Neural Computing and Applications (2020) 32:9545–9565
https://doi.org/10.1007/s00521-019-04467-4

ORIGINAL ARTICLE

Multi-feature weighting neighborhood density clustering

Shuliang Xu[1] · Lin Feng[2] · Shenglan Liu[2] · Jian Zhou[1] · Hong Qiao[3,4]

Received: 22 January 2019 / Accepted: 26 August 2019 / Published online: 5 September 2019

Abstract

Clustering is an important data mining method to discover knowledge and patterns. Feature weighting is widely applied in high-dimensional data mining. In this paper, a multi-feature weighting neighborhood density clustering algorithm is proposed. It uses different dimension reduction algorithms to generate different features, and then, the weights of the features are determined by the discrimination ability. For the clustering algorithm, the center points can be selected by the upper approximation set and lower approximation set. At last, the final clustering result is from the fusion of multiple clustering results. The proposed algorithms and comparison algorithms are executed on the synthetic and real-world data sets. The test results show that the proposed algorithm outperforms the comparison algorithms on the most experimental data sets. The experimental results prove that the proposed algorithm is effective for data clustering.

Keywords Clustering analysis · Multi-feature · Neighborhood density · Rough set · Granular computing

1 Introduction

In the information society, it is often not easy to obtain data labels and unlabeled data set is common in many fields. How to mine knowledge and patterns from unlabeled data set has become a hot topic [16, 18, 25, 46]. Clustering analysis is a fundamental tool for unlabeled data mining [17, 24] and has been widely used in social networks, recommendation system, medical data analysis and image processing, etc [29, 39, 40, 43, 49]. High-dimensional data clustering is a challenging problem in cluster analysis. For traditional clustering algorithms, the time and memory requirement remarkably increase with the increase in the dimension of data set; therefore, they are inefficient to cluster high-dimensional data [21, 23, 47, 50]. In order to handle the challenge, many linear and nonlinear dimension reduction algorithms, including principal component analysis (PCA) [27], singular value decomposition (SVD) [19], nonnegative matrix factorization (NMF) [31], locally linear embedding (LLE) [37], Laplacian eigenmaps (LE) [4] and isometric feature mapping (ISOMAP) [41], etc., are introduced into clustering analysis [48]. However, the clustering methods with linear and nonlinear dimension reduction treat all dimensions as equally importance which is inconsistent with the fact because the importance of each dimension for different clusters is also different [45].

Feature weighting is to give a weight for each attribute according to the significance of the attribute, and the significant attribute has a larger weight; it is an effective way to respond to the challenge of high-dimensional data. Up to now, a series of related works have been published by many researchers [5, 14, 34, 38]. Jing et al. [26] propose an entropy weighting *k*-means algorithm; feature weighting is used to calculate the distance of two data points and the algorithm introduces a weight entropy into the objective optimization function; the final clustering result has a minimum intra-cluster distance and weight entropy. Deng et al. [13] develop Jing's work and present an enhanced

✉ Lin Feng
fenglin@dlut.edu.cn

Shuliang Xu
xushulianghao@126.com

[1] Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian, China

[2] School of Innovation and Entrepreneurship, Dalian University of Technology, Dalian, China

[3] Institute of Automation, Chinese Academy of Sciences, Beijing, China

[4] State Key Laboratory for Management and Control of Complex Systems, Beijing, China

soft subspace clustering; in order to measure the quality of clustering result comprehensively, the enhanced algorithm introduces three criteria into the objective optimization function; the first item is the intra-cluster compactness, the second item is the weight entropy, and the third item is the inter-cluster distance; by solving the optimization problem, a better clustering result is obtained. Boongoen et al. [7] present a filter clustering approach based on data reliability and propose three clustering algorithms; the approach utilizes feature weighting to generate clusters and the weights are determined by the data reliability which is calculated from neighbors; by iteratively solving parameters, a clustering algorithm based on partition called as R-KM is proposed; in addition, an inter-cluster similarity is defined based on the data reliability; by merging two similar clusters, a new hierarchical clustering algorithm called as R-CL is proposed; Boongoen et al. also use the data reliability to define the similarity matrix of data points and generate a new spectral clustering algorithm called R-SPT. Chen et al. [9] propose a feature group weighting clustering method for high-dimensional data; the method employs a double weighting mechanism; the feature set is divided into several feature subsets and each feature subset is used to produce a clustering result with feature weighting mechanism; then, the clustering results of different feature subsets also employ weighting mechanism; the final result is the fusion of different clustering results. Guo et al. [20] propose a soft subspace clustering algorithm with an improved feature weight self-adjustment mechanism which is a development of *k*-means, the weights of features for each cluster can be adaptively updated according to the importance of the feature for clustering quality, and the algorithm minimizes the intra-cluster compactness and maximizes the inter-cluster separation in the dimensional space. Ren et al. [36] propose a weighted adaptive mean shift clustering algorithm; the algorithm is a development of mean shift algorithm [11] and uses a weighted distance to obtain a bandwidth; the initial modes are generated by mean shift algorithm; then, it uses pattern matching to complete the clustering process; if a data point does not belong to any existing cluster, the data point is seen as a new cluster; the weighted adaptive mean shift clustering algorithm can effectively avoid the influence of noise data points. Bai et al. [3] propose two dissimilarity for categorical data set and developed two clustering algorithms called as MKM_NOF and MKM_NDM, respectively; the feature weighting mechanism is introduced into the two algorithms; the weight of each attribute is determined by the frequency of data points on the attribute; the two algorithms produce a clustering result with minimum intra-cluster dissimilarity.

The above works have promoted the development of high-dimensional data clustering; however, it is not easy for the partitional clustering algorithm to determine the initial central points; in addition, single feature cannot accurately represent data points because the information provided by single feature is limited and many ensemble clustering algorithms assign equal weights for different clustering results which cannot reflect the quality differences of different clustering results. It has been proved that ensemble learning with multiple features can improve the performance of algorithm because multiple features can give different views to describe samples [32, 51, 52]. Therefore, a multi-feature weighting neighborhood density clustering method (MWND) is proposed in this paper. MWND employs multiple dimension reduction algorithms to reduce the dimension of data points and generate different features which can be seen as different views; the weight of each attribute is determined by the discrimination ability of the attribute. In order to select suitable initial center points, MWND uses neighborhood rough set to evaluate the importance of each data point and the data points with large importance are selected as initial center points; after multiple clustering results are generated by weighted soft subspace clustering, the weights of the clustering results are determined by the compactness and granularity; the final result is the fusion of the multiple clustering results. The main contributions of this paper are as follows:

- We extend *k*-means algorithm and develop a multi-feature weighting neighborhood density clustering algorithm with soft subspace clustering. We uses multiple features to represent data points and improve the performance of MWND algorithm by ensemble learning mechanism.
- We introduce neighborhood rough set and employ neighborhood density to evaluate the importance of each data point. The initial center points are selected by the upper approximation set and lower approximation set, and it can accelerate the algorithm convergence speed and alleviate the clustering algorithm to fall into local optimal solution.
- We use compactness and granularity to determine the weights of the clustering results of different features and utilize the advantage of ensemble learning to obtain multiple clustering results from different views. The final clustering result is the fusion of different clustering results.

The remainder of the paper is organized as follows: Sect. 2 gives brief introductions about neighborhood rough set and *k*-means clustering algorithm; the detail theories and steps of MWND are explained in Sect. 3; the proposed algorithm and comparison algorithms are executed on the real and artificial data sets, and the experimental results are shown in Sect. 4; Sect. 5 concludes the discussions of this paper and gives some future research directions.

2 Background knowledge

In this section, some related concepts and notations about neighborhood rough set and k-means clustering algorithm are reviewed. Section 2.1 will explain the theory of neighborhood rough set and Sect. 2.2 will introduce the steps of k-means algorithm.

2.1 Neighborhood rough set

Neighborhood rough set is an effective information processing tool for numerical data. The advantage of neighborhood rough set is the ability to acquire the knowledge which is based on uncertainty measurement. It claims that the objects should belong to the same class if the conditional attributes of the object have similar values; otherwise, the consistency of the classification is violated [22]. The following definitions can be seen from References [1, 22].

Let $U = \{x_1, x_2, \cdots, x_n\}$ be an universe and $A = \{a_1, a_2, \cdots, a_m\}$ be an attribute set. In a m-dimensional Euclidean space, for $\forall x_i, x_j \in U$, $x_i = [x_{i1}, x_{i2}, \cdots, x_{im}]$ and $x_j = \left[x_{j1}, x_{j2}, \cdots, x_{jm}\right]$, the distance of x_i and x_j is as follows:

$$\varDelta(x_i, x_j) = \left(\sum_{k=1}^{m}\left(x_{ik} - x_{jk}\right)^2\right)^{\frac{1}{2}} \tag{1}$$

Let $\langle U, \varDelta\rangle$ be a metric space, for $\forall x \in U$, $\delta \geq 0$, the δ-neighborhood of x is defined as

$$\delta(x) = \{y | \varDelta(x, y) \leq \delta, y \in U\} \tag{2}$$

The neighborhood of x is a closed sphere; x is as the center point and δ is as the radius. If $\forall x, y \in U$ and $y \in \delta(x)$, it is said that x and y have a neighborhood relationship NR.

Let $\langle U, NR\rangle$ be a neighborhood approximation space and $X \subseteq U$, the upper approximation $\overline{NR}X$ and lower approximation $\underline{NR}X$ of X in the neighborhood approximation space are defined as

$$\begin{aligned} &\overline{NR}X = \{x_i | \delta(x_i) \cap X \neq \varnothing, x_i \in U\} \quad \text{and} \\ &\underline{NR}X = \{x_i | \delta(x_i) \subseteq X, x_i \in U\} \end{aligned} \tag{3}$$

It is obvious that for $\forall X \subseteq U$, $\underline{NR}X \subseteq X \subseteq \overline{NR}X$. $\underline{NR}X$ is also called as the positive region of X in the approximation space which is denoted as $POS(X) = \underline{NR}X$. The positive region of X is the set which includes the objects which belong to the neighborhood of X. The negative region $NEG(X)$ of X is defined as

$$NEG(X) = U - \overline{NR}X = \{x_i | \delta(x_i) \cap X = \varnothing\} \tag{4}$$

The negative region means that the objects in the region are independent of X. The approximation boundary of the approximation space is called as boundary region and the boundary region of X is defined as

$$BN(X) = \overline{NR}X - \underline{NR}X \tag{5}$$

If $BN(X) = \varnothing$, it means $\underline{NR}X = \overline{NR}X$ and it is said that X is crisp in the approximation space $\langle U, NR\rangle$. If $BN(X) \neq \varnothing$, it means $\underline{NR}X \neq \overline{NR}X$ and it is said that X is rough in the approximation space $\langle U, NR\rangle$.

For a neighborhood information system $NDT = \langle U, A, f\rangle$ and $B \subseteq A$, U is partitioned into n equivalence classes by B which means $U = \{X_1, X_2, \cdots, X_n\}$, the partition granularity of U by B is defined as

$$GK(B) = \frac{1}{|U|^2}\sum_{i=1}^{n}\frac{|X_i|^2}{|U|^2} \tag{6}$$

where f is an information function which establishes a mapping between the object sets and attribute sets and $f(x, a)$ means the attribute value of x on the attribute a. Granularity is a measure of the fineness of knowledge. The smaller the granularity is, the finer the partition of the universe will be [35, 44]. It is obvious that $\frac{1}{|U|^3} \leq GK(B) \leq \frac{1}{|U|^2}$. When $X_i = \{x_i\}(i = 1, 2, \cdots, n)$, $GK(B)$ is minimum and $GK(B) = \frac{1}{|U|^3}$; when $X_i = U$ and $X_j = \varnothing(j = 1, 2, \cdots, n, j \neq i)$, $GK(B)$ is maximum and $GK(B) = \frac{1}{|U|^2}$.

2.2 *k*-means clustering algorithm

k-means algorithm [33] is one of the most common clustering algorithms, and it is a partition-based clustering algorithm, which uses distance as the similarity measure. For two data points, if the distance is smaller, the similarity will be higher and they are more likely in the same cluster. k-means algorithm randomly selects k initial center data points and minimizes the following objective optimization function:

$$\begin{aligned} &\min_{U,V} F(U, V) = \sum_{i=1}^{k}\sum_{j=1}^{n} u_{ij}\sum_{r=1}^{m}\left(x_{jr} - v_{ir}\right)^2 \\ &\text{s.t. } u_{ij} \in \{0, 1\}, \sum_{i=1}^{k} u_{ij} = 1 \text{ and } 0 \leq \sum_{j=1}^{n} u_{ij} \leq n \end{aligned} \tag{7}$$

where $U = \left[u_{ij}\right]_{k \times n}$ is the partition matrix; $\boldsymbol{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_n]$; $\boldsymbol{v}_i$ is the center point of the ith cluster and $\boldsymbol{v}_i = [v_{i1}, v_{i2}, \cdots, v_{im}]^{\mathrm{T}}$.

k-means algorithm needs to iteratively generate clusters and update cluster center data points. In each iteration, the clustering center points are recalculated, and then, a new clustering result is obtained by repartitioning the data points. If the number of iterations reaches a predefined threshold or the difference of *F(U,V)* between the two iterations is less than a threshold, the algorithm terminates iteration and the current clustering result is the final clustering result.

3 Multi-feature weighting neighborhood density clustering algorithm

In this section, we will firstly present the method generating multiple features and then explain how to select the initial clustering center points according to neighborhood rough set; finally, we introduce the detail clustering processes of MWND. The structure of MWND algorithm is given in Fig. 1.

