JOURNAL OF INFORMATION AND COMMUNICATION TECHNOLOGY
https://e-journal.uum.edu.my/index.php/jict

How to cite this article:
Hengda, W., Mohamad Mohsin, M. F., & Mohd Pozi, M. S. (2024). Beta distribution weighted fuzzy C-ordered-means clustering. *Journal of Information and Communication Technology, 23*(3), 523-559. https://doi.org/10.32890/jict2024.23.3.6

Beta Distribution Weighted Fuzzy C-Ordered-Means Clustering

*[1]Wang Hengda, [2]Mohamad Farhan Mohamad Mohsin &
[3]Muhammad Syafiq Mohd Pozi
[1,2&3]School of Computing,
Universiti Utara Malaysia, Malaysia
[1]Artificial Intelligence Department,
Chongqing Institute of Engineering, China

*[1]hd_wang_work@163.com
[2]farhan@uum.edu.my
[3]syafiq.pozi@uum.edu.my
*Corresponding author

Received: 23/5/2024 Revised: 18/7/2024 Accepted: 21/7/2024 Published: 28/7/2024

ABSTRACT

The fuzzy C-ordered-means clustering (FCOM) is a fuzzy clustering algorithm that enhances robustness and clustering accuracy through the ordered mechanism based on fuzzy C-means (FCM). However, despite these improvements, the FCOM algorithm's effectiveness remains unsatisfactory due to the significant time cost incurred by its ordered operation. To address this problem, an investigation was conducted on the ordered weighted model of the FCOM algorithm leading to proposed enhancements by introducing the beta distribution weighted fuzzy C-ordered-means clustering (BDFCOM). The BDFCOM algorithm utilises the properties of the Beta distribution to weight sample features, thus not only circumventing the time cost problem of the traditional ordered mechanism but also reducing the influence of noise. Experiments were conducted on six UCI datasets to validate the effectiveness of the BDFCOM, comparing its performance against seven other clustering algorithms using six evaluation indices. The results show that compared to the average of the other seven algorithms, BDFCOM improves about 15 percent on F1-score, 11 percent on Rand Index, 13 percent on Adjusted Rand Index, 3 percent on Fowlkes-Mallows Index and 16 percent on Jaccard Index. For the other two ordered mechanism FCM algorithms, the time consumption was also reduced by 90.15 percent on average. The proposed algorithm, which designs a new way of feature weighting for ordered mechanisms, advances the field of ordered mechanisms. And, this paper provides a new method in the application field where there is a lot of noise in the dataset.

Keywords: Fuzzy clustering, beta distribution, feature weighted, ordered mechanism.

INTRODUCTION

With the advent of the big data era, data presents unprecedented richness and complexity. How to mine valuable information from these data has become a common focus of researchers and industry players (Harumeka & Purwa., 2023; Song et al., 2023; Sutranggono et al., 2024). Fuzzy clustering utilises fuzzy mathematics to quantify the fuzzy relationship between samples so that it can accurately and objectively characterise the distribution of samples (Seman & Mohd Sapawi, 2018; Singh & Kumar, 2024; Wang & Xue, 2024). The fuzzy C-means (FCM) algorithm is one of the representative algorithms for fuzzy clustering (Davis & Economou, 1984). The FCM improves the traditional clustering algorithm by introducing the idea of fuzzy mathematics, which makes the clustering results more in line with the actual situation of the data. However, the FCM also has some shortcomings, the most prominent of which is the sensitivity to noise data.

To address this problem, researchers have conducted extensive research (Krishnapuram & Keller, 1993; Lu et al., 2024; Yin et al., 2024). Leski (2016) proposed a fuzzy C-ordered-means clustering (FCOM) algorithm. This algorithm effectively improves its robustness and enables it to achieve better results when dealing with datasets that contain noise. Besides that, the feature-weighted fuzzy C-ordered-means clustering algorithm (FWFCOM) algorithm saves time and cost by optimising the ordered mechanism (Yongli et al., 2019). It adopts the order of the samples instead of the sorting number, which can represent the weights of the samples more accurately. However, its performance is still low, and it cannot meet the needs of some complex application scenarios.

Based on the limitation in the existing solution, this paper proposes the beta distribution weighted fuzzy C-ordered-means (BDFCOM) clustering algorithm. Building upon the foundation of the FCOM algorithm, BDFCOM incorporates a feature weighting approach utilising the beta distribution method. This strategy eliminates the time-consuming ordered operations during the calculation process, enhancing clustering efficiency. Additionally, the beta distribution weighting method enhances the algorithm's performance in handling noise, improving results when dealing with datasets containing substantial noise levels.

RELATED WORK

This section introduces knowledge of fuzzy clustering algorithms and problems such as the FCOM algorithm, range normalisation and beta distribution. This knowledge is preparatory to the algorithms proposed in this paper.

FCOM algorithm

Fuzzy clustering algorithms, such as the FCM, are known for their effectiveness in partitioning data into clusters. However, they often struggle with robustness in the presence of noise data. To address this limitation, Leski (2016) proposed the FCOM with the objective function as Equation 1.

$$J(U,V)=\sum_{c=1}^{C}\sum_{i=1}^{N}\beta_{ci}(u_{ci})^{m}D(x_i,v_c) \tag{1}$$

This objective function is subject to Equation 2.

$$\begin{cases} \underset{\substack{1\le c\le C\\ 1\le i\le N}}{\forall} u_{ci}\in[0,1]; \\ \sum_{c=1}^{C}\beta_{ci}u_{ci}=F_i; \\ 0<\sum_{i=1}^{N}u_{ci}<N. \end{cases} \tag{2}$$

where β_{ci} denotes the typicality represented by the *i*th sample in the *c*th cluster, and F_i is the combined typicality of the *i*th data for all clusters. F_i uses the maximum value in the S-*norm* (Rovatti & Fantuzzi, 1996) the formula is as Equation 3.

$$\underset{1\le i\le N}{\forall} F_i = \beta_{1i} \;\vee\; \beta_{2i} \;\vee\; \beta_{3i} \;\vee\; \cdots \;\vee\; \beta_{ci} \tag{3}$$

β_{ci} uses the algebraic product in the T-*norm* (Wu, 2022), the formula is as Equation 4.

$$\underset{\substack{1\le c\le C\\ 1\le i\le N}}{\forall} \beta_{ci} = \beta_{ci1}\cdot\beta_{ci2}\cdot \;\cdots\; \cdot\beta_{cij} \tag{4}$$

The function $D(x_i,v_c)$ is computed by H_{cij} and E_{cij}, i.e., $.\,D(x_i,v_c) = \sum_{j=1}^{K} D(x_{ij},v_{cj}) = \sum_{j=1}^{K} H_{cij}\cdot\left(E_{cij}\right)^2$. H_{cij} is a weighted robust parameter. Commonly used formulas are Equations 5 to 11 (Tyler, 2008):

$$H_{cij} = \begin{cases} 0, E_{cij} = 0; \\ L\,(E_{cij})/(E_{cij})^2, E_{cij} \neq 0. \end{cases} \tag{5}$$

$$H_{cij} = \begin{cases} 0, E_{cij} = 0; \\ 1/\left|E_{cij}\right|, E_{cij} \neq 0. \end{cases} \tag{6}$$

$$H_{cij} = \begin{cases} 1/\delta^2, |\, E_{cij} \,| \le \delta; \\ 1/(\delta \,|\, E_{cij} \,|), |\, E_{cij} \,| > \delta. \end{cases} \tag{7}$$

$$H_{cij} = \begin{cases} 0, E_{cij} = 0; \\ 1/\{(E_{cij})^2[1+\exp(\text{-}\alpha(|\, E_{cij} \,|\text{-}\beta))]\}, E_{cij} \neq 0. \end{cases} \tag{8}$$

$$H_{cij} = \begin{cases} 0, E_{cij} = 0; \\ 1/\{E_{cij}[1+\exp(\text{-}\alpha(|\, E_{cij} \,|\text{-}\beta))]\}, E_{cij} \neq 0. \end{cases} \tag{9}$$

$$H_{cij} = \begin{cases} 0, E_{cij} = 0; \\ \log(1+E_{cij}^2)/E_{cij}^2, E_{cij} \neq 0. \end{cases} \tag{10}$$

$$H_{cij} = \begin{cases} 0, E_{cij} = 0; \\ \log(1+E_{cij}^2)/\,|\, E_{cij} \,|, E_{cij} \neq 0. \end{cases} \tag{11}$$

