y.; Yang, G.; Ma, Y.; et al. The impact of hyperglycaemic crisis episodes on long-term outcomes for inpatients presenting with acute organ injury: A prospective, multicentre follow-up study. *Front. Endocrinol.* **2022**, *13*, 1057089. [CrossRef] [PubMed]
50. Jin, T.; Zhu, Y.; Shu, Y.; Cao, J.; Yan, H.; Jiang, D. Uncertain optimal control problem with the first hitting time objective and application to a portfolio selection model. *J. Intell. Fuzzy Syst.* **2022**, *44*, 1585–1599. [CrossRef]
51. Li, M.; Zhang, W.; Hu, B.; Kang, J.; Wang, Y.; Lu, S. Automatic assessment of depression and anxiety through encoding pupil-wave from HCI in VR scenes. *ACM Trans. Multimed. Comput. Commun. Appl.* **2022**. [CrossRef]
52. Chen, M.; Shao, H.; Dou, H.; Li, W.; Liu, B. Data augmentation and intelligent fault diagnosis of planetary gearbox using ILoFGAN under extremely limited sample. *IEEE Trans. Reliab.* **2022**, 1–9. [CrossRef]
53. Zhou, X.; Cai, X.; Zhang, H.; Zhang, Z.; Jin, T.; Chen, H.; Deng, W. Multi-strategy competitive-cooperative co-evolutionary algorithm and its application. *Inf. Sci.* **2023**, *635*, 328–344. [CrossRef]
54. Chen, H.; Chen, Y.; Wang, Q.; Chen, T.; Zhao, H. A New SCAE-MT Classification Model for Hyperspectral Remote Sensing Images. *Sensors* **2022**, *22*, 8881. [CrossRef]
55. Chen, T.; Song, P.; He, M.; Rui, S.; Duan, X.; Ma, Y.; Armstrong, D.G.; Deng, W. Sphingosine-1-phosphate derived from PRP-Exos promotes angiogenesis in diabetic wound healing via the S1PR1/AKT/FN1 signalling pathway. *Burn. Trauma* **2023**, *11*, tkad003. [CrossRef]
56. Li, Y.; Xiang, G. A Linear Discriminant Analysis Classification Algorithm Based on Mahalanobis Distance. *Comput. Simulat.* **2006**, *23*, 86–88.
57. Yan, Z.; Zhang, Z.; Wang, Y.; Jin, Y.; Yan, T. Improved Deep Embedding Clustering Algorithm Based on Weighted Ma-halanobis Distance. *J. Comput. Applicat.* **2019**, *39*, 122–126.
58. Cai, J.; Xie, F.; Zhang, Y. A New Fuzzy Clustering Algorithm Based on Mahalanobis Distance Feature Weighting. *Comput. Eng. Applicat.* **2012**, *48*, 422.
59. Ma, Y.; Li, X.; Xue, S. A Fusion Algorithm of Spectral Clustering and Quantum Clustering Based on Manifold Distance Kernel. *J. Northwest Normal Univ. (Nat. Sci. Ed.)* **2023**, *59*, 37–46.
60. Fan, J.; Deng, X.; Liu, Y. A Spectral Clustering Algorithm Based on Fréchet Distance. *J. Guangdong Univ. Technol.* **2023**, *40*, 39–44.
61. Trillos, N.G.; Little, A.; McKenzie, D.; Murphy, J.M. Fermat Distances: Metric Approximation, Spectral Convergence, and Clustering Algorithms. *arXiv* **2023**, arXiv:2307.05750.
62. Zhang, Y.; Fang, K. *Introduction to Multivariate Statistical Analysis*; Science Press: Beijing, China, 1982.
63. Fiedler, M. Algebraic connectivity of graphs. *Czechoslovak Math. J.* **1973**, *23*, 298–305. [CrossRef]
64. Pang, Y.; Xie, J.; Nie, F.; Li, X. Spectral clustering by joint spectral embedding and spectral rotation. *IEEE Trans. Cybernet.* **2018**, *50*, 247–258. [CrossRef]
65. Cai, X.; Dai, G.; Yang, L. A Survey of Spectral Clustering Algorithms. *Comput. Sci.* **2008**, *35*, 14–18.
66. Cai, D.; He, X.; Han, J. Document clustering using locality preserving indexing. *IEEE Trans. Knowl. Data Eng.* **2005**, *17*, 1624–1637. [CrossRef]
67. Strehl, A.; Ghosh, J. Cluster ensembles—A knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.* **2002**, *3*, 583–617.
68. Wen, G. *Research on Spectral Clustering Method for High-Dimensional Data*; Guangxi Normal University: Guangxi, China, 2022. [CrossRef]
69. Zhang, M. *Research on the Evaluation Index of Symbolic Data Clustering*; Shanxi University: Taiyuan, China, 2013.
70. He, J. Spectral Clustering Algorithm for Improved Similarity Measurement. *J. Guilin Inst. Aerospace Eng.* **2017**, *22*, 123–127.