3.1 Formation of multi-feature

For high-dimensional data clustering, dimension reduction is an effective way to solve this problem. By using multiple dimension reduction algorithms, it can obtain multiple descriptions of a data set from multiple views. In this paper, PCA [27], LLE [37], LE [4] and NMF [31] are employed to decrease the number of dimensions and generate multiple features. Let $\boldsymbol{X}$ be the data matrix; PCA is to find a set of orthogonal feature vectors, so that the variance of the data projection in those directions reaches the maximum. The objective optimization function of PCA is as follows:

$$\max_{U} \boldsymbol{U}^T\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{U} \quad \text{s.t. } \boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I} \tag{8}$$

After solving the matrix $\boldsymbol{U}$, the new matrix is $\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{X}$. LLE and LE are manifold learning dimension reduction methods and the data points can be reconstructed according to their neighbors in the low-dimensional space. For LLE algorithm, the new data $\boldsymbol{Y}$ can be solved by the following optimization problem:

$$\min_{Y} F(\boldsymbol{Y}) = \sum_{i=1}^{n}\left\|\boldsymbol{y}_i - \sum_{j=1}^{k} w_{ij}\boldsymbol{y}_j\right\|_2^2 \quad \text{s.t. } \sum_{i=1}^{n}\boldsymbol{y}_i = \boldsymbol{0}, \quad \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{y}_i\boldsymbol{y}_i^T = \boldsymbol{I} \tag{9}$$

where w_{ij} is the weight of the *j*th *k*-nearest neighbor of $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ is the low-dimensional projection of $\boldsymbol{x}_i$. For LE algorithm, $\boldsymbol{Y}$ can be solved by the following problem:

$$\min_{Y} F(\boldsymbol{Y}) = \boldsymbol{tr}(\boldsymbol{Y}^T\boldsymbol{L}\boldsymbol{Y}) \quad \text{s.t. } \boldsymbol{Y}^T\boldsymbol{D}\boldsymbol{Y} = \boldsymbol{I} \tag{10}$$

where $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$ is the Laplacian matrix, $\boldsymbol{D}$ is the degree matrix of the graph which is constructed from the data $\boldsymbol{X}$ and $\boldsymbol{W}$ is the adjacency matrix of the graph. Nonnegative matrix factorization algorithm is to decompose a nonnegative matrix into two low-dimensional nonnegative matrices. It can be expressed as follows:

$$\min_{A,B} F(\boldsymbol{A},\boldsymbol{B}) = \left\|\boldsymbol{X} - \boldsymbol{A}\boldsymbol{B}^T\right\|_F^2 \quad \text{s.t. } \boldsymbol{A} \geq 0 \text{ and } \boldsymbol{B} \geq 0 \tag{11}$$

where $\boldsymbol{A}$ is the weight matrix between the original feature and the feature after dimension reduction; $\boldsymbol{B}$ is the data matrix after dimension reduction; $\boldsymbol{A} \in \mathbb{R}^{m\times d}$ and $\boldsymbol{B} \in \mathbb{R}^{n\times d}$. In order to solve $\boldsymbol{A}$ and $\boldsymbol{B}$, it can use alternating least squares (ALS) to solve the optimization problem [30, 48].

$$\boldsymbol{A} \leftarrow \boldsymbol{X}\boldsymbol{B}\left(\boldsymbol{B}^T\boldsymbol{B}\right)^{-1} \text{ and } \boldsymbol{B} \leftarrow \boldsymbol{X}^T\boldsymbol{A}\left(\boldsymbol{A}^T\boldsymbol{A}\right)^{-1} \tag{12}$$

For each attribute of a feature, it is obvious that each attribute has a different effect on different clusters; therefore, it can use weighting mechanism to reflect this property. For an attribute, it is known from unsupervised dimension reduction that the standard deviation of attribute values can represent the separating capacity of an attribute and a large standard deviation means a good separating capacity [12]; if data points can be well separated by the attribute, the weight of the attribute should be large; if data points are highly gathering together on the attribute, the weight of the attribute should be small. The degree of data separation can evaluate the discrimination ability of an attribute. Therefore, it can utilize standard deviation to

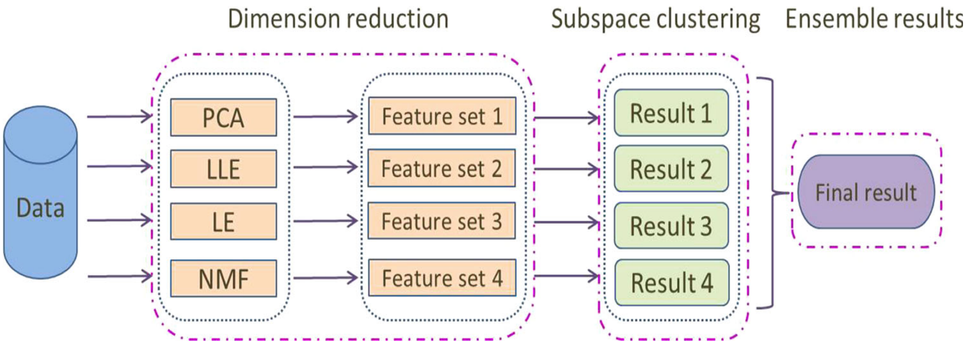


Fig. 1 The structure of MWND algorithm

present the discrimination ability. Let $\boldsymbol{f}_i$ be the ith attribute of $\boldsymbol{X}$, $\boldsymbol{X} = [\boldsymbol{f}_1, \boldsymbol{f}_2, \cdots, \boldsymbol{f}_d]$ and $\boldsymbol{f}_i = [f_{i1}, f_{i2}, \cdots, f_{in}]^{\mathrm{T}} (i = 1, 2, \cdots, d)$. The standard deviation of the ith attribute of $\boldsymbol{X}$ is as follows:

$$\delta_i = \sqrt{\frac{1}{n}\sum_{j=1}^{n}\left(f_{ij} - \bar{f}_i\right)^2} \quad \text{and} \quad \bar{f}_i = \frac{1}{n}\sum_{j=1}^{n} f_{ij} \tag{13}$$

The weight of $\boldsymbol{f}_i$ can be computed as follows:

$$\varpi_i = \frac{\delta_i}{\sum_{j=1}^{d}\delta_j} \tag{14}$$

Therefore, the weight of the feature $\boldsymbol{f}$ is $\boldsymbol{\varpi} = [\varpi_1, \varpi_2, \cdots, \varpi_d]^{\mathrm{T}}$. After multiple features are obtained by using PCA, LLE, LE and NMF, the weight matrix of the features can be computed by Eqs. (13)–(14). For each attribute, the weight represents the significance of an attribute; the large weight means that the attribute has a more important influence on clustering task.

3.2 The selection of initial center points

The initial clustering center points are important for clustering task; the good initial clustering center points can improve the performance of the clustering algorithm and accelerate the algorithm convergence speed [2, 8, 15, 28]. Let $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n\}$ and $\boldsymbol{x}_i = [x_{i1}, x_{i2}, \cdots, x_{id}]^{\mathrm{T}} (i = 1, 2, \cdots, n)$; for $\forall \boldsymbol{x}_i \in \boldsymbol{X}$, the neighborhood of $\boldsymbol{x}_i$ is computed as follows:

$$\delta(\boldsymbol{x}_i) = \left\{\boldsymbol{x}_j \mid \Delta(\boldsymbol{x}_i, \boldsymbol{x}_j) \le \delta, \boldsymbol{x}_j \in \boldsymbol{X}\right\} \tag{15}$$

It is obvious that the parameter δ has a great influence on the generation of neighborhood. A large δ has a trend to generate a large neighborhood, and a small δ has a trend to generate a small neighborhood. In order to evaluate δ, we randomly select N samples from X and δ is defined according to the selected samples.

$$\delta = \tau \cdot \sqrt{\frac{1}{N(N-1)}\sum_{i=1}^{N}\sum_{j=1}^{N}\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|_2^2} \tag{16}$$

where τ is predefined parameters and $\tau > 0$. For the data point $\boldsymbol{x}_i$, the neighborhood can be utilized to evaluate the density of $\boldsymbol{x}_i$. For $\forall \boldsymbol{x}_j \in \delta(\boldsymbol{x}_i)$, if $\delta(\boldsymbol{x}_j) \subseteq \delta(\boldsymbol{x}_i)$, it means that the neighborhood of $\boldsymbol{x}_j$ completely belongs to the neighborhood of $\boldsymbol{x}_i$ and $\delta(\boldsymbol{x}_j) \subseteq \underline{NR}(\delta(\boldsymbol{x}_i))$; therefore, the density of the data point $\boldsymbol{x}_i$ is updated as follows:

$$Den(\boldsymbol{x}_i) \leftarrow Den(\boldsymbol{x}_i) + \lambda \tag{17}$$

where λ is a predefined parameter and $\lambda \ge 0$. If $\delta(\boldsymbol{x}_j) \nsubseteq \delta(\boldsymbol{x}_i)$ and $\delta(\boldsymbol{x}_j) \cap \delta(\boldsymbol{x}_i) \ne \varnothing$, it means that the neighborhood of $\boldsymbol{x}_j$ partially belongs to the neighborhood of $\boldsymbol{x}_i$ and $\delta(\boldsymbol{x}_j) \subseteq \overline{NR}(\delta(\boldsymbol{x}_i))$; therefore, the density of the data point $\boldsymbol{x}_i$ is updated as follows:

$$Den(\boldsymbol{x}_i) \leftarrow Den(\boldsymbol{x}_i) + \lambda \cdot \frac{\left|\delta(\boldsymbol{x}_j) \cap \delta(\boldsymbol{x}_i)\right|}{\left|\delta(\boldsymbol{x}_j)\right|} \tag{18}$$

Then, sort $Den(\boldsymbol{x})$ in descending order and compare two adjacent densities in sequence; if the distance of the corresponding two points is larger than 2δ, the data point with larger density is selected as an initial clustering center point. Repeat the above steps until k clustering center points are selected or the center point set is not changed. The algorithm of selecting initial clustering center points is summarized in Algorithm 1.

Algorithm 1 Selecting initial clustering center points

```
Input: The data set $\boldsymbol{X}$, the number of clusters $k$ and the parameter $\lambda$;
Output: The $k$ center points $\boldsymbol{V}$.
Calculate the neighborhood radius $\delta$;
for each $\boldsymbol{x}_i \in \boldsymbol{X}$ do
    Calculate the neighborhood $\delta(\boldsymbol{x}_i)$;
    for each $\boldsymbol{x}_j \in \delta(\boldsymbol{x}_i)$ do
        if $\delta(\boldsymbol{x}_j) \subseteq \underline{NR}(\delta(\boldsymbol{x}_i))$ then
            $Den(\boldsymbol{x}_i) \leftarrow Den(\boldsymbol{x}_i) + \lambda$;
        else if $\delta(\boldsymbol{x}_j) \subseteq \overline{NR}(\delta(\boldsymbol{x}_i))$ then
            $Den(\boldsymbol{x}_i) \leftarrow Den(\boldsymbol{x}_i) + \lambda \cdot \frac{|\delta(\boldsymbol{x}_j) \cap \delta(\boldsymbol{x}_i)|}{|\delta(\boldsymbol{x}_j)|}$;
        end if
    end for
end for
Sort $Den(\boldsymbol{x})$ in descending order;
for each $Den(i) \in Den(\boldsymbol{x})$ do
    $i_0 \leftarrow \arg(Den(\boldsymbol{x}_{i_0}) == Den(i))$;
    $j_0 \leftarrow \arg(Den(\boldsymbol{x}_{j_0}) == Den(i+1))$;
    if $\|\boldsymbol{x}_{i_0} - \boldsymbol{x}_{j_0}\|_2 \ge 2\delta$ then
        if $length(\boldsymbol{V}) < k$ then
            $\boldsymbol{V} \leftarrow \boldsymbol{V} \cup \{\boldsymbol{x}_{i_0}\}$;
        end if
    end if
end for
```

From Algorithm 1, it is known that the initial clustering center points can be selected by neighborhood rough set. The data points which belong to lower approximation set have a larger weight in determining the clustering center points than the data points which belong to upper approximation set. The k selected data points not only have the largest density, but also stay as far as possible from each other. By Algorithm 1, it can reduce the blindness of selecting initial data center points and the fluctuation of the clustering result has been also reduced.

3.3 The clustering processes and the fusion of multiple clustering results

After the multiple features are generated, the initial clustering center points are obtained in each feature. The multiple different features generate multiple groups of clustering center points. For each feature, let $\boldsymbol{X} =$

$[\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n]$, $\boldsymbol{x}_i = [x_{i1}, x_{i2}, \cdots, x_{id}]^{\mathrm{T}} (i = 1, 2, \cdots, n)$, $\boldsymbol{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_k]$ and $\boldsymbol{v}_j = \left[v_{j1}, v_{j2}, \cdots, v_{jd}\right]^{\mathrm{T}}$, the objective optimization function is as follows:

$$\min_{U,V,W,\varpi} F(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}, \boldsymbol{\varpi}) = \sum_{j=1}^{k} \left[\sum_{i=1}^{n} u_{ji} \sum_{r=1}^{d} w_{jr} \varpi_r \left(x_{ir} - v_{jr}\right)^2 + \alpha \sum_{r=1}^{d} w_{kr} \log w_{kr} + \beta \sum_{r=1}^{d} \varpi_r \log \varpi_r \right] \tag{19}$$

$$\text{s.t. } u_{ji} \in \{0, 1\}, \sum_{j=1}^{k} u_{ji} = 1, 0 \leq w_{jr} \leq 1,$$

$$\sum_{r=1}^{d} w_{jr} = 1 \text{ and } \sum_{r=1}^{d} \varpi_r = 1$$

In Eq. (19), the first term is the compactness of clusters and the second term and the third term are the weight entropy. Therefore, it can employ EM algorithm [6] to solve the parameters. The Lagrange function of the objective optimization function is as follows:

$$L = \sum_{j=1}^{k} \left[\sum_{i=1}^{n} u_{ji} \sum_{r=1}^{d} w_{jr} \varpi_r \left(x_{ir} - v_{jr}\right)^2 + \alpha \sum_{r=1}^{d} w_{jr} \log w_{jr} \right] + \beta \sum_{r=1}^{d} \varpi_r \log \varpi_r - \mu \left(\sum_{r=1}^{d} w_{jr} - 1 \right) - \eta \left(\sum_{r=1}^{d} \varpi_r - 1 \right) \tag{20}$$

$\therefore \frac{\partial L}{\partial w_{jr}} = \sum_{i=1}^{n} u_{ji} \varpi \left(x_{ir} - x_{jr}\right)^2 + \alpha\left(\log w_{jr} + 1\right) - \mu = 0 \Rightarrow$ $\log w_{jr} = \frac{1}{\alpha}\left(\mu - \alpha - \sum_{i=1}^{n} u_{ji} \varpi \left(x_{ir} - v_{jr}\right)^2 \right)$

If $\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{\varpi}$ are given, $\boldsymbol{W}$ can be solved as follows:

$$\therefore w_{jr} = \exp\left(\frac{-\sum_{i=1}^{n} u_{ji} \varpi_r \left(x_{ir} - v_{jr}\right)^2}{\alpha} \right) \cdot \exp\left(\frac{\mu - \alpha}{\alpha}\right) \text{ and } \sum_{r=1}^{d} w_{jr} = 1 \tag{21}$$

$$\therefore \exp\left(\frac{\mu - \alpha}{\alpha}\right) = 1 / \sum_{r=1}^{d} \exp\left(\frac{-\sum_{i=1}^{n} u_{ji} \varpi_r \left(x_{ir} - v_{jr}\right)^2}{\alpha} \right) \tag{22}$$

$$w_{jr} = \exp\left(\frac{-\sum_{i=1}^{n} u_{ji} \varpi_r \left(x_{ir} - v_{jr}\right)^2}{\alpha} \right) \cdot 1 / \sum_{r=1}^{d} \exp\left(\frac{-\sum_{i=1}^{n} u_{ji} \varpi_r \left(x_{ir} - v_{jr}\right)^2}{\alpha} \right) \tag{23}$$

If $\boldsymbol{V}, \boldsymbol{W}, \boldsymbol{\varpi}$ are given, $\boldsymbol{U}$ can be solved as follows:

$$u_{ji} = \begin{cases} 1 & \text{if } \sum_{r=1}^{d} w_{jr} \varpi_r \left(x_{ir} - v_{jr}\right)^2 \leq \sum_{r=1}^{d} w_{tr} \varpi_r \left(x_{ir} - v_{tr}\right)^2 \\ & \text{for } 1 \leq t \leq k \\ 0 & \text{otherwise} \end{cases} \tag{24}$$

If $\boldsymbol{U}, \boldsymbol{W}, \boldsymbol{\varpi}$ are given, $\boldsymbol{V}$ can be solved as follows:

$$v_{jr} = \sum_{i=1}^{n} u_{ji} x_{ir} / \sum_{i=1}^{n} u_{ji} \quad \text{for } 1 \leq j \leq k, 1 \leq r \leq d \tag{25}$$

If $\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}$ are given, $\boldsymbol{\varpi}$ can be solved as follows:

$$\varpi_r = \exp\left(\frac{\eta - \beta}{\beta}\right) \cdot \exp\left(- \frac{\sum_{j=1}^{k} \sum_{i=1}^{n} u_{ji} w_{jr} \left(x_{ir} - v_{jr}\right)^2}{\beta} \right) \text{ and } \sum_{r=1}^{d} \varpi_r = 1 \tag{26}$$

Therefore, $\boldsymbol{\varpi}$ can be calculated as follows:

$$\varpi_r = \exp\left(- \frac{\sum_{j=1}^{k} \sum_{i=1}^{n} u_{ji} w_{jr} \left(x_{ir} - v_{jr}\right)^2}{\beta} \right) / \sum_{r'=1}^{d} \exp\left(- \frac{\sum_{j=1}^{k} \sum_{i=1}^{n} u_{ji} w_{jr'} \left(x_{ir'} - v_{jr'}\right)^2}{\beta} \right) \tag{27}$$

For each feature, the clustering processes can be summarized in Algorithm 2.