The parameter E_{cij} is the distance between x_{ij} *and* v_{cj} (the distance between *j*th attribute of the *i*th sample and the *j*th attribute of the center of mass of the *c*th cluster), with Equation 12.

$$E_{cij} = x_{ij} - v_{cj} \tag{12}$$

Construct the sorting function π( ):{1,2,...,*N*}→{1,2,...,*N*}, and perform the ordered operation on all E_{cij} corresponding to the same attribute in the same cluster to satisfy as Equation 13.

$$\left|E_{c\pi(1)j}\right| \le \left|E_{c\pi(2)j}\right| \le \left|E_{c\pi(3)j}\right| \le \cdots \le \left|E_{c\pi(N)j}\right| \tag{13}$$

According to the OWA operator, an operator α_i similar to OWA (Garg & Arora, 2019) can be obtained, and its sectional linearity can be expressed as Equation 14.

$$\alpha_i = \{[(p_c N - i) / (2 p_l N) + 0.5] \wedge 1\} \vee 0 \tag{14}$$

where *i* denotes the ordered ordinal number, and the smaller the value of *i*, the more advanced the ordinal, and the higher the corresponding weight.

The S-shaped linearity of the operator α_i can be expressed as Equation 15.

$$\alpha_i = 1 / \{1 + \exp[\frac{2.944}{p_a N}(i - p_c N)]\} \tag{15}$$

where p_c, p_l and p_a are all greater than 0 and are used to control the slope of the OWA.

For sectional linearity, the magnitude of *k* is from p_cN-p_lN to p_cN+p_lN with a value domain of [1,0] in the form of a segmented descending straight line, while for S-shaped linearity, the magnitude of *k* is from p_cN-p_aN to p_cN+p_aN with a value domain of [0.95,0.05] in the form of an S-shaped descending curve.

The parameter β_{cij} is obtained from α_i obtained from the OWA operation and the ordered function π( ):{1,2,...,*N*}→{1,2,...,*N*}, i.e., $\beta_{cij} = \alpha_{\pi^{-1}(i)}$.

After minimising Equation 1, the iterative formulas for u_{ci} and v_{cj} are obtained as Equation 16 to 17.

$$\underset{\substack{1<i<N \\ 1<c<C}}{\forall} u_{ci} = F_i D(x_i, v_s)^{\frac{1}{1-m}} / [\sum_{t=1}^{C} \beta_{ti} D(x_i, v_t)^{\frac{1}{1-m}}] \tag{16}$$

$$\underset{\substack{1<c<C \\ 1<j<K}}{\forall} v_{cj} = [\sum_{i=1}^{N} \beta_{ci}(u_{ci})^{m} H_{cij} x_{ij}] / [\sum_{i=1}^{N} \beta_{ci}(u_{ci})^{m} H_{cij}] \qquad (17)$$

Range Normalisation

The range of the attributes of the data affects the iteration of the clustering centre, and the feature weights that most of the algorithms add are also affected by the magnitudes of the different ranges (Keshkeh et al., 2022; Khairuddin et al., 2023; Sainin et al., 2021). Therefore, when confronted with data of different ranges or orders of magnitude, performing some transformation on them is usually necessary before comparing or computing (Abu-Shareha, 2022; Fauzi et al., 2022; Sharif et al., 2024).

Range normalisation is one of the most commonly used methods (Habib et al., 2024; Park & Eom, 2024; Yang et al., 2024). It eliminates the interference caused by different scales on the data analysis by linearly transforming the original data and mapping the data values into specific intervals, such as [0,1] (Chen et al., 2024; Liao et al., 2024; Roszkowska & Wachowicz, 2024). Range normalisation can improve accuracy, efficiency and stability and reduce the risk of overfitting. In addition, range normalization facilitates the interpretation of results (Amorim et al., 2023; Herbreteau & Kervrann, 2024; Radzi et al., 2024). The specific formula is Equation 18.

$$x\_norm = \frac{x - x\_min}{x\_max - x\_min} \qquad (18)$$

Where *x_norm* is the regularised data, *x* is the original data, and *x_min* and *x_max* are the minimum and maximum values in that dataset, respectively.

Beta Distribution

The beta distribution is a continuous probability distribution defined on the interval [0,1], denoted as x~Beta(α, β) (Forbes et al., 2011). The flexibility of the beta distribution allows it to adapt to a wide range of probability distributions of different shapes, thus making it useful in a variety of application scenarios.

The beta distribution's Probability Density Function (PDF) is defined as Equation 19.

$$f(x;\ \alpha, \beta) = \frac{x^{\alpha-1} * (1-x)^{\beta-1}}{B(\alpha, \beta)} \tag{19}$$

where x is a random variable taking values between [0,1]; α and β are the shape parameters of the distribution, which determine the shape of the distribution.

The Beta function is defined as Equation 20.

$$B(\alpha, \beta) = \int_0^1 t^{\alpha-1} * (1-t)^{\beta-1} dt \tag{20}$$

Beta distribution can show a variety of shapes, including U-shaped, bell-shaped, increasing, decreasing and even linear. When the values of α and β are equal, the beta distribution exhibits perfect symmetry; when the values of α and β are not equal, the beta distribution exhibits skewness. This skewness allows the Beta distribution to describe data with various degrees of skewness flexibly.

THE PROPOSED METHOD

This section explains the enhancements made in the BDFCOM algorithm, specifically addressing the incorporation of beta distribution weighted parameters and their subsequent optimisation.

To optimise the weights of feature attributes, we use beta distribution to calculate the feature weights, which allows more flexibility in the size of the weights and better reflects the impact of attribute size.

First, the BDFCOM algorithm's objective function is Equation 21.

$$J(U,V) = \sum_{c=1}^{C} \sum_{i=1}^{N} \omega_{ci} (u_{ci})^m D(x_i, v_c) \tag{21}$$

where $\boldsymbol{U}$ is the $C \times N$ membership matrix, and each element u_{ci} (u_{ci} [0,1]) represents the membership of each sample to each cluster; the ∈parameter ω_{ci} denotes the weighted value of beta distribution of the ith data to the cth cluster; and $D(x_i, v_c)$ denotes the distance of the ith sample to the cth cluster, which can be computed from H_{cij} and E_{cij}, i.e., . $D(x_i, v_c) = \sum_{j=1}^{K} D(x_{ij}, v_{cj}) = \sum_{j=1}^{K} H_{cij} \cdot \left(E_{cij}\right)^2$. Weighted robustness parameter

H_{cij} and residual E_{cij} are calculated similarly to FCOM, using Equations 5 to 12.

The objective function is subject to Equation 22.

$$\begin{cases} \underset{\substack{1\le c\le C\\1\le i\le N}}{\forall} u_{ci} \in [0,1]; \\ \sum_{c=1}^{C} \omega_{ci} u_{ci} = T_i; \\ 0 < \sum_{i=1}^{N} u_{ci} < N. \end{cases} \tag{22}$$

where T_i is the global beta distribution weighted value about the *i*th sample, i.e., the combined typicality assessment of the *i*th sample about all attributes for all clusters. Calculate T_i through the maximum value method in S-*norm* as Equation 23.

$$\underset{1\le i\le N}{\forall} T_i = \omega_{1i} \vee \omega_{2i} \vee \omega_{3i} \vee \cdots \vee \omega_{ci} \tag{23}$$

For the beta distribution weighted value ω_{ci} of the *i*th sample for the *c*th cluster, it is calculated by the algebraic product method in T-*norm* as Equation 24.

$$\underset{\substack{1\le c\le C\\1\le i\le N}}{\forall} \omega_{ci} = \omega_{ci1} \cdot \omega_{ci2} \cdot \omega_{ci3} \cdot \cdots \cdot \omega_{cij} \tag{24}$$

where the parameter ω_{cij} denotes the beta distribution weighted value of the *j*th attribute value of the *i*th sample to the *j*th attribute of the *c*th cluster.