Algorithm 2 The clustering processes for sample feature

```
Input: The data set $X$, the number of clusters $k$, the weight of the feature $\varpi$; the parameters $\alpha$, $\beta$ and $\tau$;
Output: The clustering result.
Initialize $W$ and generate the initial clustering center points $V$ according to Algorithm 1;
while The objective function does not converge do
    Update $W$ by Eq. (23);
    Update $U$ by Eq. (24);
    Update $V$ by Eq. (25);
    Update $\varpi$ by Eq. (27);
end while
```

The multiple features will generate multiple clustering results. Let the clustering result of the ith feature be $\boldsymbol{C}_i = \{\boldsymbol{C}_{i1}, \boldsymbol{C}_{i2}, \cdots, \boldsymbol{C}_{ik}\}(i = 1, 2, \cdots, n_f)$, the granularity of the ith clustering result is defined as follows:

$$GK(\boldsymbol{C}_i) = \frac{1}{|\boldsymbol{X}|^2} \cdot \sum_{r=1}^{k} \frac{|\boldsymbol{C}_{ir}|^2}{|X|^2} \tag{28}$$

Therefore, it can employ granularity to evaluate the fineness of the clustering result. For a clustering result, we expect that the result is more compact with a minimum mean square error. The evaluation criterion of the clustering result for each feature is defined as follows:

$$\begin{aligned} E(\boldsymbol{C}_i) = & \frac{1}{|\boldsymbol{X}|} \cdot \sum_{r=1}^{k} \sum_{j=1}^{|C_{ir}|} \left\| \boldsymbol{x}_j - \boldsymbol{v}_i \right\|_2^2 \\ & + \frac{1}{|\boldsymbol{X}|^2} \cdot \sum_{r=1}^{k} \frac{|\boldsymbol{C}_{ir}|^2}{|X|^2} \end{aligned} \tag{29}$$

Therefore, the weight of the clustering result $\boldsymbol{C}_i$ is as follows:

$$\boldsymbol{cw}(i) = \frac{exp(-E(\boldsymbol{C}_i))}{\sum_{j=1}^{n_f} exp(-E(\boldsymbol{C}_j))} \tag{30}$$

where n_f is the number of features. Then, hierarchical clustering method is used to fuse the multiple clustering results. At first, each data point is seen as a cluster. Let $M \in \mathbb{R}^{n \times n}$ be a similarity matrix of data points. For $\forall \boldsymbol{x}_i, \boldsymbol{x}_j \in \boldsymbol{X}$, the similarity of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ in the clustering result $\boldsymbol{C}_r$ is calculated as follows:

$$\begin{aligned} & \varphi_r(\boldsymbol{x}_i, \boldsymbol{x}_j) \\ & = \begin{cases} \boldsymbol{cw}(r) & \text{if } \boldsymbol{x}_i, \boldsymbol{x}_j \text{ in the same cluster} (i \neq j) \\ 0 & \text{otherwise} \end{cases} \quad (r = 1, 2, \cdots, n_f) \end{aligned} \tag{31}$$

Therefore, the similarity of $\boldsymbol{M}$ is defined as follows:

$$\begin{aligned} & \boldsymbol{M}(i,j) \leftarrow \boldsymbol{M}(i,j) + \varphi(\boldsymbol{x}_i, \boldsymbol{x}_j) \quad \text{and} \\ & \varphi(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{r=1}^{n_f} \varphi_r(\boldsymbol{x}_i, \boldsymbol{x}_j) \end{aligned} \tag{32}$$

After some data points are merged, some new clusters are generated. For $\boldsymbol{C}_i, \boldsymbol{C}_j \subseteq \boldsymbol{X}$, the similarity of $\boldsymbol{C}_i$ and $\boldsymbol{C}_j$ is calculated as follows:

$$\begin{aligned} & \boldsymbol{M}(i,j) = \frac{1}{|\boldsymbol{C}_i| \cdot |\boldsymbol{C}_j|} \sum_{i'=1}^{|C_i|} \sum_{j'=1}^{|C_j|} \varphi\left(\boldsymbol{x}_{i'}, \boldsymbol{x}_{j'}\right), \\ & \boldsymbol{x}_{i'} \in \boldsymbol{C}_i, \boldsymbol{x}_{j'} \in \boldsymbol{C}_j \end{aligned} \tag{33}$$

If $\boldsymbol{x}_i \in \boldsymbol{X}, \boldsymbol{C}_j \subseteq \boldsymbol{X}$ and $\boldsymbol{x}_i \notin \boldsymbol{C}_j$, the similarity of $\boldsymbol{x}_i$ and $\boldsymbol{C}_j$ is calculated as follows:

$$\boldsymbol{M}(i,j) = \frac{1}{|\boldsymbol{C}_j|} \sum_{t=1}^{|C_j|} \varphi(\boldsymbol{x}_i, \boldsymbol{x}_t), \quad \boldsymbol{x}_t \in \boldsymbol{C}_j \tag{34}$$

Then, the clusters are merged until there are only k clusters and the k clusters are the final clustering result. The fusion of multiple clustering results is shown in Fig. 2. Therefore, the detail steps of MWND are summarized in Algorithm 3.

```
Algorithm 3 MWND
Input: The data set $\boldsymbol{X}$, the number of clusters $k$, the parameters $\alpha$, $\beta$ and $\tau$;
Output: The final clustering result.
Generate the four features and calculate the weight of each attribute as Section 3.1;
for each feature do
    Select initial clustering center points as Algorithm 1;
    Obtain the clustering result $\boldsymbol{C}_i$ as Algorithm 2;
    Obtain the weight of the clustering result as Eq. (30);
end for
Calculate the initial similarity matrix $\boldsymbol{M}$ as Eq. (32);
while The number of clusters is larger than $k$ do
    Merge two most similar clusters to a new cluster;
    Delete the rows and columns of the two clusters in $\boldsymbol{M}$;
    Add a row and a column into $\boldsymbol{M}$ which represents the new cluster;
    Calculate the similarity degree as Eq. (31), Eq. (33) or Eq. (34);
end while
```

From Algorithm 3, it is known that the fusion of multiple clustering results employs a weighted voting mechanism. The weight presents the reliability of the clustering result; the result with a large weight has a great influence on the final clustering result. For MWND algorithm, there are three weighted mechanisms. The first weight $\boldsymbol{\varpi}$ presents the discernibility ability of attributes. The second weight $\boldsymbol{W}$ presents the effect of an attribute on a cluster which generates the subspace of a cluster. The third weight $\boldsymbol{cw}$ presents the reliability of a clustering result. By the three

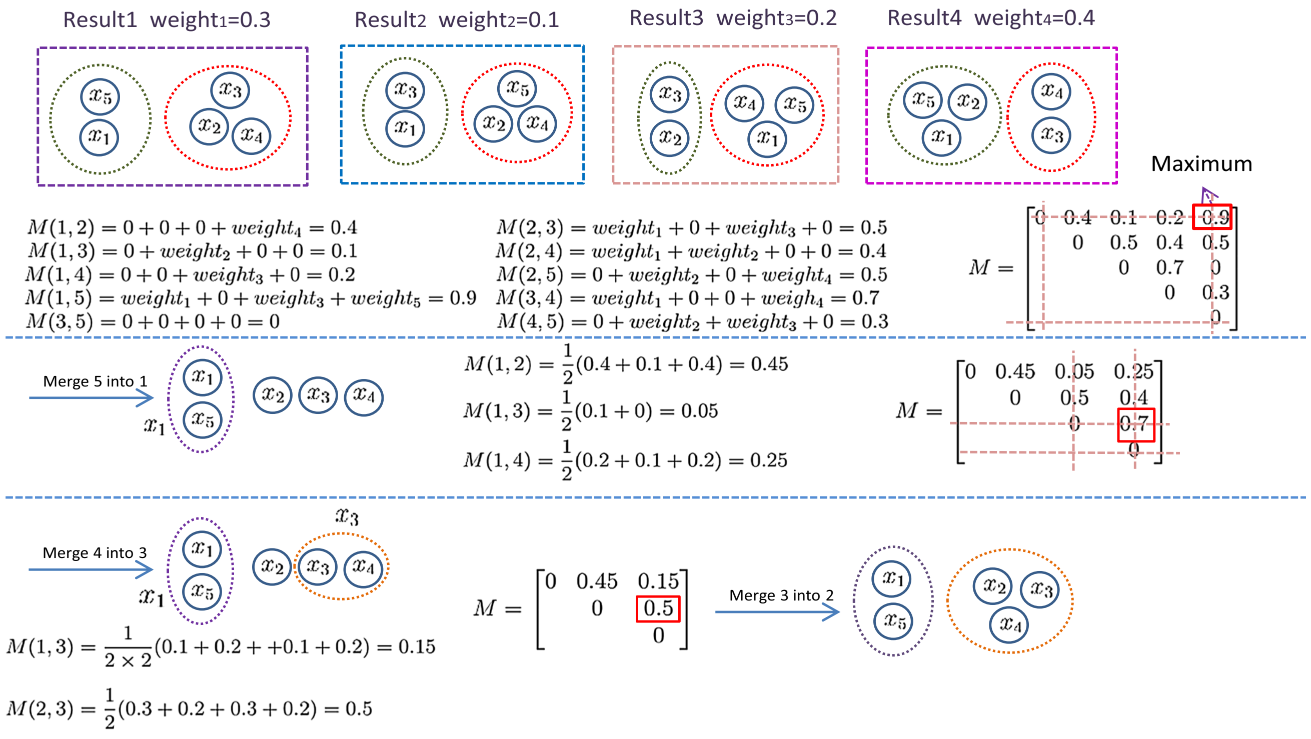

Result1 weight1=0.3
Result2 weight2=0.1
Result3 weight3=0.2
Result4 weight4=0.4
x_5 x_1 x_3 x_2 x_4
Maximum
$M(1,2) = 0+0+0+weight_4 = 0.4$
$M(1,3) = 0+weight_2+0+0 = 0.1$
$M(1,4) = 0+0+weight_3+0 = 0.2$
$M(1,5) = weight_1+0+weight_3+weight_5 = 0.9$
$M(3,5) = 0+0+0+0 = 0$
$M(2,3) = weight_1+0+weight_3+0 = 0.5$
$M(2,4) = weight_1+weight_2+0+0 = 0.4$
$M(2,5) = 0+weight_2+0+weight_4 = 0.5$
$M(3,4) = weight_1+0+0+weigh_4 = 0.7$
$M(4,5) = 0+weight_2+weight_3+0 = 0.3$
$M = \begin{bmatrix} 0 & 0.4 & 0.1 & 0.2 & \boxed{0.9} \\ & 0 & 0.5 & 0.4 & 0.5 \\ & & 0 & 0.7 & 0 \\ & & & 0 & 0.3 \\ & & & & 0 \end{bmatrix}$
Merge 5 into 1
x_1
$M(1,2) = \frac{1}{2}(0.4+0.1+0.4) = 0.45$
$M(1,3) = \frac{1}{2}(0.1+0) = 0.05$
$M(1,4) = \frac{1}{2}(0.2+0.1+0.2) = 0.25$
$M = \begin{bmatrix} 0 & 0.45 & 0.05 & 0.25 \\ & 0 & 0.5 & 0.4 \\ & & 0 & \boxed{0.7} \\ & & & 0 \end{bmatrix}$
Merge 4 into 3
x_1
x_3
$M(1,3) = \frac{1}{2\times 2}(0.1+0.2++0.1+0.2) = 0.15$
$M(2,3) = \frac{1}{2}(0.3+0.2+0.3+0.2) = 0.5$
$M = \begin{bmatrix} 0 & 0.45 & 0.15 \\ & 0 & \boxed{0.5} \\ & & 0 \end{bmatrix}$
Merge 3 into 2

Fig. 2 The fusion of multiple clustering results

weighted mechanisms and ensemble learning, the performance of the clustering result can be improved.

Theorem 1 Algorithm 3 is convergent within finite iterations.

Proof From Algorithm 3, it is known that MWND algorithm is made up of n_f Algorithm 2.

For Algorithm 2, let $\boldsymbol{U}_0, \boldsymbol{V}_0, \boldsymbol{W}_0$ and $\boldsymbol{\varpi}_0$ be t_0 iteration, $\boldsymbol{U}_1, \boldsymbol{V}_1, \boldsymbol{W}_1$ and $\boldsymbol{\varpi}_1$ be t_1 iteration $(t_0 < t_1)$ and $f(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}, \boldsymbol{\varpi})$ be the value of the objective function.

$\because$ From Eq. (19), it is known that $f(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}, \boldsymbol{\varpi})$ is a convex function

$\therefore f(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W})$ must exist a minimum

Suppose $f(\boldsymbol{U}_0, \boldsymbol{V}_0, \boldsymbol{W}_0, \boldsymbol{\varpi}_0) = f(\boldsymbol{U}_1, \boldsymbol{V}_1, \boldsymbol{W}_1, \boldsymbol{\varpi}_1)$

From Algorithm 2, it is known that there must be $f(\boldsymbol{U}_0, \boldsymbol{V}_0, \boldsymbol{W}_0, \boldsymbol{\varpi}_0) < f(\boldsymbol{U}_1, \boldsymbol{V}_1, \boldsymbol{W}_1, \boldsymbol{\varpi}_1)$ if Algorithm 2 is executed from t_0 iteration to t_1 iteration.

$\therefore$ The supposition contradicts the known condition.

$\therefore$ Algorithm 2 is convergent within finite iterations. $\therefore$ Algorithm 3 is convergent within finite iterations because MWND algorithm is made up of n_f Algorithm 2. □

In Algorithm 3, the search of neighborhood is $O(n^2)$; the determining method of neighborhood radius is $O(N^2)$; the determining density of the data point is $O(n)$; therefore, the complexity of the selection of initial center points is $O(n^2)$. In Algorithm 2, the complexity is the complexity of k-means algorithm which is $O(n \times I \times k \times m) \approx O(n)$ where I is the number of iterations; Eq. (28) is $O(k)$; Eq. (29) is less than $O(k \times n)$; the complexity of determining the weight of the clustering result is $O(n_f \times k \times n)$; the complexity of Eq. (32) is $O\left(n_f^2\right)$; the complexity of merging clustering results is $O\left(n_f^2\right)$; therefore, the complexity of MWND algorithm is $O(n^2)$.

4 Experimental results and analysis

In order to test the performance of MWND, we choose SCIFWSA [20], FSSCND [10], DIFSC [42] and LRGR [21] as comparison algorithms. MWND and comparison algorithms are executed on 17 real data sets and artificial data sets. For MWND algorithm, the parameters are set as follows if without special description: $\alpha = 1.2$, $\beta = 2$, $\tau = 0.6$ and $\lambda = 2$; the parameters of SCIFWSA are set as: $\epsilon = 0.01$; the parameters of FSSCND are set as: $m = 7$, $\rho = 7$ and $\lambda = 0.1$; the parameters of DIFSC are set as: $m = 1.5$, $r = 1.1$, $\alpha = 3.5$, $\beta = 1.2$, $\varepsilon_u = 1 \times 10^{-7}$ and $\varepsilon_w = 0.1$; the parameters of LRGR are set as: $\alpha = 0.3$, $\beta = 1 \times 10^{-8}$ and $\lambda = 0.3$.

In order to evaluate the performance of the algorithms, the following evaluation criteria [14] are used:

1. Rand statistic:

$$R = \frac{SS + DD}{SS + SD + DS + SS} \tag{35}$$

2. Fowlkes and Mallows index:

$$FM = \sqrt{\frac{SS}{SS + SD} \cdot \frac{SS}{SS + DS}} \tag{36}$$

where *SS:* the number of data points that they are in the same cluster and they are also in the same class; *SD:* the number of data points that they are in the same cluster but they are in different classes; *DS:* the number of data points that they are in different clusters but they are in the same class; *DD:* the number of data points that they are in different clusters and they are also in different classes.

3. Purity:

$$P = \frac{1}{N}\sum_{i=1}^{k} \max\left(\left|\boldsymbol{C}_i \cap \boldsymbol{Y}_j\right|\right) \tag{37}$$

where the final clustering result is $\boldsymbol{C} = \{\boldsymbol{C}_1, \boldsymbol{C}_2, \cdots, \boldsymbol{C}_k\}$, $\boldsymbol{Y} = \{\boldsymbol{Y}_1, \boldsymbol{Y}_2, \cdots, \boldsymbol{Y}_n\}$ is the class label of the data set and N is the number of data points of data set.

4. Mean square error:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k} u_{ij}\left\|\boldsymbol{x}_i - \boldsymbol{V}_j\right\|_2 \tag{38}$$

where n is the number of data points of data set, $\boldsymbol{V}_j$ is the clustering center point of the jth cluster. $u_{ij} = 1$ if $\boldsymbol{x}_i$ belongs to the jth cluster; otherwise, $u_{ij} = 0$. For MSE, a small MSE means more compact clusters which is expected result.

4.1 The performance comparison of MWND and the comparison algorithms

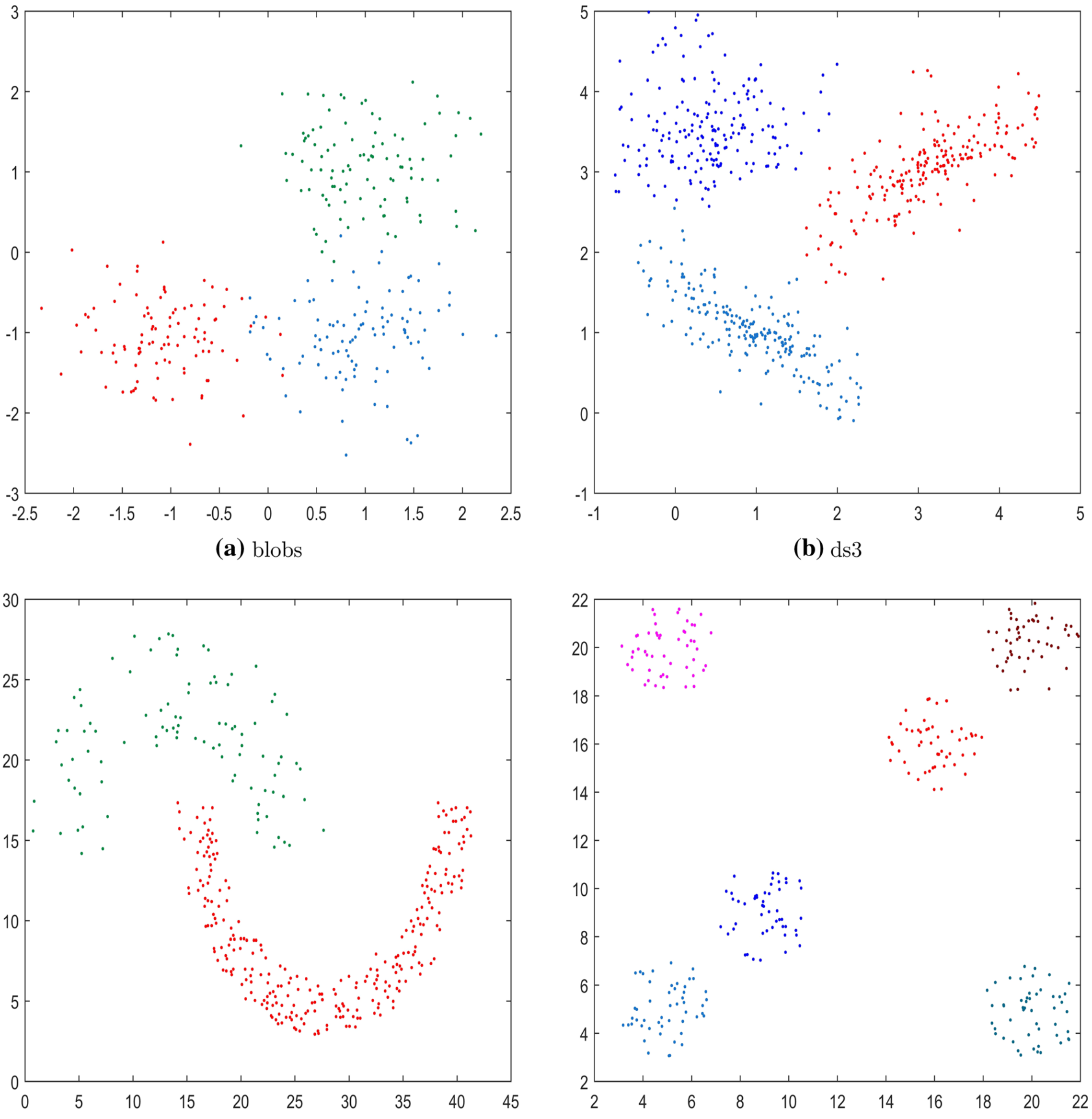
In order to test the performance of MWND, we execute MWND and the comparison algorithms on the synthetic and real-world data sets. The details of the synthetic and real-world data sets are given in Fig. 3 and Table 1, respectively. The test results are shown in Tables 2, 3, 4, 5 and 6 and Fig. 4.

Tables 2, 3, 4 and 5 show the results of MWND and the comparison algorithms on the real data sets. From the results, MWND obtains 7, 7, 5 best results for R, FM and P, respectively; for MSE, MWND only obtains 3 best results which means that FSSCND outperforms MWND; however, it is known that MWND is the second best for MSE. Considering the evaluation criteria comprehensively, it is known that the performance of MWND outperforms the comparison algorithms on the most data sets. Table 6 shows the results of MWND testing on the synthetic data sets. From the results, it is known that the advantage of MWND is obvious and the performance of MWND outperforms the comparison algorithms on the synthetic data sets. Figure 4 shows the data distributions of clustering results of MWND. By comparing the data distributions of Figs. 3 and 4, the clustering result of MWND accords with the real result basically although a few data points are not partitioned into the correct clusters. The above results prove that MWND is effective for clustering tasks. From Table 5, it is known that the MSE of FSSCND is better than MWND. It means that FSSCND can obtain a more compact clustering result. After analyzing Tables 2, 3 and 4, MWND is better than FSSCND although the MSE of FSSCND is better than MWND in Table 5. It can infer that few data points which are far away from centering points are incorrectly partitioned into incorrect clusters and the incorrect partition result increases the MSE of MWND. Therefore, the MSE of MWND is not better than FSSCND.

4.2 The improvement of the performance with the multiple features fusion for MWND

In order to test the improvement of MWND with the multiple features fusion, we choose *glass, wine, zoo, Parkinson's* and *blobs* as the experimental data sets. We execute MWND, MWND-PCA (MWND algorithm with only one feature which is generated by PCA algorithm), MWND-LLE (MWND algorithm with only one feature which is generated by LLE algorithm), MWND-LE (MWND algorithm with only one feature which is generated by LE algorithm) and MWND-NMF (MWND algorithm with only one feature which is generated by NMF algorithm). The parameters are set as: $\lambda = \alpha = \beta = 2$. The test results are shown in Table 7.

Table 7 shows the clustering results of MWND and the comparison algorithms with single feature. From the results, it is known that MWND outperforms the comparison algorithms for R, FM and P on the most data sets. For MSE, MWND obtains 3 best results and MWND-NMF only obtains 2 best results. Therefore, the performance of MWND is the best of the five algorithms. In MWND algorithm, multiple dimension reduction algorithms are used which will generate multiple features. Each feature can be seen as a view to present data points. The fusion of the clustering results of multiple features includes the result information of different views. In addition, the fusion of the clustering results of multiple features is also an ensemble learning. From the theory of ensemble learning, the result performance of multiple learning methods is better than single learning method. Therefore, the



(a) blobs

(b) ds3

(c) jain

(d) spherical

Fig. 3 The data distribution of the synthetic data sets

Table 1 The details of the real-world data sets

Data set	Attributes	Classes	Distribution of classes
Glass	9	2	$\{187, 30\}$
Wine	13	3	$\{59, 71, 48\}$
Zoo	17	7	$\{41, 20, 5, 13, 4, 8, 10\}$
Parkinsons	23	2	$\{48, 147\}$
Adult	14	2	$\{230, 70\}$
Magic	10	2	$\{189, 111\}$
Vehicle	18	2	$\{228, 72\}$
Waveform	21	3	$\{93, 96, 111\}$

performance of MWND is better than the comparison algorithms with single feature on the most data sets. From the above, it can also know that MWND outperforms MWND-PCA, MWND-LLE, MWND-LE and MWND-NMF; in other words, it means that the multi-feature fusion mechanism has improved the performance of MWND.

4.3 The improvement of the performance with the selection of initial center points for MWND

In order to test the improvement of the performance with the selection of initial center points for MWND, we choose *glass, wine, zoo, Parkinson's, blobs* and *spherical* as the

Table 2 The R of the experimental algorithms on the real-world data sets

	SCIFWSA	FSSCND	DIFSC	LRGR	MWND
Glass	0.6137 ± 0.0989	0.6620 ± 0.1785	0.5324 ± 0.0276	0.6852 ± 0.0875	**0.7776** ± **0.1637**
Wine	0.6625 ± 0.0388	0.6923 ± 0.0850	0.6137 ± 0.0093	0.4488 ± 0.0535	**0.7025** ± **0.0977**
Zoo	0.7553 ± 0.1018	0.8315 ± 0.0161	0.6008 ± 0.0362	0.4610 ± 0.1160	**0.8563** ± **0.0687**
Parkinsons	0.5216 ± 0.0244	0.5356 ± 0.0273	0.5634 ± 0.0644	**0.5918** ± **0.054**	0.5802 ± 0.0584
Adult	0.5328 ± 0.0461	0.5768 ± 0.0622	0.5638 ± 0.0197	0.5796 ± 0.0600	**0.5859** ± **0.0417**
Magic	0.5105 ± 0.0085	0.5219 ± 0.0245	0.5053 ± 0.0064	0.5102 ± 0.0139	**0.5351** ± **0.0214**
Vehicle	0.5464 ± 0.0288	0.5247 ± 0.0291	0.5216 ± 0.0124	0.5742 ± 0.0558	**0.5871** ± **0.0429**
Waveform	0.5765 ± 0.0301	0.6180 ± 0.0485	0.4893 ± 0.0183	0.4619 ± 0.0809	**0.6307** ± **0.0357**

Table 3 The FM of the experimental algorithms on the real-world data sets

	SCIFWSA	FSSCND	DIFSC	LRGR	MWND
Glass	0.7255 ± 0.0858	0.7550 ± 0.1398	0.6651 ± 0.0330	0.7149 ± 0.0462	**0.8515** ± **0.1241**
Wine	0.5349 ± 0.0504	0.6214 ± 0.0806	0.4909 ± 0.0253	0.4757 ± 0.0574	**0.6604** ± **0.0713**
Zoo	0.5590 ± 0.1305	0.6179 ± 0.0431	0.4409 ± 0.0297	0.3705 ± 0.0627	**0.7034** ± **0.1443**
Parkinsons	0.5985 ± 0.0227	0.6286 ± 0.0414	0.6561 ± 0.0222	**0.7272** ± **0.0981**	0.7019 ± 0.0838
Adult	0.6333 ± 0.0715	0.6969 ± 0.0887	0.6966 ± 0.0040	0.7038 ± 0.0946	**0.7067** ± **0.0546**
Magic	0.5483 ± 0.0247	0.5842 ± 0.0576	0.5547 ± 0.0107	0.6227 ± 0.0625	**0.6265** ± **0.0757**
Vehicle	0.6342 ± 0.0510	0.6059 ± 0.0269	0.6048 ± 0.0012	0.6967 ± 0.0946	**0.7065** ± **0.0671**
Waveform	0.3869 ± 0.0415	0.4935 ± 0.0567	0.4194 ± 0.0225	0.4458 ± 0.0869	**0.4979** ± **0.0402**

Table 4 The P of the experimental algorithms on the real-world data sets

	SCIFWSA	FSSCND	DIFSC	LRGR	MWND
Glass	0.8749 ± 0.0348	0.8828 ± 0.0384	0.8618 ± 0.0000	0.8618 ± 0.0000	**0.8894** ± **0.0357**
Wine	0.5329 ± 0.0667	0.6372 ± 0.0494	0.5971 ± 0.009	0.4069 ± 0.0071	**0.6541** ± **0.1416**
Zoo	0.6832 ± 0.0906	0.7694 ± 0.0946	0.4738 ± 0.0277	0.4526 ± 0.0227	**0.7893** ± **0.0720**
Parkinsons	0.7538 ± 0.0000	0.7538 ± 0.0000	**0.7560** ± **0.0027**	0.7546 ± 0.0019	0.7538 ± 0.0000
Adult	0.7667 ± 0.0000	**0.7705** ± **0.0101**	0.7667 ± 0.0000	0.7676 ± 0.0016	0.7667 ± 0.0000
Magic	0.6495 ± 0.0076	0.6510 ± 0.0099	0.6467 ± 0.0000	0.6476 ± 0.0016	**0.6562** ± **0.0124**
Vehicle	0.7300 ± 0.0000	0.7300 ± 0.0000	0.7300 ± 0.0000	**0.7305** ± **0.0013**	0.7300 ± 0.0000
Waveform	0.4933 ± 0.0589	3.9971 ± 0.1887	0.3781 ± 0.0033	0.3790 ± 0.0090	**0.5367** ± **0.0788**

experimental data sets. We execute MWND and the MWND algorithm without the mechanism of selecting initial center points (which is denoted as UMWND). The parameters are set as: $\lambda = \alpha = \beta = 1.5$. The test results are shown in Fig. 5.