In the class FCOM algorithm, the construction of the ranking function $\pi(\ ):\{1,2,...,N\}\to\{1,2,...,N\}$ is performed based on E_{cij}, which is responsible for ordering all the samples for each attribute of each category. However, this ordered operation is very time-consuming, especially when dealing with a large number of samples, attributes, or categories. Moreover, the integer order obtained after ordering has the problem of insufficient accuracy in the weight calculation of Equations 14 and 15. Specifically, when there is an obvious affinity difference in the relationship between samples, using integers to represent the order of samples will ignore such subtle differences, resulting in the algorithm not being able to accurately capture the actual relationship between samples, which in turn may affect the accuracy of the clustering results.

To overcome this shortcoming, in this paper, the ordered mechanism is first processed using the range normalisation method, after which beta distribution is used to weigh the ordered items.

The samples under the same attribute are processed by using range normalisation, as in Equation 25.

$$Y_i = \frac{E_{cij} - \min\limits_{1 \le i \le N}\left\{E_{cij}\right\}}{\max\limits_{1 \le i \le N}\left\{E_{cij}\right\} - \min\limits_{1 \le i \le N}\left\{E_{cij}\right\}} \tag{25}$$

The scope of the sample after the range normalisation is [0,1], indicating the distance of the data to the centre of the cluster; the smaller the value, the greater the impact on the cluster, and the higher the corresponding weight; the BDFCOM uses Y_i to indicate the sample order, similar to the order i in Equations 14 and 15, and the smaller the value of Y_i represents the more forward the order in the ordering.

The range normalisation allows the importance of samples close to the centroids to be more prominently represented. The centroids play a key role in the clustering process, as they tend to be more representative of the core features of that cluster. With range normalisation, the influence of these samples can be captured more accurately, thus improving the effectiveness of clustering.

Then, according to Y_i, which is brought to x in Equations 19 and 20, a more refined Beta distribution weighting function is used to determine the ordered weights ω_{cij}.

There is a major problem with using continuous functions to compute the weights of an algorithm: whenever the weights are computed, integration is required. Such an operation consumes a lot of computational resources and is less efficient. To solve this problem, in this paper, Piecewise Cubic Hermite Interpolating is chosen as the method for calculating the weights. After using the Piecewise Cubic Hermite Interpolating method, the algorithm no longer needs to perform the integral operation each time but directly obtains the weight value through the interpolation function.

After obtaining ω_{cij}, ω_{ci} is computed by using Equation 24.

With the constraints of Equation 22, using the Lagrange multiplier method for Equation 21 and minimising it, the following Lagrange function can be constructed as Equation 26.

$$J(U,V;\lambda_i) = \sum_{c=1}^{C}\sum_{i=1}^{N}\omega_{ci}(u_{ci})^m D(x_i,v_c) \text{-} \lambda_i[\sum_{c=1}^{C}\omega_{ci}u_{ci} \text{-} T_i] \tag{26}$$

Equation 26 is used for U to obtain a partial derivation and set to zero yields as Equation 27.

$$\frac{\partial J}{\partial U} = m\omega_{ci}(u_{ci})^{m-1}D(x_i,v_c) - \lambda_k\omega_{ci} = 0 \tag{27}$$

The u_{ci} can be obtained as Equation 28 using Equations 22 and 27:

$$\underset{\substack{1<i<N\\1<c<C}}{\forall} u_{ci} = T_i D(x_i,v_s)^{\frac{1}{1-m}} / [\sum_{t=1}^{C}\omega_{ti}D(x_i,v_t)^{\frac{1}{1-m}}] \tag{28}$$

Similarly, using Equation 26 to partialise and set zero for **V** yields as Equation 29.

$$\frac{\partial J}{\partial V} = \text{-}2\sum_{i=1}^{N}\omega_{ci}(u_{ci})^m H_{cij}(x_{ij} - v_{cj}) = 0 \tag{29}$$

The v_{cj} can be obtained as Equation 30 using Equations 29 and 22.

$$\underset{\substack{1<c<C\\1<j<K}}{\forall} v_{cj} = [\sum_{i=1}^{N}\omega_{ci}(u_{ci})^m H_{cij}x_{ij}] / [\sum_{i=1}^{N}\omega_{ci}(u_{ci})^m H_{cij}] \tag{30}$$

The BDFCOM algorithm can be expressed in pseudo-code as Algorithm 1.

Algorithm 1: BDFCOM

Step 1. Determine the number of clusters C and the weight index $m \in (1,\infty)$. Choose theε- sensitivity dissimilarity measure. Choose the weighted robustness parameter computation method. Initialise the membership matrix $U^{(0)}$. Setting the Iteration Threshold ξ. $\omega_{ci}=1$, $H_{cij}=1$, $T_i=1$ and set the number of iterations $T=1$;

Step 2. Construct the Beta Distribution function using Equations 19 and 20 with the number of samples N. The interpolation function PCHIP is constructed by Piecewise Cubic Hermite Interpolating by taking 1000 discrete points uniformly in the interval;

(continued)

Algorithm 1: BDFCOM

```
    Step 3. Update the cluster center matrix $V^{(T)}$ using Equation 30;
    Step 4. Update the residual $E_{cij}$ using Equation 12;
    Step 5. Update the weighted robustness parameter $H_{cij}$ using Equations 5 to 11;
    Step 6. Calculate the dissimilarity metric distance $D_{ci}$;
    Step 7. Calculate the ordered ordinal number $Y_i$ using Equation 25;
    Step 8. Calculate the Beta distribution weight of each attribute per sample for each attribute per cluster $\omega_{cij}$ using interpolation function PCHIP;
    Step 9. Calculate the Beta distribution weight of each attribute per sample for each cluster $\omega_{ci}$ using Equation 24;
    Step 10. Calculate the Beta distribution weight for each sample $T_i$ using Equation 23;
    Step 11. Update the membership matrix $U^{(T)}$ using Equation 28;
    Step 12. If $||U^{(T)}-U^{(T-1)}||_F>\xi$, make $T \leftarrow T+1$ and go to Step 3; otherwise end.
```

ANALYSIS AND RESULTS

This section outlines the experimental setup. It encompasses the dataset utilised and the evaluation employed to assess the performance of the proposed BDFCOM algorithm.

Experimental Datasets

Six datasets are selected from the UCI database to evaluate the performance of the BDFCOM algorithm. The information of each dataset is shown in Table 1, and the experimental data used for the experiments in this paper are real datasets (Markelle et al., n.d.).

Table 1

Experimental Datasets

Dataset	Sample Size	Attribute Count	Cluster Number	Explicit Explanation
Iris	150	4	3	It is the best-known database in pattern recognition. The Iris dataset contains 3 classes and 50 samples per class with 4 attributes per sample. One class can be separated linearly from the other two.
Zoo	101	16	7	It is the simple Zoo dataset from the UCI, and each sample contains 17 Boolean-valued attributes. There are 101 samples in total, each with 7 animal classifications.
Win	178	13	3	This dataset results from a chemical analysis of three different varieties of wines produced in the same region of Italy from the UCI dataset. The analyses determined the amount of 13 components in each of these 3 types of wines. There are 178 samples in total.
Waveform	5000	21	3	It is the dataset from UCI - Waveform Database Generator (Version 1), with 5000 samples and 21 attribute values, divided into 3 categories. Each class is generated by combining 2 of the 3 “base” waves. Each sample is generated by adding noise (mean 0, variance 1) to each attribute.
Pendigits	3498	16	10	It is a dataset for a pen-based handwritten digit recognition database. Publicly available in the UCI Machine Learning Library. There are 3498 samples, 10 classifications (0 to 9), and 16 attribute values.
Lung cancer	32	56	3	It is the dataset for lung cancer from the UCI. This dataset describes 3 types of pathologic lung cancers with 32 samples, each with 56 attribute values.

Experimental Results and Analysis

The experiment consists of two parts: the first part of the experiment verifies the performance of the algorithm, including the robustness under the original and noise datasets, and compares the BDFCOM algorithm with the FCM, PCM, FCOM, FWFCOM, LKFCM_LK (Hu et al., 2021), LKFCM_FS (Hu et al., 2021), and ANNDP_WFCM (Chunhao et al., 2023) under six types of evaluation criteria; the second part of the experiment focuses on evaluating the efficiency of BDFCOM against FCOM-like algorithms.

In this paper, six evaluation criteria were used to comprehensively evaluate the results, which are F1-Score, Rand index/adjusted Rand index, Fowlkes-Mallows index, Jaccard index and time cost. Among them, the F1-Score and Fowlkes-Mallows index takes precision and recall into account; the Rand index was used to measure the degree of consistency between clustering results and the real classification, while the adjusted Rand index takes into account the randomised division, which better evaluates the consistency of the clustering model with the real classification; the Jaccard index is used to compare the degree of similarity between clustering results and the real classification; time cost is used to compare the algorithm's computational efficiency (Campello, 2007; Christen et al., 2023; Materum & Teologo Jr, 2021).

During the experimental process, for the seven algorithms, the fuzzy index $m = 2$ was set; in the BDFCOM, the values of α and β from Equations 19 to 20 were $\alpha = 2.5$ and $\beta = 0.4$ for controlling the slope, respectively; for Equation 7, the parameters of BDFCOM, FCOM, and FWFCOM were set the same, the weighted robust function of similarity measure was set with parameter $\delta = 1.0$ for thresholding control; for Equations 8 and 9, the weighted loss function of similarity measure was set with parameter $\alpha = 6.0$ and parameter $\beta = 1.0$ for controlling the slopes; for Equation 14, the parameter $p_c = 0.5$ and the parameter $p_l = 0.2$ were set to control the slope of the segmented linear OWA operator. For Equation 15, the parameter $p_c = 0.5$ and parameter $p_a = 0.2$ were set to control the slope of the S-shaped linear OWA operator. The clustering results of each algorithm were evaluated by taking the average of 10 runs. Noise data were added to the six original datasets in 10%, 20%, 30%, 40% and 50% to form the noise dataset.

For the Iris, none of the algorithms broke down at 0-50% noise; for the Zoo, the PCM broke down at 0-40% noise, and the rest of the

algorithms did not break down; for the Win, the PCM broke down at 30-50% noise, the LKFCM_LK broke down at 20% and 50% noise, and the rest of the algorithms did not break down. For the Waveform, the PCM broke down at 0-50% noise, the FCM, FWFCOM, LKFCM_LK, LKFCM_FS, and ANNDP_WFCM broke down at 10-50% noise, and FCOM and BDFCOM broke down at 40-50% noise. For the Pendigits, the PCM broke down at 0% and 20-50% noise, and the rest of the algorithms did not break down. For the Lungcancer, the FCM, PCM, FWFCOM, LKFCM_LK, LKFCM_FS, and ANNDP_WFCM broke down at 10-50% noise, and the BDFCOM did not break down.
1) F1-Score

Table 2 provides the experimental results of eight algorithms on the six datasets. It can be seen that BDFCOM gets the highest value among all the algorithms for the F1-Score.

Table 2

F1-Score of Each Algorithm on the Original Dataset

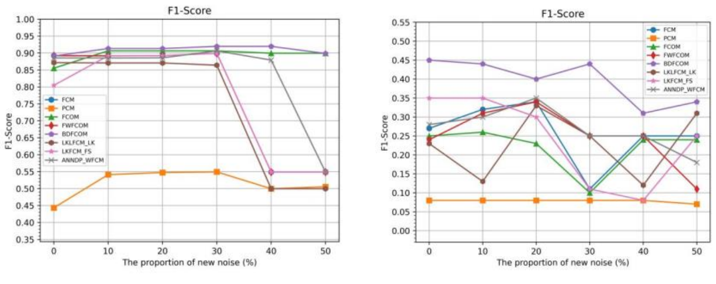
Dataset	FCM	PCM	FCOM	FWFCOM	LKFCM_LK	LKFCM_FS	ANNDP_WFCM	BDFCOM
Iris	0.89	0.44	0.85	0.89	0.87	0.80	0.88	0.89
Zoo	0.27	0.08	0.25	0.24	0.23	0.35	0.28	0.45
Win	0.68	0.12	0.33	0.69	0.69	0.29	0.68	0.71
Waveform	0.26	0.16	0.23	0.41	0.29	0.3	0.41	0.42
Pendigits	0.15	0.01	0.20	0.22	0.15	0.16	0.18	0.33
Lungcancer	0.23	0.07	0.26	0.22	0.23	0.24	0.28	0.56

The experimental results for the noise dataset are shown in Figure 1. Excluding the F1-Score when the algorithm broke down, in the Iris, Zoo, Win, Waveform, Pendigits, and Lung cancer, with the gradual addition of noise data, the F1-Score of the BDFCOM was better than that of the other algorithms in all cases. In the Iris, the F1-Score of BDFCOM, FCM, and FWFCOM had the highest values at the beginning, with the addition of noise data, when the noise was added to 10% and 20%. The F1-Score of BDFCOM, FCOM and ANNDP_WFCM were the highest when the noise was added to 30%. The F1-Score of BDFCOM, FCOM and ANNDP_WFCM had the highest value, with the noise at 40%. The F1-Score of BDFCOM is the highest value. When the noise was added to 50%, the F1-Score of BDFCOM and FCOM were the highest value.

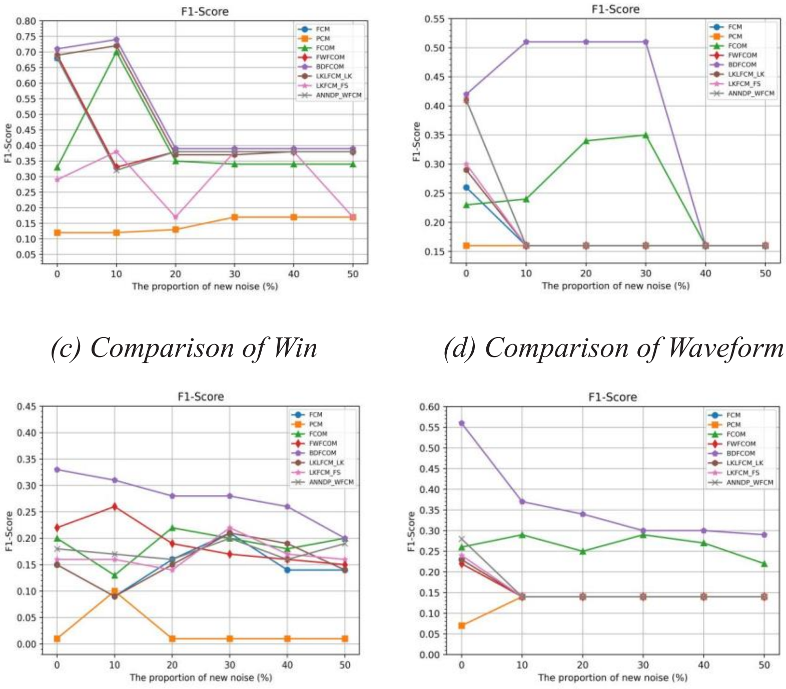
The average F1-Score of FCM, PCM, FCOM, FWFCOM, LKFCM_LK, LKFCM_FS, ANNDP_WFCM, BDFCOM on all datasets for different noise ratios were 0.32(0.05), 0.17(0.02), 0.36(0.05), 0.33(0.05), 0.32(0.06), 0.29(0.06), 0.34(0.05), 0.46(0.04), and the mean of the change in the F1-Score after the addition of noise data is shown in parentheses. Combining *a* to *f* in Figure 1, it can be seen that with the increase of noise data, removing the F1-Score when the algorithm breaks down, the F1-Score value of BDFCOM is higher than (or equal to) the other algorithms. Among the relative changes, FCOM and BDFCOM had small changes, FCM, PCM, FWFCOM, and ANNDP_WFCM had moderate changes, and LKFCM_LK and LKFCM_FS had large changes. In conclusion, after adding different proportions of noise data, BDFCOM had good robustness in maintaining a high F1-Score.

Figure 1

Robustness Comparison: F1-Score of Various Algorithms on Different Datasets



(a) Comparison of Iris *(b) Comparison of Zoo*



(c) Comparison of Win

(d) Comparison of Waveform

(e) Comparison of Pendigits

(f) Comparison of Lungcancer

2) Rand Index/Adjusted Rand Index

Table 3 lists the experimental results of eight algorithms on the six datasets. It can be seen that BDFCOM gets the highest value for the Rand index. Among them, BDFCOM was better than other algorithms for the Rand index in the Win and Pendigits. In the Iris, BDFCOM obtained the highest value along with FCOM, FWFCOM, and ANNDP_WFCM; in the Zoo, BDFCOM obtained the highest value along with FCOM and LKFCM_FS. In the Waveform, BDFCOM obtained the highest value with FCM; in the Lung cancer, BDFCOM obtained the highest value with FCOM, LKFCM_LK, and LKFCM_FS. To summarise, BDFCOM obtained the highest Rand index in the experiment.