Figure 5 shows the results of R, FM and P of UMWND and MWND testing on the experimental data sets. From the results, it can be seen that MWND is better than UMWND for those evaluation criteria. Table 8 shows the MSE results of UMWND and MWND testing on the experimental data sets. It is known that MWND obtains 4 best results and UMWND only obtains 2 best results; therefore, the MSE of MWND is less than that of UMWND on the most data sets. From the above analyses, it is known that MWND outperforms UMWND and the selection of initial center points can improve the performance of MWND. In MWND algorithm, initial cluster center data points can be selected to improve the performance of MWND algorithm and speed up the convergence speed. In UMWND algorithm which has no the mechanism of the selection of initial cluster center data points, it will spend more time on the adjustment of cluster center data points and get trapped in a local optimum with a larger possibility than MWND algorithm. Therefore, it is necessary to introduce the selection of initial cluster center data points.

Table 5 The MSE of the experimental algorithms on the real-world data sets

	SCIFWSA	FSSCND	DIFSC	LRGR	MWND
Glass	2.4106 ± 0.0642	**2.2913 ± 0.1053**	2.3348 ± 0.0011	2.4984 ± 0.0050	2.3184 ± 0.1402
Wine	3.0656 ± 0.1410	2.9101 ± 0.2131	3.1580 ± 0.0568	3.4762 ± 0.0201	**2.9032 ± 0.1476**
Zoo	2.8444 ± 0.2662	**2.4850 ± 0.2361**	3.4077 ± 0.1202	3.8073 ± 0.0562	2.5377 ± 0.1823
Parkinson's	3.7484 ± 0.1530	**3.4808 ± 0.0596**	3.6261 ± 0.0516	4.0643 ± 0.0085	3.8774 ± 0.1287
Adult	3.3602 ± 0.0227	3.2936 ± 0.0220	3.3244 ± 0.0087	3.3919 ± 0.0071	**3.2908 ± 0.0511**
Magic	2.6356 ± 0.0840	**2.4782 ± 0.1159**	2.6122 ± 0.0657	2.7893 ± 0.0107	2.5449 ± 0.1411
Vehicle	3.5352 ± 0.3679	**3.2167 ± 0.1930**	3.5395 ± 0.1136	3.9266 ± 0.0059	3.5926 ± 0.3367
Waveform	4.2429 ± 0.1461	3.9459 ± 0.1428	4.4153 ± 0.0041	4.4811 ± 0.0091	**3.8457 ± 0.2206**

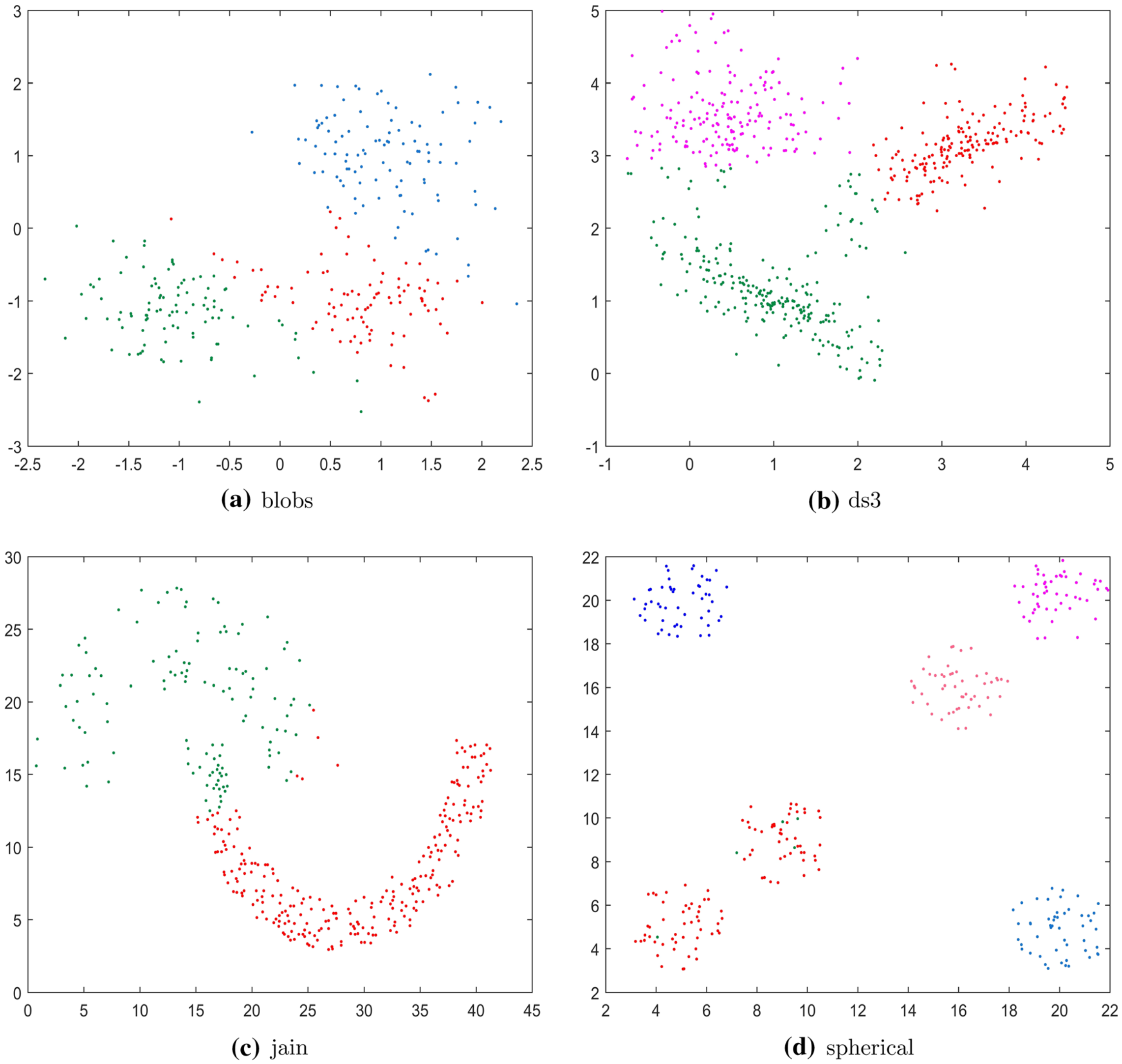
Table 6 The test result of the experimental algorithms on the synthetic data sets

	SCIFWSA	FSSCND	DIFSC	LRGR	MWND
Blobs					
R	0.8037 ± 0.0921	0.7435 ± 0.0657	0.6574 ± 0.0336	0.4019 ± 0.0692	**0.8964 ± 0.0792**
FM	0.7306 ± 0.1111	0.6800 ± 0.0597	0.5436 ± 0.0671	0.5298 ± 0.0388	**0.8616 ± 0.0899**
P	0.9143 ± 0.1201	0.8952 ± 0.1122	0.6314 ± 0.0419	0.7205 ± 0.1995	**0.9267 ± 0.1176**
MSE	0.7502 ± 0.1233	0.7954 ± 0.1079	0.9390 ± 0.0401	1.2968 ± 0.0245	**0.6293 ± 0.1196**
ds3					
R	0.8196 ± 0.0873	0.7597 ± 0.0908	0.6970 ± 0.0468	0.3380 ± 0.0038	**0.8677 ± 0.0942**
FM	0.7503 ± 0.1148	0.6798 ± 0.1075	0.5993 ± 0.0679	0.5723 ± 0.0037	**0.8269 ± 0.1045**
P	0.9133 ± 0.0984	0.8693 ± 0.1612	0.6620 ± 0.0344	0.6472 ± 0.3093	**0.9249 ± 0.1326**
MSE	0.7496 ± 0.1644	0.8076 ± 0.1501	1.0002 ± 0.0683	1.3422 ± 0.0056	**0.6482 ± 0.1387**
Jain					
R	0.6090 ± 0.1094	0.6370 ± 0.1368	0.6207 ± 0.1224	0.6159 ± 0.0292	**0.7578 ± 0.1719**
FM	0.6698 ± 0.0886	0.6941 ± 0.1141	0.6657 ± 0.1120	0.7738 ± 0.0144	**0.7900 ± 0.1525**
P	0.7786 ± 0.0533	0.7932 ± 0.0761	0.7694 ± 0.0780	0.7488 ± 0.0221	**0.8587 ± 0.1062**
MSE	1.0455 ± 0.0937	1.0591 ± 0.0976	0.9663 ± 0.0015	1.3079 ± 0.0252	**0.9382 ± 0.0816**
Spherical					
R	0.8772 ± 0.0356	0.8294 ± 0.0606	0.8009 ± 0.0264	0.3098 ± 0.1177	**0.9236 ± 0.0245**
FM	0.7107 ± 0.0502	0.6127 ± 0.0847	0.5433 ± 0.0396	0.3856 ± 0.0152	**0.8133 ± 0.0483**
P	0.7238 ± 0.0649	0.6619 ± 0.0927	0.5586 ± 0.0331	0.241 ± 0.0669	**0.7990 ± 0.0638**
MSE	0.4701 ± 0.1200	0.5882 ± 0.1521	0.9627 ± 0.0346	1.2902 ± 0.0689	**0.3091 ± 0.0534**

4.4 The impacts of the parameters α and β on the performance of MWND

In order to study the impacts of the parameters on the performance of MWND, we choose *glass* as the experimental data set. $\alpha, \beta, \lambda \in [0, 10]$. Because MSE is far larger than 1, the MSE value of the curve is $\frac{1}{3} \cdot MSE$. We test MWND with different parameters values and the test results are shown in Fig. 6.

From the changed trend of the curves, it is known that the performance of MWND changes with the changes of the parameters. The value of α is the factor of the weight entropy of subspace feature. When $\alpha = 0.5$, the performance of MWND is the best. β is the factor of the global feature which is also the same in different clusters. From Fig. 6b, it is known that the performance of MWND is the best when $\beta = 6.5$. λ can influence the density values of data points when selecting the clustering center points. When $\lambda = 0.5$ or 5.5, the performance of MWND is the best of all. In addition, it can be seen that P and FM are more sensitive to the parameters and the values of the evaluation criteria have large fluctuations if the values of the parameters are changed; by contrast, the fluctuation of the values of P and MSE is smaller than that of P and FM which means that P and FM are not too sensitive to the parameters. In Fig. 6, α and β have an influence on the performance of MWND and the two parameters balance the weight entropy of two weights. From [26], it is known



(a) blobs

(b) ds3

(c) jain

(d) spherical

Fig. 4 The clustering result of MWND on the synthetic data sets

that a small weight entropy can improve the performance of an algorithm. If $\alpha = 0$ or $\beta = 0$, Eq. (19) becomes a subspace clustering algorithm without subspace weight entropy or global weight entropy. If $\alpha \rightarrow +\infty$ or $\beta \rightarrow +\infty$, w_{jr} and ϖ_r become constant values from Eq. (23) and Eq. (27) which is equal to clustering algorithm without weighting mechanism and Eq. (19) becomes a common clustering algorithm without weighting mechanism. Therefore, it is important to select appropriate parameters for MWND algorithm.

4.5 The test results of MWND and the comparison algorithms on the other real data sets

In order to further test the performance of MWND, we execute MWND and the comparison algorithms on *car*, *Statlog*, *Agrawal*, *yeast* and *Wilt* data sets. The details of the data sets are given in Table 9, and the test results are shown in Tables 10, 11, 12, 13, 14 and 15.

Tables 10, 11, 12, 13 and 14 show the test results of the four algorithms. On *car* data set, MWND obtains the best results for FM and P; the R and MSE of MWND are the second best. On *Statlog* data set, the P of MWND is the best of all; the R, FM and MSE of MWND are the second best. On *Agrawal* data set, MWND obtains the best results for R, FM and P; the MSE of MWND is the second best. On *yeast* data set, the FM, P and MSE of MWND are the second best; R is the third best. On *Wilt* data sets, MWND obtains the best results except for MSE. Table 15 shows the average results of the four algorithms on the five data sets. From the test results in Table 15, it is known that the overall performance of MWND is the best of all although the performance of MWND is not the best on some data

Table 7 The test result of the experimental algorithms on the data sets

	MWND-PCA	MWND-LLE	MWND-LE	MWND-NMF	MWND
Glass					
R	0.7303 ± 0.0863	0.7311 ± 0.0507	0.7395 ± 0.1305	0.7290 ± 0.1315	**0.7514** ± **0.1346**
FM	0.8333 ± 0.0731	**0.8476** ± **0.0424**	0.8296 ± 0.1129	0.8295 ± 0.0910	0.8384 ± 0.0901
P	0.8631 ± 0.0022	0.8618 ± 0.0000	0.8841 ± 0.0281	0.8841 ± 0.0386	**0.8907** ± **0.0479**
MSE	2.3878 ± 0.1543	2.4611 ± 0.0574	2.3955 ± 0.0908	2.3389 ± 0.1293	**2.2850** ± **0.1099**
Wine					
R	0.5405 ± 0.2630	0.4650 ± 0.0954	0.6363 ± 0.2300	0.7298 ± 0.0830	**0.7381** ± **0.0877**
FM	0.6677 ± 0.1324	0.5466 ± 0.0399	0.6681 ± 0.1268	0.6724 ± 0.0602	**0.6834** ± **0.0859**
P	0.5562 ± 0.2213	0.4815 ± 0.0572	0.6453 ± 0.2138	0.7063 ± 0.1295	**0.7143** ± **0.1210**
MSE	3.2420 ± 0.3858	3.3819 ± 0.0878	3.0400 ± 0.4072	**2.8364** ± **0.1701**	2.8545 ± 0.1748
Zoo					
R	0.5257 ± 0.2180	0.6272 ± 0.1693	0.5618 ± 0.2976	0.7182 ± 0.1351	**0.8569** ± **0.0427**
FM	0.5546 ± 0.0901	0.5189 ± 0.1349	0.6001 ± 0.1530	0.5882 ± 0.1361	**0.6910** ± **0.0945**
P	0.5516 ± 0.1163	0.5870 ± 0.1195	0.5658 ± 0.1686	0.6365 ± 0.1245	**0.7893** ± **0.0532**
MSE	3.4481 ± 0.3590	3.2238 ± 0.3757	3.3116 ± 0.6187	3.0737 ± 0.4422	**2.4689** ± **0.1534**
Waveform					
p	0.4071 ± 0.1266	0.4293 ± 0.0901	0.5901 ± 0.1185	0.5766 ± 0.1173	**0.6097** ± **0.0628**
FM	**0.5664** ± **0.0220**	0.5149 ± 0.0603	0.5281 ± 0.0304	0.4994 ± 0.0429	0.5131 ± 0.0294
P	0.4186 ± 0.0835	0.4067 ± 0.0369	0.5305 ± 0.0822	0.4981 ± 0.0657	**0.5490** ± **0.0310**
MSE	4.3738 ± 0.2240	4.3272 ± 0.1685	3.9451 ± 0.3756	**3.8856** ± **0.3262**	3.9233 ± 0.3025
Blobs					
R	0.3928 ± 0.1466	0.6646 ± 0.2372	0.5569 ± 0.2985	0.6744 ± 0.1507	**0.8395** ± **0.0860**
FM	0.5901 ± 0.0513	0.7029 ± 0.1079	0.6868 ± 0.1710	0.6563 ± 0.0580	**0.7892** ± **0.1055**
P	0.3862 ± 0.1170	0.6424 ± 0.2358	0.5557 ± 0.3000	0.6333 ± 0.1367	**0.8257** ± **0.1151**
MSE	1.2506 ± 0.1643	0.9235 ± 0.2988	1.0473 ± 0.3675	0.9026 ± 0.1908	**0.6911** ± **0.1099**

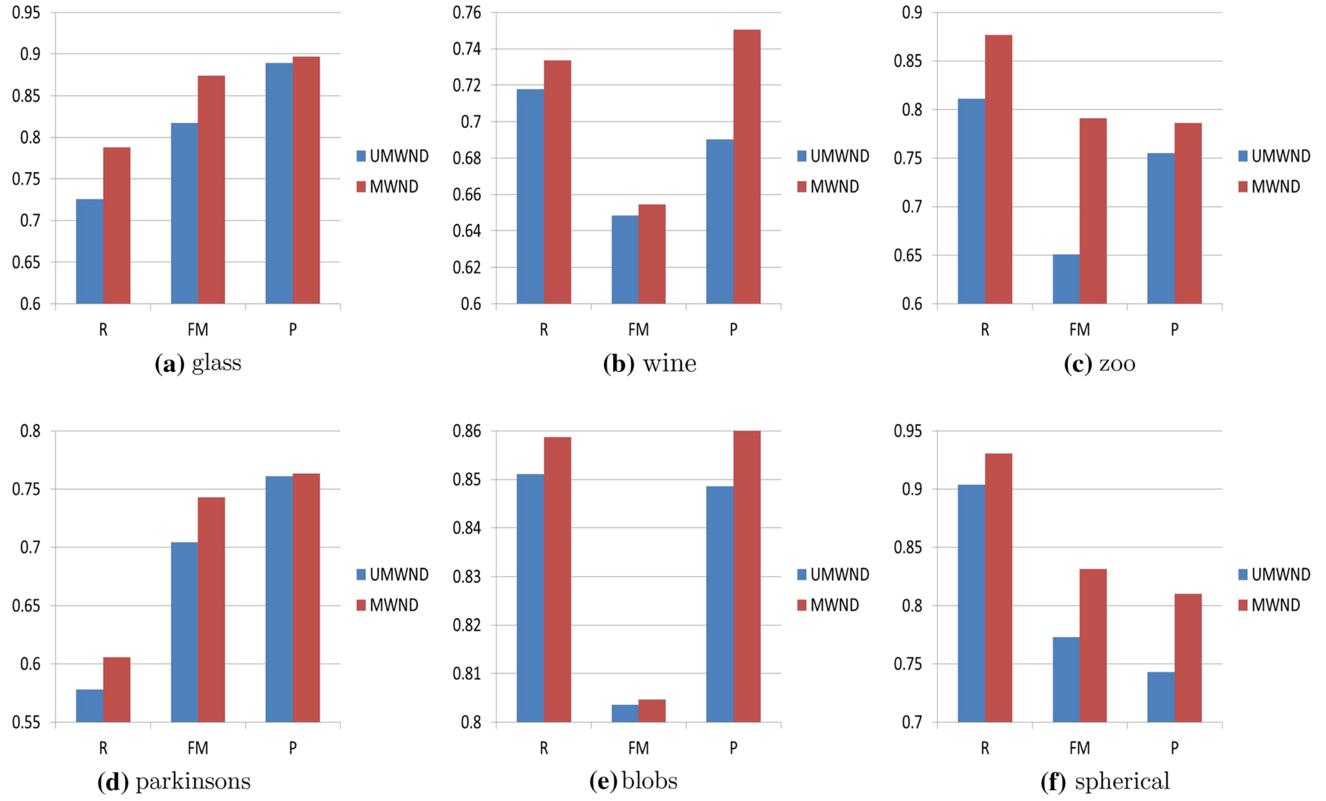
sets. Therefore, it is known that MWND is effective for clustering task. From the results, it can be seen that the performance of MWND is not better than FSSCND on *Statlog* and *yeast* data sets and there are more classes on the two data sets which means FSSCND outperforms MWND on data sets with too many classes. If the classes of data set are too many, the proportion of sample for each class will become small. In FSSCND algorithm, it introduces a noise detection mechanism and data points with too small proportion will be seen as noise which improves the robustness of FSSCND algorithm. Therefore, FSSCND outperforms MWND on the data sets with many classes.

4.6 The performance of MWND evaluated by the other evaluation criteria

In order to future evaluate the performance of MWND algorithm, we choose *glass* and *wine* as experimental data sets and MWND algorithm and the comparison algorithms are executed on the data sets. Kulczynski index (K) and Czekanowski-Dice index (CD) are used to evaluate the performance of MWND algorithm. The test results are shown in Figs. 7 and 8.

$$CD = \frac{2 \cdot TP}{2 \cdot TP + FN + FP} \quad \text{and} \quad K = \frac{1}{2}\left(\frac{TP}{TP + FP} + \frac{TP}{TP + FN}\right) \tag{39}$$

Figures 7 and 8 show the test results of MWND algorithm and the comparison algorithms on *glass* and *wine* data sets. From Figs. 7 and 8, the results of MWND algorithm are better than the other algorithms. The K and CD of MWND algorithm are the best of all algorithms. From the results, it is also known that the difference between MWND algorithm and LRGR algorithm is small. In LRGR algorithm, a low-rank representation with graph regularization is introduced; in other words, LRGR algorithm has a dimension reduction mechanism and a sparse constraint to get a robust ability; therefore, the K and CD of LRGR algorithm are satisfied. However, it has been proved that ensemble learning has a better learning ability than single learning method. Therefore, the results of MWND algorithm are better than LRGR algorithm.



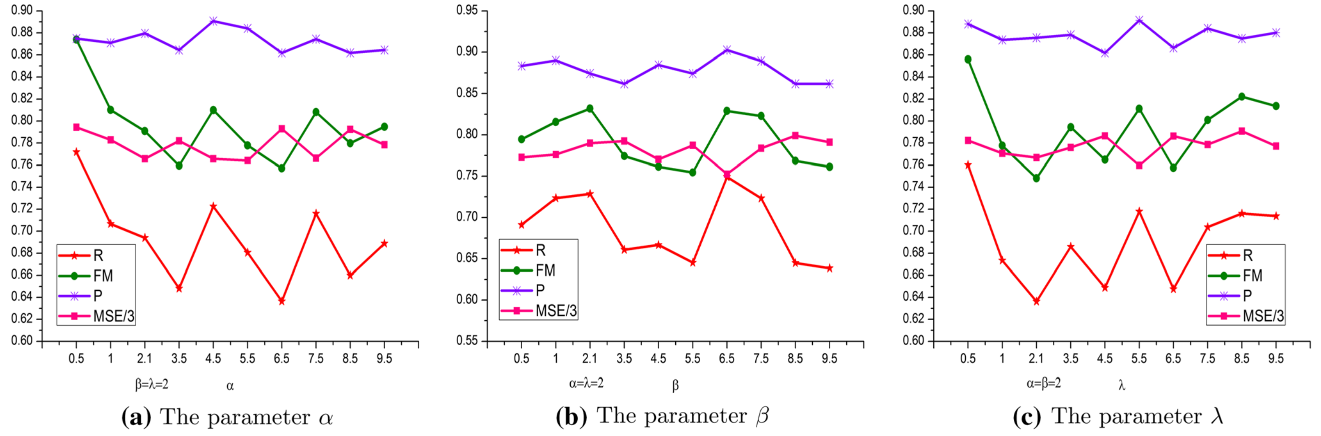
(a) glass **(b)** wine **(c)** zoo

(d) parkinsons **(e)** blobs **(f)** spherical

Fig. 5 The test result of UMWND and MWND on the experimental data sets

Table 8 The MSE of MWND and UMWND on the different data sets

	Glass	Wine	Zoo	Parkinson's	Blobs	Spherical
UMWND	**2.2898 ± 0.0762**	2.9162 ± 0.1923	2.6490 ± 0.3525	**3.8754 ± 0.1500**	0.6664 ± 0.1124	0.3338 ± 0.0535
MWND	2.3318 ± 0.1157	**2.7992 ± 0.2188**	**2.5641 ± 0.2964**	3.8866 ± 0.0937	**0.6617 ± 0.0881**	**0.2942 ± 0.0371**



(a) The parameter α **(b)** The parameter β **(c)** The parameter λ

Fig. 6 The impact of the parameters values on the performance of MWND

Table 9 The details of the real-world data sets

Data set	Attributes	Samples	Classes	Distribution of classes
Car	7	1728	4	$\{384, 1210, 65, 69\}$
Statlog	20	2310	7	$\{330, 330, 330, 330, 330, 330, 330\}$
Agrawal	10	2000	2	$\{682, 1318\}$
Yeast	9	1484	10	$\{244, 51, 429, 463, 163, 20, 44, 35, 30, 5\}$
Wilt	6	4339	2	$\{74, 4265\}$

Table 10 The test result of the experimental algorithms on *car* data set

	R	FM	P	MSE
SCIFWSA	0.5052 ± 0.0242	0.4359 ± 0.0323	0.7105 ± 0.0207	2.2858 ± 0.0609
FSSCND	0.4915 ± 0.0131	0.4107 ± 0.0203	0.7008 ± 0.0015	**2.1036 ± 0.0182**
DIFSC	**0.5367 ± 0.0000**	0.4848 ± 1.2e−16	0.7002 ± 1.1e−16	2.1972 ± 0.0000
MWND	0.5281 ± 0.0339	**0.4891 ± 0.0460**	**0.7277 ± 0.0259**	2.1304 ± 0.0648

Table 11 The test result of the experimental algorithms on *Statlog* data set

	R	FM	P	MSE
SCIFWSA	0.6800 ± 0.1128	0.2952 ± 0.0321	0.3539 ± 0.0368	3.2085 ± 0.1397
FSSCND	**0.7859 ± 0.0588**	**0.4498 ± 0.0650**	0.4213 ± 0.0766	**2.4176 ± 0.1811**
DIFSC	0.6401 ± 0.1016	0.4078 ± 0.0131	0.4025 ± 0.0490	3.2178 ± 0.0653
MWND	0.6898 ± 0.0895	0.4346 ± 0.0346	**0.4251 ± 0.0756**	2.8873 ± 0.2240

Table 12 The test result of the experimental algorithms on *Agrawal* data set

	R	FM	P	MSE
SCIFWSA	0.5098 ± 0.0136	0.5415 ± 0.0170	0.6651 ± 0.0047	2.8761 ± 0.0282
FSSCND	0.5103 ± 0.0114	0.5793 ± 0.0509	0.6590 ± 1.2e−16	**2.7780 ± 0.0627**
DIFSC	0.5057 ± 0.0009	0.5471 ± 0.0019	0.6533 ± 0.0000	2.7943 ± 7.0e−05
MWND	**0.5306 ± 0.0153**	**0.6590 ± 0.0637**	**0.6590 ± 1.0e−16**	2.7911 ± 0.0646

Table 13 The test result of the experimental algorithms on *yeast* data set

	R	FM	P	MSE
SCIFWSA	0.6449 ± 0.1169	0.2657 ± 0.0622	0.3668 ± 0.0267	2.2223 ± 0.0694
FSSCND	**0.7082 ± 0.0208**	0.2508 ± 0.0211	**0.4447 ± 0.0319**	**1.8205 ± 0.0399**
DIFSC	0.5558 ± 0.0914	0.3043 ± 0.0401	0.3703 ± 0.0164	2.2336 ± 0.0155
MWND	0.6407 ± 0.0269	**0.3206 ± 0.0449**	0.4157 ± 0.0331	1.8452 ± 0.0502

Table 14 The test result of the experimental algorithms on *Wilt* data set

	R	FM	P	MSE
SCIFWSA	0.6378 ± 0.1367	0.7870 ± 0.0886	**0.9830 ± 1.1e−16**	1.7216 ± 0.0567
FSSCND	0.6161 ± 0.1542	0.7718 ± 0.0963	**0.9830 ± 9.1e−17**	**1.6497 ± 0.0308**
DIFSC	0.6797 ± 0.1419	0.8138 ± 0.0918	**0.9830 ± 1.1e−16**	1.7172 ± 0.0219
MWND	**0.8244 ± 0.2092**	**0.8983 ± 0.1284**	**0.9830 ± 6.4e−17**	1.7528 ± 0.0628

4.7 The impact of the parameter τ on the performance of MWND

In order to test the impact of the parameters τ on the performance of MWND algorithm, we choose *glass* as experimental data set. MWND algorithm is executed on the data set with different τ values. The test results are shown in Figs. 9 and 10.