Table 3

Rand Index of Each Algorithm on the Original Dataset

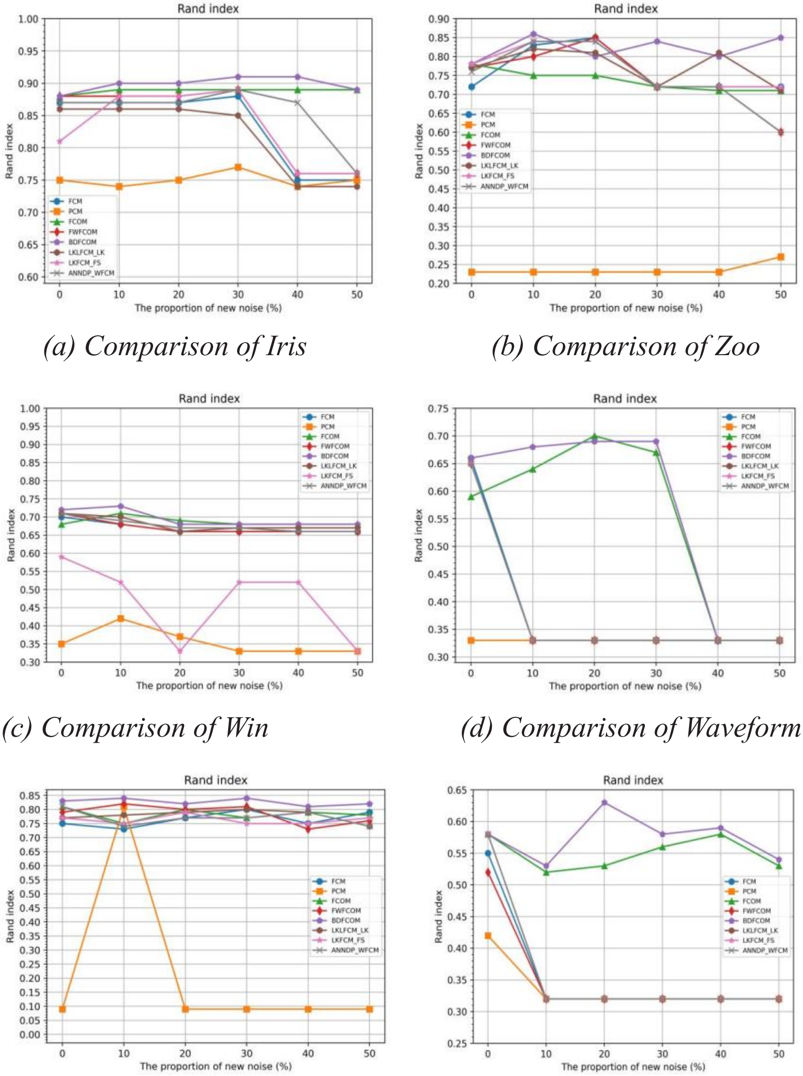
Dataset	FCM	PCM	FCOM	FWFCOM	LKFCM_LK	LKFCM_FS	ANNDP_WFCM	BDFCOM
Iris	0.87	0.75	0.88	0.88	0.86	0.81	0.88	0.88
Zoo	0.72	0.23	0.78	0.77	0.77	0.78	0.28	0.78
Win	0.70	0.35	0.68	0.71	0.71	0.59	0.68	0.72
Waveform	0.66	0.33	0.59	0.65	0.65	0.65	0.65	0.66
Pendigits	0.75	0.09	0.81	0.79	0.77	0.77	0.18	0.83
Lung cancer	0.55	0.42	0.58	0.52	0.58	0.58	0.28	0.58

The experimental results for the noise dataset are shown in Figure 2. Excluding the Rand index when the algorithm broke down, in the Pendigits, with the gradual addition of noise data, the Rand index of BDFCOM was better than the other algorithms in all the situations. In the Iris, the Rand indexes of BDFCOM and FCOM, FWFCOM were all the highest values initially. As the noise data were added when the noise was 10-40%, the Rand index of BDFCOM was the highest value, and when the noise was 50%, the Rand index of both BDFCOM and FCOM was the highest value, and it can be seen that the Rand index of BDFCOM is always at the highest value. In Lung cancer, the Rand index of BDFCOM with FCOM, LKFCM_LK, LKFCM_FS, and ANNDP_WFCM were initially at the highest value. As noise data were added, the Rand index of BDFCOM was at its highest value when the noise value was 10-50%, which can be seen that the Rand index of BDFCOM is always at the highest value.

The Rand index of BDFCOM was also better than all other algorithms, with the gradual addition of noise data in the Zoo, Win and Waveform. The average Rand index of FCM, PCM, FCOM, FWFCOM, LKFCM_LK, LKFCM_FS, ANNDP_WFCM, and BDFCOM with different noise proportions for all datasets were 0.53 (0.03), 0.31 (0.05), 0.59 (0.02), and 0.54 (0.03), 0.54(0.03), 0.51(0.04), 0.54(0.03), 0.62(0.02), and the mean of the change in the Rand index after adding noise data is shown in parentheses. Combining *a* to *f* in Figure 2, it can be seen that with the increase of noise data, excluding the Rand index when the algorithm broke down, the Rand index of BDFCOM was higher than other algorithms. Among the relative changes, FCOM and BDFCOM had small relative changes, FCM, FWFCOM, LKFCM_LK, and ANNDP_WFCM had moderate relative changes, and PCM and LKFCM_FS had significant relative changes. In conclusion, after adding different proportions of noise data, BDFCOM is robust enough to maintain a high Rand index.

Figure 2

Robustness Comparison: Rand Index of Various Algorithms on Different Datasets



(a) Comparison of Iris *(b) Comparison of Zoo*

(c) Comparison of Win *(d) Comparison of Waveform*

(e) Comparison of Pendigits *(f) Comparison of Lungcancer*

Table 4 lists the experimental results of eight algorithms on the six datasets. It can be seen that BDFCOM obtained the highest value for the adjusted Rand index. Among them, BDFCOM was better than other algorithms for the adjusted Rand index in the Zoo, Win, Pendigits, and Lung cancer. In the Iris, BDFCOM obtained the highest value along with FCM and FWFCOM. In the Waveform, BDFCOM obtained the highest value along with FCM. In summary, BDFCOM obtained the highest adjusted Rand index in the experiment.

Table 4

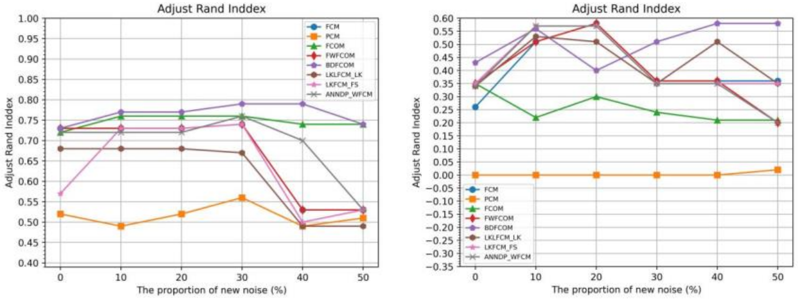
Adjusted Rand Index of Each Algorithm on the Original Dataset

Dataset	FCM	PCM	FCOM	FWFCOM	LKFCM_LK	LKFCM_FS	ANNDP_WFCM	BDFCOM
Iris	0.73	0.52	0.72	0.73	0.68	0.57	0.72	0.73
Zoo	0.26	0.00	0.35	0.35	0.34	0.35	0.34	0.43
Win	0.34	0.01	0.33	0.36	0.36	0.12	0.35	0.37
Waveform	0.25	0.00	0.15	0.23	0.23	0.23	0.23	0.25
Pendigits	0.23	0.00	0.20	0.23	0.24	0.24	0.25	0.30
Lungcancer	0.05	0.06	0.06	-0.01	0.05	0.03	0.06	0.08

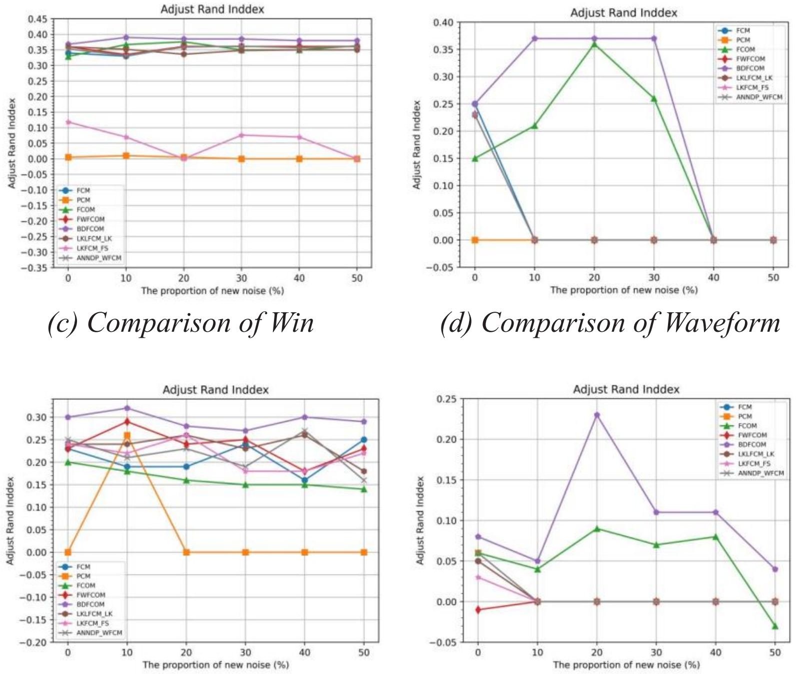
The experimental results for the noise dataset are shown in Figure 3. Excluding the adjusted Rand index when the algorithm broke down, in the Win, Waveform, Pendigits, and Lung cancer, where the noise data were added gradually, the adjusted Rand index of BDFCOM was better than that of the other algorithms in all situations. In the Iris, the adjusted Rand index of BDFCOM with FCM and FWFCOM was the highest value initially. With the addition of noise data, when the noise was 10-40%, the adjusted Rand index of BDFCOM was the highest value, and when the noise was 50%, the adjusted Rand index of BDFCOM and FCOM were the highest value. It can be seen that the adjusted Rand index of BDFCOM always had the highest value. In the Zoo, where noise data was gradually added, the adjusted Rand index of BDFCOM was also better than the other algorithms. The average adjusted Rand index of FCM, PCM, FCOM, FWFCOM, LKFCM_LK, LKFCM_FS, ANNDP_WFCM, and BDFCOM with different noise proportions in all datasets were 0.28 (0.04), 0.09 (0.02), 0.28 (0.04), 0.27 (0.04), 0.27 (0.04), 0.22 (0.05), 0.28 (0.04), 0.37 (0.05), and the mean of the change under the adjusted Rand index after the addition of noise data is shown in parentheses. Combining *a* to *f* in Figure 3, it can be seen that with the increase of noise data, excluding the adjusted Rand index when the algorithm broke down, the adjusted Rand index of BDFCOM was higher than that of other algorithms. In conclusion, after adding different proportions of noise data, BDFCOM had better robustness in maintaining a higher adjusted Rand index.

Figure 3

Robustness Comparison: Adjusted Rand Index of Various Algorithms on Different Datasets



(a) Comparison of Iris *(b) Comparison of Zoo*



(c) Comparison of Win

(d) Comparison of Waveform

(e) Comparison of Pendigits

(f) Comparison of Lungcancer

3) Fowlkes-Mallows Index

Table 5 lists the experimental results of eight algorithms on the six datasets. It can be seen that BDFCOM obtained higher values for the Fowlkes-Mallows index. Among them, BDFCOM is better than other algorithms on Fowlkes-Mallows index metrics in the Zoo and Pendigits. In the Iris, BDFCOM obtained the highest value along with FCM, FCOM, and FWFCOM. In the Win, BDFCOM obtained the highest value along with FCOM. PCM obtained the highest Fowlkes-Mallows index in the Waveform, while FCM obtained the highest Fowlkes-Mallows index in the Lung cancer, and the Fowlkes-Mallows index of BDFCOM was close to them. To summarise, BDFCOM obtained a higher (highest or better) Fowlkes-Mallows index overall in the experiment.

Table 5

Fowlkes-Mallows Index of Each Algorithm on the Original Dataset

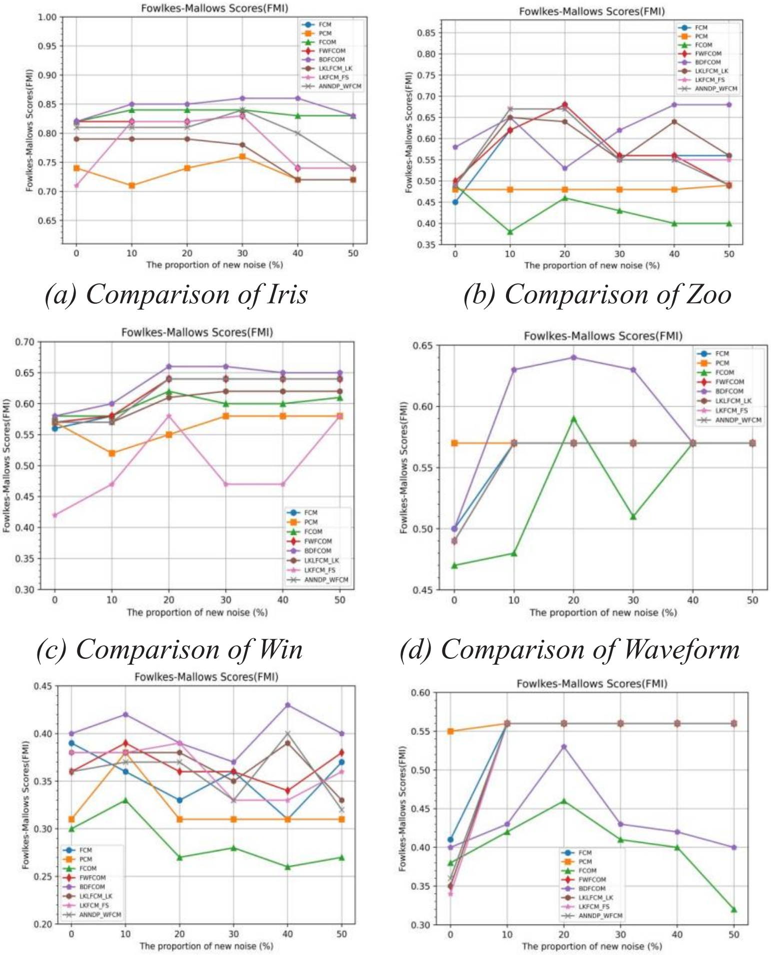
Dataset	FCM	PCM	FCOM	FWFCOM	LKFCM_LK	LKFCM_FS	ANNDP_WFCM	BDFCOM
Iris	0.82	0.74	0.82	0.82	0.79	0.71	0.81	0.82
Zoo	0.45	0.48	0.49	0.50	0.49	0.49	0.49	0.58
Win	0.56	0.57	0.58	0.57	0.57	0.42	0.57	0.58
Waveform	0.50	0.57	0.47	0.49	0.49	0.49	0.49	0.50
Pendigits	0.39	0.31	0.30	0.36	0.38	0.38	0.36	0.40
Lung cancer	0.41	0.55	0.38	0.35	0.35	0.34	0.36	0.40

The experimental results for the noise dataset are shown in Figure 4. Excluding the Fowlkes-Mallows index when the algorithm broke down, in the Pendigits, where the noise data were added gradually, the Fowlkes-Mallows index of BDFCOM was better than the other algorithms in all situations. In the Iris, the Fowlkes-Mallows index of BDFCOM with FCM, FCOM, and FWFCOM was the highest value initially. With the addition of noise data, the Fowlkes-Mallows index of BDFCOM was the highest value when the noise was 10-40%, and when the noise was 50%, the Fowlkes-Mallows index of both BDFCOM and FCOM was the highest value. It can be seen that the Fowlkes-Mallows index of BDFCOM always had the highest value. In the Win, the Fowlkes-Mallows index of BDFCOM and FCOM were both at the highest value at the beginning, and with the addition of noise data, when the noise was 10-50%, it can be seen that the Fowlkes-Mallows index of BDFCOM was at the highest value all the time.