Figure 9 shows the results of MWND algorithm with different τ values. τ can determine the neighborhood radius of Eq. (16) which can in future determine the density of each data point. Figure 10 shows the time cost of MWND

Table 15 The average result of the experimental algorithms on *car, Statlog, Agrawal, yeast* and *Wilt* data sets

	R	FM	P	MSE
SCIFWSA	0.5955 ± 0.0820	0.4651 ± 0.2115	0.6159 ± 0.2631	2.4629 ± 0.5843
FSSCND	0.6224 ± 0.1263	0.4925 ± 0.1952	0.6418 ± 0.2279	**2.1539 ± 0.4547**
DIFSC	0.5836 ± 0.0733	0.5116 ± 0.1917	0.6219 ± 0.2495	2.4320 ± 0.5818
MWND	**0.6427 ± 0.1234**	**0.5603 ± 0.2249**	**0.6421 ± 0.2357**	2.2814 ± 0.5290

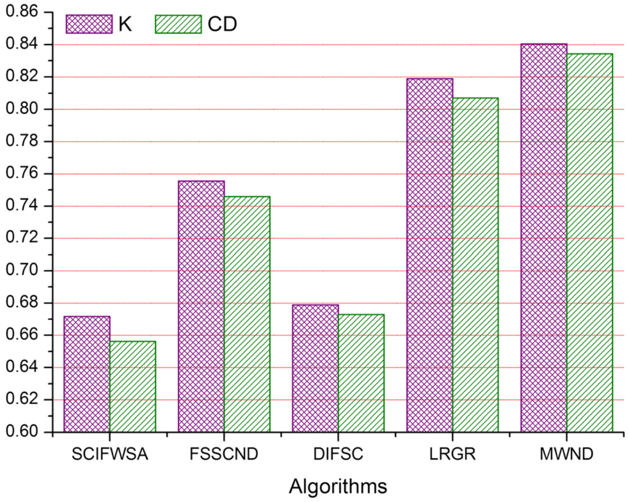


Fig. 7 The K and CD of MWND algorithm on glass data set ($\lambda = 9$, $\alpha = \beta = 2$, $\tau = 0.6$)

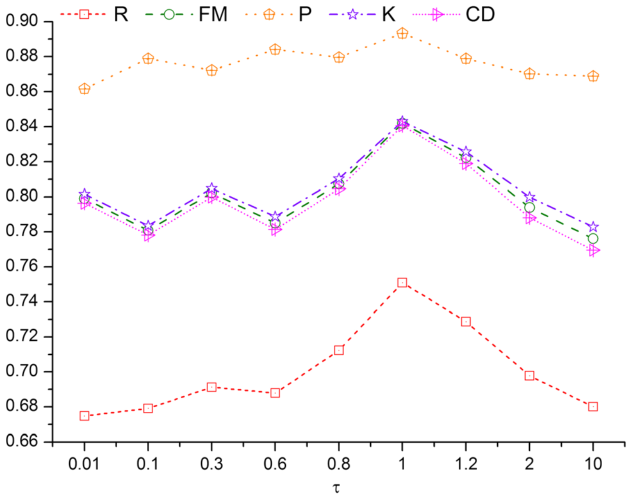


Fig. 9 The result of MWND algorithm on glass data set with different τ values ($\lambda = 5.5$, $\alpha = \beta = 2$)

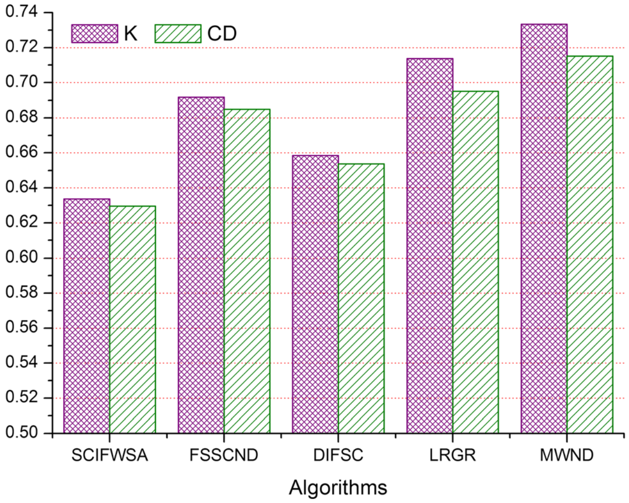


Fig. 8 The K and CD of MWND algorithm on wine data set ($\lambda = 9$, $\alpha = \beta = 2$, $\tau = 0.6$)

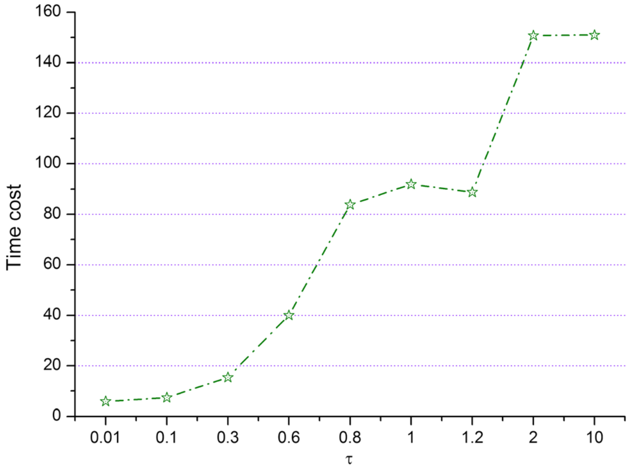


Fig. 10 The time cost of MWND algorithm on glass data set with different τ values ($\lambda = 5.5$, $\alpha = \beta = 2$)

algorithm with different τ values. From Fig. 9, it can be seen that the whole performance of MWND algorithm improves with τ increasing. When τ is 1, the performance of MWND algorithm is the best. If $\tau > 1$, the performance of MWND algorithm decreases with τ increasing. The above results show that $\tau = 1$ is favorable to MWND algorithm on *glass* data set. From Fig. 10, it can be seen that τ can affect the time cost of MWND algorithm. The time cost of $\tau = 2$ is higher than the time cost of $\tau = 0.1$ and it means that the time cost is sensitive to τ. When τ is larger than 2, the time cost is almost unchanged; it can be inferred that the neighborhood radius is too large at this time and the density of each data point is determined by almost all data points under this radius.

Table 16 The test results of MWND algorithm and FSSCND algorithm with different dimensions on zoo data set

	Dimensions	2	4	7	10	12	15
R	MWND	0.7977 ± 0.0343	0.8989 ± 0.0186	0.8383 ± 0.0471	0.8631 ± 0.0544	0.8708 ± 0.0571	0.8692 ± 0.0518
	FSSCND	0.6960 ± 0.1432	0.8305 ± 0.0705	0.7786 ± 0.1089	0.8600 ± 0.0901	0.8635 ± 0.0779	0.8080 ± 0.0458
FM	MWND	0.5636 ± 0.0811	0.8132 ± 0.0221	0.6469 ± 0.1149	0.7234 ± 0.1017	0.7522 ± 0.0883	0.7137 ± 0.1186
	FSSCND	0.6595 ± 0.0810	0.7635 ± 0.0698	0.5857 ± 0.1389	0.6961 ± 0.1717	0.7022 ± 0.1759	0.5943 ± 0.1016
P	MWND	0.7341 ± 0.0176	0.7794 ± 0.0169	0.7327 ± 0.0274	0.8020 ± 0.0580	0.7850 ± 0.0588	0.7963 ± 0.0574
	FSSCND	0.6478 ± 0.0613	0.7016 ± 0.0558	0.6959 ± 0.0814	0.8105 ± 0.0839	0.7680 ± 0.0790	0.7185 ± 0.0634
K	MWND	0.5662 ± 0.0805	0.8189 ± 0.0198	0.6474 ± 0.1147	0.7287 ± 0.0995	0.7585 ± 0.0846	0.7166 ± 0.1174
	FSSCND	0.7102 ± 0.0481	0.7903 ± 0.0558	0.5951 ± 0.1349	0.7004 ± 0.1700	0.7026 ± 0.1757	0.5962 ± 0.1021
CD	MWND	0.5610 ± 0.0818	0.8075 ± 0.0252	0.6465 ± 0.1151	0.7182 ± 0.1040	0.7461 ± 0.0922	0.7108 ± 0.1198
	FSSCND	0.6141 ± 0.1098	0.7379 ± 0.0827	0.5772 ± 0.1449	0.6919 ± 0.1734	0.7017 ± 0.1761	0.5924 ± 0.1011
Time	MWND	16.0401	12.5638	10.0922	8.8854	10.7418	9.1552
	FSSCND	0.6963	0.5719	0.6216	0.6059	1.1096	0.6914

Table 17 The test results of MWND algorithm and FSSCND algorithm with different numbers of samples on Statlog data set

	Samples	100	300	600	900	1200	1500
R	MWND	0.7516 ± 0.0614	0.7413 ± 0.1005	0.6293 ± 0.1826	0.6936 ± 0.0867	0.6327 ± 0.1033	0.6506 ± 0.0944
	FSSCND	0.7935 ± 0.0383	0.7977 ± 0.0297	0.7843 ± 0.0421	0.7285 ± 0.1861	0.8065 ± 0.0354	0.8076 ± 0.0516
FM	MWND	0.4336 ± 0.0862	0.4563 ± 0.0562	0.4172 ± 0.0658	0.4278 ± 0.0363	0.4240 ± 0.0616	0.4354 ± 0.0337
	FSSCND	0.4201 ± 0.0661	0.4520 ± 0.0841	0.4639 ± 0.0719	0.4316 ± 0.0650	0.4637 ± 0.0743	0.4824 ± 0.0887
P	MWND	0.5186 ± 0.0841	0.4933 ± 0.0712	0.3988 ± 0.1234	0.4679 ± 0.0434	0.4292 ± 0.0874	0.4391 ± 0.0586
	FSSCND	0.5314 ± 0.0720	0.5324 ± 0.0821	0.5183 ± 0.0614	0.4971 ± 0.1381	0.5630 ± 0.0866	0.5483 ± 0.0878
K	MWND	0.4596 ± 0.0858	0.4879 ± 0.0468	0.4762 ± 0.0525	0.4726 ± 0.0383	0.4889 ± 0.0601	0.4983 ± 0.0301
	FSSCND	0.4294 ± 0.0651	0.4632 ± 0.0870	0.4822 ± 0.0735	0.4594 ± 0.0491	0.4728 ± 0.0731	0.4941 ± 0.0860
CD	MWND	0.4096 ± 0.0889	0.4281 ± 0.0701	0.3690 ± 0.0896	0.3884 ± 0.0465	0.3688 ± 0.0680	0.3818 ± 0.0482
	FSSCND	0.4111 ± 0.0675	0.4410 ± 0.0816	0.4465 ± 0.0717	0.4091 ± 0.0894	0.4549 ± 0.0756	0.4713 ± 0.0920
Time	MWND	10.5134	89.0620	639.0142	2203.7092	5736.3729	10479.9021
	FSSCND	0.78026	1.9287	4.4487	6.2568	9.1736	11.9147

4.8 The scalability of MWND algorithm

In order to test the scalability of MWND algorithm, we choose zoo and Statlog as the experimental data sets and test MWND algorithm and FSSCND algorithm with different dimensions and numbers of samples. The test results are shown in Tables 16 and 17.

Table 16 shows the test results of MWND algorithm and FSSCND algorithm with different dimensions on zoo data set. From the results, it can be seen that the test results are different with different dimensions. It means the dimension of data set has an influence on the performance of the algorithms. Table 17 shows the test results of MWND algorithm and FSSCND algorithm with different numbers of samples on Statlog data set. From the results, it is known that the performance change of the two algorithms is not significant for the most evaluation criteria. Therefore, the number of samples has no obvious influence on the performance of the two algorithms. Tables 16 and 17 also show the time costs of the two algorithm. In Table 16, the changes of the time costs are not significant. It indicates that the time costs of MWND algorithm and FSSCND algorithm are not too sensitive to the dimensions of data set. However, the changes of the time costs are significant in Table 17. It indicates that the time cost of MWND algorithm is more sensitive to the number of samples which is consistent with the time complexity analysis in Sect. 3.