In the Waveform, the Fowlkes-Mallows index of BDFCOM and FCM were both at the highest value at the beginning, and with the addition of noise data, when the noise was 10-50%, it can be seen that the Fowlkes-Mallows index of BDFCOM was at the highest value all the time. In the Zoo and Lung cancer, where noise data were gradually added, the Fowlkes-Mallows index of BDFCOM was also better than the other algorithms. The average Fowlkes-Mallows index of FCM, PCM, FCOM, FWFCOM, LKFCM_LK, LKFCM_FS, ANNDP_WFCM, and BDFCOM with different noise proportions in all datasets were 0.57(0.03), 0.53(0.01), 0.51(0.03), 0.57(0.03), 0.56(0.03), 0.54(0.04), 0.57(0.03), 0.58(0.03), and the mean of the change in the Fowlkes-Mallows index after the addition of noise data is shown in parentheses. Combining *a* to *f* in Figure 4, it can be seen that with the increase of noise data, excluding the Fowlkes-Mallows index when the algorithm broke down, the Fowlkes-Mallows index of BDFCOM is higher than that of other algorithms. In conclusion, after adding different proportions of noise data, BDFCOM had better robustness in maintaining a higher Fowlkes-Mallows Index.

Figure 4

Robustness Comparison: Fowlkes-Mallows Index of Various Algorithms on Different Datasets



(a) Comparison of Iris

(b) Comparison of Zoo

(c) Comparison of Win

(d) Comparison of Waveform

(e) Comparison of Pendigits

(f) Comparison of Lungcancer

4) Jaccard Index

Table 6 lists the experimental results of eight algorithms on the six datasets. It can be seen that BDFCOM gets the highest value for the Jaccard index. Among them, BDFCOM was better than other algorithms on the Jaccard index in the Zoo, Win, Waveform, Pendigits, and Lung cancer; in the Iris, BDFCOM obtained the highest value along with FCM and FWFCOM. To summarise, BDFCOM achieved the highest Jaccard index in the experiment.

Table 6

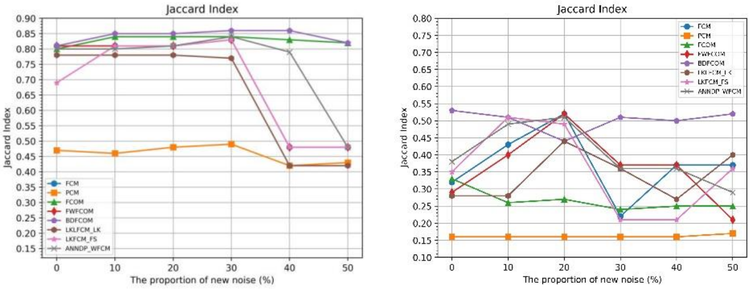
Jaccard Index of Each Algorithm on the Original Dataset

Dataset	FCM	PCM	FCOM	FWFCOM	LKFCM_LK	LKFCM_FS	ANNDP_WFCM	BDFCOM
Iris	0.81	0.47	0.80	0.81	0.78	0.69	0.80	0.81
Zoo	0.32	0.16	0.33	0.29	0.28	0.35	0.38	0.53
Win	0.55	0.10	0.37	0.56	0.56	0.30	0.55	0.58
Waveform	0.24	0.10	0.20	0.28	0.27	0.29	0.28	0.30
Pendigits	0.11	0.01	0.13	0.15	0.09	0.10	0.11	0.26
Lung cancer	0.20	0.05	0.25	0.17	0.20	0.22	0.24	0.38

The experimental results for the noise dataset are shown in Figure 5. Excluding the Jaccard index when the algorithm broke down, in the Win, Waveform, Pendigits, and Lung cancer, with the gradual addition of noise data, the Jaccard index of BDFCOM was better than that of the other algorithms in all the situations. In the Iris, the Jaccard index of BDFCOM with FCM and FWFCOM were all at the highest value at the beginning. With the addition of noise data, when the noise was 10-40%, the Jaccard index of BDFCOM was the highest value, and when the noise was 50%, the Jaccard index of BDFCOM and FCOM were all at the highest value. It can be seen that the Jaccard index of BDFCOM always had the highest value. With noise data added gradually in the Zoo, the Jaccard index of BDFCOM was also better than the other algorithms. The average Jaccard index of FCM, PCM, FCOM, FWFCOM, LKFCM_LK, LKFCM_FS, ANNDP_WFCM, and BDFCOM with different noise proportions in all datasets were 0.30 (0.05), 0.15 (0.01), 0.34 (0.04), 0.31 (0.05), 0.30 (0.04), 0.26 (0.07), 0.32 (0.05), 0.44 (0.03), and the mean of the change in the Jaccard index after adding the noise data is shown in parentheses. Combining *a* to *f* in Figure 5, it can be seen that with the increase of noise data, excluding the Jaccard index when the algorithm broke down, the Jaccard index of BDFCOM was higher than other algorithms in whole. The relative changes of the Jaccard index of PCM and BDFCOM were small, the relative changes of the Jaccard index of FCM, FCOM, FWFCOM, LKFCM_LK, ANNDP_WFCM were moderate, and the relative changes of the Jaccard index of LKFCM_FS were large. For LKFCM_FS was relatively large. In conclusion, after adding different proportions of noise data, BDFCOM is robust enough to maintain a high Jaccard Index.

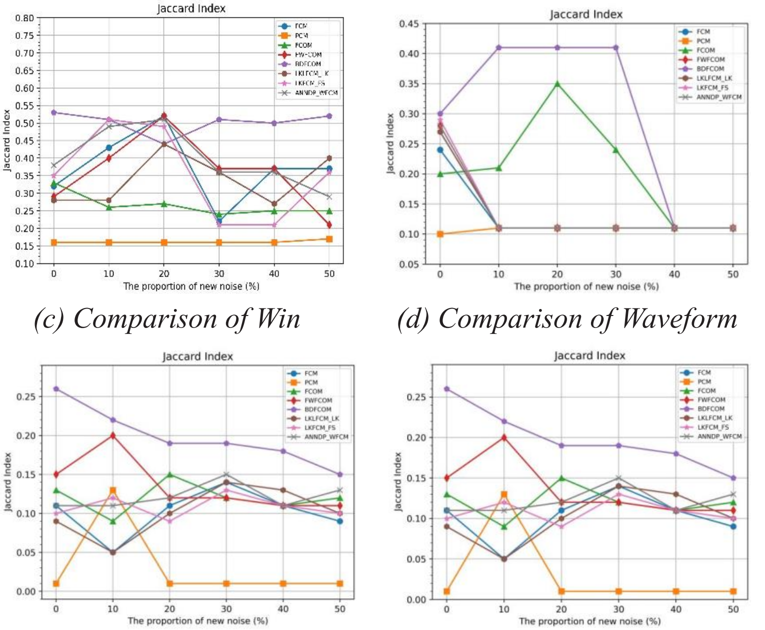
Figure 5

Robustness Comparison: Jaccard Index of Various Algorithms on Different Datasets



(a) Comparison of Iris

(b) Comparison of Zoo



(c) Comparison of Win

(d) Comparison of Waveform

(e) Comparison of Pendigits

(f) Comparison of Lung cancer

5) Time cost

Table 7 shows the running time of each clustering algorithm in the six datasets. The experimental results show that FCM is the most efficient, while FCOM, FWFCOM, and BDFCOM incorporate an ordered operation and, therefore, have longer running times. However, the time cost of the BDFCOM decreased drastically compared to FCOM and FWFCOM because BDFCOM improves the ordering process, which is the most time-consuming.