4.9 The robustness of MWND algorithm

In order to test the robustness of MWND algorithm, we choose waveform as the experimental data set and test

Table 18 The test results of MWND algorithm with different noise levels on waveform data set

	Noise dimensions (%)	2	6	10	14	18	20
R	5	0.5138 ± 0.1037	0.5417 ± 0.1127	0.5503 ± 0.0742	0.5716 ± 0.0596	0.5285 ± 0.1419	0.5428 ± 0.0596
	10	0.5760 ± 0.0653	0.5842 ± 0.0934	0.5577 ± 0.1051	0.6167 ± 0.0362	0.4803 ± 0.1002	0.5926 ± 0.0851
	15	0.6012 ± 0.0601	0.5944 ± 0.0549	0.5649 ± 0.0631	0.5352 ± 0.0733	0.5346 ± 0.0838	0.5489 ± 0.0979
	20	0.6024 ± 0.0587	0.6228 ± 0.0412	0.5747 ± 0.0660	0.5444 ± 0.0766	0.5447 ± 0.0790	0.5133 ± 0.0794
FM	5	0.5099 ± 0.0432	0.4934 ± 0.0505	0.4736 ± 0.0342	0.4954 ± 0.0372	0.5047 ± 0.0385	0.5028 ± 0.0423
	10	0.5071 ± 0.0378	0.5002 ± 0.0208	0.4893 ± 0.0343	0.4702 ± 0.0253	0.4918 ± 0.0439	0.5117 ± 0.0607
	15	0.5018 ± 0.0169	0.4958 ± 0.0226	0.4937 ± 0.0172	0.4881 ± 0.0224	0.4683 ± 0.0422	0.4877 ± 0.0222
	20	0.4779 ± 0.0439	0.4998 ± 0.0249	0.5018 ± 0.0255	0.4913 ± 0.0339	0.4782 ± 0.0194	0.4827 ± 0.0474
P	5	0.4776 ± 0.0641	0.5167 ± 0.1047	0.5005 ± 0.0817	0.5171 ± 0.0540	0.4595 ± 0.0767	0.5129 ± 0.0462
	10	0.5319 ± 0.0755	0.5271 ± 0.0599	0.4976 ± 0.0711	0.5148 ± 0.0192	0.4524 ± 0.0946	0.5462 ± 0.0950
	15	0.5319 ± 0.0546	0.5310 ± 0.0251	0.5114 ± 0.0371	0.4881 ± 0.0533	0.4586 ± 0.0657	0.4810 ± 0.0616
	20	0.5300 ± 0.0465	0.5305 ± 0.0141	0.5219 ± 0.0561	0.5038 ± 0.0571	0.5000 ± 0.0542	0.4676 ± 0.0521
K	5	0.5376 ± 0.0623	0.5130 ± 0.0704	0.4861 ± 0.0399	0.5067 ± 0.0402	0.5317 ± 0.0682	0.5217 ± 0.0530
	10	0.5193 ± 0.0378	0.5123 ± 0.0358	0.5043 ± 0.0477	0.4727 ± 0.0276	0.5224 ± 0.0667	0.5231 ± 0.0640
	15	0.5095 ± 0.0154	0.5042 ± 0.0271	0.5057 ± 0.0169	0.5054 ± 0.0399	0.4827 ± 0.0537	0.5038 ± 0.0370
	20	0.4829 ± 0.0469	0.5049 ± 0.0296	0.5136 ± 0.0275	0.5078 ± 0.0427	0.4921 ± 0.0265	0.5044 ± 0.0629
CD	5	0.4844 ± 0.0343	0.4756 ± 0.0416	0.4616 ± 0.0321	0.4844 ± 0.0361	0.4806 ± 0.0237	0.4849 ± 0.0347
	10	0.4952 ± 0.0393	0.4889 ± 0.0174	0.4755 ± 0.0314	0.4677 ± 0.0236	0.4639 ± 0.0290	0.5008 ± 0.0604
	15	0.4942 ± 0.0208	0.4876 ± 0.0206	0.4821 ± 0.0211	0.4719 ± 0.0108	0.4548 ± 0.0366	0.4726 ± 0.0183
	20	0.4731 ± 0.0423	0.4948 ± 0.0212	0.4904 ± 0.0272	0.4757 ± 0.0301	0.4650 ± 0.0197	0.4624 ± 0.0372
Time	5	36.7222	46.7116	38.4313	37.7778	40.0639	38.3396
	10	38.2754	37.4190	39.7201	38.7987	39.0321	39.4250
	15	37.5896	38.6037	39.5947	38.0325	37.9392	42.0718
	20	35.4483	35.9370	39.8502	36.4786	43.557	38.1053

MWND algorithm with different noise levels. The test results are shown in Table 18.

Table 18 shows the results of MWND algorithm with different noise levels and noise dimensions. From the results, it can be seen that the change of the performance is not significant for the most evaluation criteria when the noise level is improved or the noise dimensions are improved. It means the performance of MWND algorithm has a good robustness for noise. In MWND algorithm, it introduces ensemble soft subspace clustering to generate multiple clustering results. It is obvious that the clustering result which is deep affected by noise will have a smaller weight. The final result is the fusion of multiple clustering results with weighting mechanism, and the clustering result affected by noise has a small influence on the final result. Therefore, MWND algorithm has a good robustness for noise.

4.10 Discussions

In the experiments, we execute MWND algorithm and the comparison algorithms on a series of data sets. The performance of the algorithms is evaluated by different evaluation criteria. The test results show that the performance of MWND algorithm can be improved by introducing the selection of initial cluster center data points and the fusion of multiple clustering results which proves that MWND algorithm is an effective algorithm for clustering task. In addition, we study the impacts of the parameters on the performance of MWND algorithm. It shows that a too large value or a too small value will bring a negative impact on the performance of MWND algorithm, and it is important to select appropriate parameters for MWND algorithm. We also execute MWND algorithm and the comparison algorithms on the other real data sets. The test results show that FSSCND outperforms MWND on the data sets with too many classes. We analyze the reason of the results, and it is shown that the noise detection mechanism of FSSCND algorithm improves the robustness of FSSCND algorithm. The reason of FSSCND's good performance also gives us a direction that we can improve the robustness of MWND algorithm to improve the performance of MWND algorithm.

5 Conclusions

In this paper, a new multi-feature weighting neighborhood density clustering algorithm (MWND) is proposed. MWND employs multiple different dimension reduction algorithms to generate multiple features and utilizes neighborhood density to select initial clustering center points which are determined by the upper approximation set and lower approximation set of neighborhood rough set. After each feature generates a clustering result, the final result of MWND is the fusion of multiple clustering results with three weighting mechanisms. In the experiments, MWND and the comparison algorithms are tested on the artificial and real data sets. The experimental results show that the performance of MWND outperforms the comparison algorithms and MWND is effective for clustering task.

However, it is known that there are several predefined parameters which improve the difficulty of adjusting parameters. In addition, the performance of MWND is not good on some irregular data sets from the experimental results. Therefore, how to decrease the dependence on prior knowledge and improve the performance of MWND for irregular data sets will be the future research directions.

Acknowledgements This work was supported by National Key Research and Development Program of China (Nos.2017YFB1300200, 2017YFB1300203), National Natural Science Fund of China (Nos. 61672130, 61602082, 61627808, 91648205), the Open Program of State Key Laboratory of Software Architecture (No. SKLSAOP1701), MOE Research Center for Online Education of China (No. 2016YB121), LiaoNing Revitalization Talents Program (No. XLYC1806006), the Fundamental Research Funds for the Central Universities (Nos. DUT19RC(3)012, DUT17RC(3)071) and the development of science and technology of Guangdong province special fund project (No. 2016B090910001). The authors are grateful to the editor and the anonymous reviewers for constructive comments that helped to improve the quality and presentation of this paper.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. An S, Hu Q, Yu D (2015) Robust rough set and applications. Tsinghua University Press, Beijing
2. Arthur D, Vassilvitskii S (2007) k-means++: the advantages of careful seeding. In: Proceedings of the eighteenth annual ACM-SIAM symposium on discrete algorithms. Society for Industrial and Applied Mathematics, pp 1027–1035
3. Bai L, Liang J, Dang C, Cao F (2013) The impact of cluster representatives on the convergence of the k-modes type clustering. IEEE Trans Pattern Anal Mach Intell 35(6):1509–1522
4. Belkin M, Niyogi P (2002) Laplacian eigenmaps and spectral techniques for embedding and clustering. Advances in neural information processing systems. MIT Press, Cambridge, pp 585–591
5. Berkhin P (2006) A survey of clustering data mining techniques. In: Kogan J, Nicholas C, Teboulle M (eds) Grouping multidimensional data. Springer, Berlin, pp 25–71
6. Bishop CM (2006) Pattern recognition and machine learning (information science and statistics). Springer, New York
7. Boongoen T, Shang C, Iam-On N, Shen Q (2011) Extending data reliability measure to a filter approach for soft subspace clustering. IEEE Trans Syst Man Cybern Part B (Cybernetics) 41(6):1705–1714
8. Celebi ME, Kingravi HA, Vela PA (2013) A comparative study of efficient initialization methods for the k-means clustering algorithm. Expert Syst Appl 40(1):200–210
9. Chen X, Ye Y, Xu X, Huang JZ (2012) A feature group weighting method for subspace clustering of high-dimensional data. Pattern Recogn 45(1):434–446
10. Chitsaz E, Jahromi MZ (2016) A novel soft subspace clustering algorithm with noise detection for high dimensional datasets. Soft Comput 20(11):4463–4472
11. Comaniciu D, Meer P (2002) Mean shift: a robust approach toward feature space analysis. IEEE Trans Pattern Anal Mach Intell 24(5):603–619
12. Cunningham P (2008) Dimension reduction. In: Cord M, Cunningham P (eds) Machine learning techniques for multimedia. Springer, Berlin, pp 91–112
13. Deng Z, Choi KS, Chung FL, Wang S (2010) Enhanced soft subspace clustering integrating within-cluster and between-cluster information. Pattern Recogn 43(3):767–781
14. Deng Z, Choi KS, Jiang Y, Wang J, Wang S (2016) A survey on soft subspace clustering. Inf Sci 348:84–106
15. Erisoglu M, Calis N, Sakallioglu S (2011) A new algorithm for initial cluster centers in k-means algorithm. Pattern Recogn Lett 32(14):1701–1705
16. Fahad A, Alshatri N, Tari Z, Alamri A, Khalil I, Zomaya AY, Foufou S, Bouras A (2014) A survey of clustering algorithms for big data: taxonomy and empirical analysis. IEEE Trans Emerg Top Comput 2(3):267–279
17. Fang C, Gao J, Wang D, Wang D, Wang J (2018) Optimization of stepwise clustering algorithm in backward trajectory analysis. Neural Comput Appl. https://doi.org/10.1007/s00521-018-3782-9
18. García-Escudero LA, Gordaliza A, Matrán C, Mayo-Iscar A (2010) A review of robust clustering methods. Adv Data Anal Classif 4(2–3):89–109
19. Golub GH, Reinsch C (1970) Singular value decomposition and least squares solutions. Numer. Math. 14(5):403–420
20. Guo G, Chen S, Chen L (2012) Soft subspace clustering with an improved feature weight self-adjustment mechanism. Int. J. Mach. Learn. Cybern. 3(1):39–49
21. He W, Chen JX, Zhang W (2017) Low-rank representation with graph regularization for subspace clustering. Soft Comput 21(6):1569–1581
22. Hu Q, Yu D, Liu J, Wu C (2008) Neighborhood rough set based heterogeneous feature subset selection. Inf Sci 178(18):3577–3594
23. Huang X, Ye Y, Guo H, Cai Y, Zhang H, Li Y (2014) Dskmeans: a new kmeans-type approach to discriminative subspace clustering. Knowl Based Syst 70:293–300
24. Huang X, Ye Y, Zhang H (2014) Extensions of kmeans-type algorithms: a new clustering framework by integrating intracluster compactness and intercluster separation. IEEE Trans Neural Netw Learn Syst 25(8):1433–1446
25. Jain AK, Murty MN, Flynn PJ (1999) Data clustering: a review. ACM Comput Surv: CSUR 31(3):264–323
26. Jing L, Ng MK, Huang JZ (2007) An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data. IEEE Trans Knowl Data Eng 19(8):1026–1041

27. Jolliffe I (2011) Principal component analysis. In: Lovric M (ed) International encyclopedia of statistical science. Springer, Berlin, pp 1094–1096
28. Kanungo T, Mount DM, Netanyahu NS, Piatko CD, Silverman R, Wu AY (2002) An efficient k-means clustering algorithm: analysis and implementation. IEEE Trans Pattern Anal Mach Intell 7:881–892
29. Kim Kj, Ahn H (2008) A recommender system using ga k-means clustering in an online shopping market. Expert Syst Appl 34(2):1200–1209
30. Kumar A, Daumé H (2011) A co-training approach for multi-view spectral clustering. In: Proceedings of the 28th international conference on machine learning (ICML-11), pp 393–400
31. Lee DD, Seung HS (1999) Learning the parts of objects by non-negative matrix factorization. Nature 401(6755):788
32. Logesh R, Subramaniyaswamy V, Malathi D, Sivaramakrishnan N, Vijayakumar V (2018) Enhancing recommendation stability of collaborative filtering recommender system through bio-inspired clustering ensemble method. Neural Comput Appl. https://doi.org/10.1007/s00521-018-3891-5
33. MacQueen J et al. (1967) Some methods for classification and analysis of multivariate observations. In: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, vol 1, Oakland, CA, USA, pp 281–297
34. Nataliani Y, Yang MS (2017) Powered Gaussian kernel spectral clustering. Neural Comput Appl. https://doi.org/10.1007/s00521-017-3036-2
35. Qian Y, Liang J, Wu W, Dang C (2011) Information granularity in fuzzy binary GRC model. IEEE Trans Fuzzy Syst 19(2):253–264
36. Ren Y, Domeniconi C, Zhang G, Yu G (2014) A weighted adaptive mean shift clustering algorithm. In: Proceedings of the 2014 SIAM international conference on data mining. SIAM, pp 794–802
37. Roweis ST, Saul LK (2000) Nonlinear dimensionality reduction by locally linear embedding. Science 290(5500):2323–2326
38. Saha S, Das R (2018) Exploring differential evolution and particle swarm optimization to develop some symmetry-based automatic clustering techniques: application to gene clustering. Neural Comput Appl 30(3):735–757. https://doi.org/10.1007/s00521-016-2710-0
39. Schroff F, Kalenichenko D, Philbin J (2015) Facenet: A unified embedding for face recognition and clustering. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 815–823
40. Tassa T, Cohen DJ (2013) Anonymization of centralized and distributed social networks by sequential clustering. IEEE Trans Knowl Data Eng 25(2):311–324
41. Tenenbaum JB, De Silva V, Langford JC (2000) A global geometric framework for nonlinear dimensionality reduction. Science 290(5500):2319–2323
42. Wang J, Fl Chung, Wang S, Deng Z (2014) Double indices-induced fcm clustering and its integration with fuzzy subspace clustering. Pattern Anal Appl 17(3):549–566
43. Wang Y, Ru Y, Chai J (2018) Time series clustering based on sparse subspace clustering algorithm and its application to daily box-office data analysis. Neural Comput Appl. https://doi.org/10.1007/s00521-018-3731-7
44. Wu WZ, Leung Y, Mi JS (2009) Granular computing and knowledge reduction in formal contexts. IEEE Trans Knowl Data Eng 21(10):1461–1474
45. Xia H, Zhuang J, Yu D (2013) Novel soft subspace clustering with multi-objective evolutionary approach for high-dimensional data. Pattern Recogn 46(9):2562–2575
46. Zhang H, Chow TW, Wu QJ (2016) Organizing books and authors by multilayer som. IEEE Trans Neural Netw Learn Syst 27(12):2537–2550
47. Zhang H, Wu QJ, Chow TW, Zhao M (2012) A two-dimensional neighborhood preserving projection for appearance-based face recognition. Pattern Recogn 45(5):1866–1876
48. Zhang X (2017) Data clustering. Science Press, Beijing
49. Zhang X, Jing L, Hu X, Ng M, Jiangxi JX, Zhou X (2008) Medical document clustering using ontology-based term similarity measures. Int J Data Warehous Min: IJDWM 4(1):62–73
50. Zhao M, Zhang H, Cheng W, Zhang Z (2016) Joint l p-and l 2, p-norm minimization for subspace clustering with outlier pursuit. In: 2016 international joint conference on neural networks (IJCNN). IEEE, pp 3658–3665
51. Zhou ZH (2012) Ensemble methods: foundations and algorithms. Chapman and Hall, London
52. Zong L, Zhang X, Zhao L, Yu H, Zhao Q (2017) Multi-view clustering via multi-manifold regularized non-negative matrix factorization. Neural Netw 88:74–89