Table 7

Time Cost of Each Algorithm on the Original Dataset (Measurement Unit: Sec)

	FCM	PCM	FCOM	FWFCOM	LKFCM_LK	LKFCM_FS	ANNDP_WFCM	BDFCOM
Iris	0.03	0.06	1.42	0.58	0.08	0.04	0.07	0.07
Zoo	0.62	0.15	2.47	16.07	0.38	0.42	0.46	1.11
Win	0.09	0.05	10.41	4.00	0.11	0.10	0.08	0.64
Waveform	0.68	1.29	7688.62	12321.17	6.05	6.9	9.11	69.94
Pendigits	1.95	2.44	31519.75	660.63	1.29	2.29	1.98	5.65
Lungcancer	0.03	0.03	5.66	0.23	0.02	0.09	0.04	0.42

Compared to FCOM and FWFCOM, BDFCOM reduced the runtime on the Iris, Zoo, Win, Waveform, Pendigits, and Lung cancer by 95.07% and 87.93%, 55.06% and 93.09%, 93.85% and 84.00%, 99.09% and 99.43%, 99.98% and 99.14%, 92.58% and 82.61%, with an average reduction of 89.27% and 91.03%. Especially for the Waveforms and Pendigits, the reduction is more significant than others because the time required for the ordered mechanism grows geometrically as the amount and dimensionality of the data increases. Using range normalisation and beta distribution weighted parameters, the order can be obtained indirectly without a time-consuming ordered mechanism, and the samples are subjected to weighting operations, so the clustering efficiency is substantially improved.

In this section, the results suggested that BDFCOM improves by about 15% on F1-score, 11% on the Rand index, 13% on the adjusted Rand index, 3% on the Fowlkes-Mallows index and 16% on the Jaccard index as compared to the average of the other seven algorithms. For the other two ordered FCM algorithms, the time consumption was also reduced by 90.15% on average. The proposed algorithm, which designs a new way of feature weighting for ordered mechanisms, advances the field of ordered mechanisms. And, this paper provides a new method in the application field where there is a lot of noise in the dataset.

CONCLUSION

This paper introduces a beta distribution weighted fuzzy C-ordered-mean clustering algorithm, BDFCOM. The algorithm inherits the ordered mechanism of the FCOM-like algorithm to ensure robustness while adopting a novel approach to ordering feature attributes through range normalisation and obtaining ordered weights using the Beta Distribution. Experiments were conducted on six real and noisy datasets to assess the clustering efficacy. The results demonstrate that BDFCOM consistently exhibits superior performance across various evaluation metrics, including F1-Score, Rand index, adjusted Rand index, Fowlkes-Mallows index, and Jaccard index, as compared to FCM, PCM, FCOM, FWFCOM, LKFCM_LK, LKFCM_FS, and ANNDP_WFCM. Moreover, BDFCOM showcases enhanced robustness to noise data. Furthermore, compared to FCOM and FWFCOM, BDFCOM addresses the computational inefficiency inherent in FCOM-like algorithms.

Although beta distribution has improved the ordered mechanism, the weight calculation is still more complicated than other FCM algorithms, which can continue to be optimised afterwards; moreover, the parameter sensitivity in beta distribution is also an issue worth exploring, and this aspect will be researched afterwards.

ACKNOWLEDGMENT

This work was supported by the Science and Technology Research Program of the Chongqing Municipal Education Commission, China [KJQN202001903, 2020]; the Science and Technology Research Program of the Chongqing Municipal Education Commission, China [KJQN202001901, 2020]; and the Science and Technology Research Program of the Chongqing Municipal Education Commission, China [KJZDM202301901, 2023].

REFERENCES

Abu-Shareha, A. A. (2022). TOPSIS-based regression algorithms evaluation. *Journal of Information and Communication Technology*, *21*(4), 513–547. https://doi.org/10.32890/jict2022.21.4.3

Amorim, L., Cavalcanti, G. D. C., & Cruz, R. M. O. (2023). The choice of scaling technique matters for classification performance. *Applied Soft Computing*, *133*, 109924. https://doi.org/10.1016/j.asoc.2022.109924

Campello, R. J. G. B. (2007). A fuzzy extension of the Rand index and other related indexes for clustering and classification assessment. *Pattern Recognition Letters*, *28*(7), 833–841. https://doi.org/10.1016/j.patrec.2006.11.010

Chen, W., Hu, Y., Peng, M., & Zhu, B. (2024). Positional normalisation-based mixed-image data augmentation and ensemble self-distillation algorithm. *Expert Systems With Applications*, 124140. https://doi.org/10.1016/j.eswa.2024.124140

Christen, P., Hand, D. J., & Kirielle, N. (2023). A review of the F-Measure: its history, properties, criticism, and alternatives. *ACM Computing Surveys*, *56*(3), 1–24. https://doi.org/10.1145/3606367

Chunhao, Z., Bin, X., Ximei, Z., & Tongtong, X. (2023). Weighted fuzzy clustering algorithm combining adaptive nearest neighbors and density peaks. *Journal of Chinese Computer Systems*, *44*(9), 1974-1982. https://doi.org/10.20009/j.cnki.21-1106/TP.2021-0988

Davis, R. H., & Economou, C. (1984). A review of fuzzy clustering methods. *Advances in Engineering Software*, *6*(4), 189–191. https://doi.org/10.1016/0141-1195(84)90002-0

Fauzi, N. S. M., Kasim, M. M., & Desa, N. H. M. (2022). Construction of air pollution index with the inclusion of aggregated weights of the pollutants. *Journal of Information and Communication Technology*, *21*(4), 495-512. https://doi.org/10.32890/jict2022.21.4.2

Forbes, C., Evans, M., Hastings, N., & Peacock, B. (2010). *Statistical distributions*. John Wiley & Sons.

Garg, H., & Arora, R. (2018). Generalised intuitionistic fuzzy soft power aggregation operator based ont-norm and their application in multicriteria decision-making. *International Journal of Intelligent Systems*, *34*(2), 215–246. https://doi.org/10.1002/int.22048

Habib, G., Malik, I. A., Ahmad, J., Ahmed, I., & Qureshi, S. (2024). Exploring the efficacy of Group-Normalization in deep learning models for Alzheimer's disease classification. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2404.00946

Harumeka, A., & Purwa, T. (2023). Does the physical type of house still affect household poverty in Indonesia? An entropy-based fuzzy weighted logistic regression approach. *Journal of Information and Communication Technology*, *22*(3), 337–361. https://doi.org/10.32890/jict2023.22.3.2

Herbreteau, S., & Kervrann, C. (2024, February 23). *On normalisation-equivariance properties of supervised and unsupervised denoising methods: a survey*. arXiv.org. https://arxiv.org/abs/2402.15352

Ho, P. L., Lee, C., Le, C., Nguyen, P. H., & Yee, J. (2024). A computational homogenisation for yield design of asymmetric microstructures using adaptive bES-FEM. *Computers & Structures*, *294*, 107271. https://doi.org/10.1016/j.compstruc.2023.107271

Hu, J., Wu, M., Chen, L., & Pedrycz, W. (2022). A novel modeling framework based on customised Kernel-Based fuzzy C-Means clustering in iron ore sintering process. *IEEE/ASME Transactions on Mechatronics*, *27*(2), 950–961. https://doi.org/10.1109/tmech.2021.3076208

Keshkeh, K., Jantan , A. ., & Alieyan, K. . (2022). A machine learning classification approach to detect tls-based malware using entropy-based flow set features. *Journal of Information and Communication Technology*, *21*(3), 279–313. https://doi.org/10.32890/jict2022.21.3.1

Khairuddin, A. R., Alwee, R., & Haron, H. (2023). Hybrid neighbourhood component analysis with Gradient tree boosting for feature selection in forecasting crime rate. *Journal of Information and Communication Technology*, *22*(2), 207–229. https://doi.org/10.32890/jict2023.22.2.3

Krishnapuram, R., & Keller, J. M. (1993). A possibilistic approach to clustering. *IEEE Transactions on Fuzzy Systems*, *1*(2), 98–110. https://doi.org/10.1109/91.227387

Leski, J. M. (2016). Fuzzy c-ordered-means clustering. *Fuzzy Sets and Systems*, *286*, 114–133. https://doi.org/10.1016/j.fss.2014.12.007

Liao, H., Xiao, Y., Wu, X., & Baušys, R. (2024). Z-DNMASort: A double normalisation-based multiple aggregation sorting method with Z-numbers for multi-criterion sorting problems. *Information Sciences*, *653*, 119782. https://doi.org/10.1016/j.ins.2023.119782

Lu, J., Ma, G., & Zhang, G. (2024). Fuzzy machine learning: A comprehensive framework and systematic review. *IEEE Transactions on Fuzzy Systems*, 1–18. https://doi.org/10.1109/tfuzz.2024.3387429

Markelle, K., Rachel, L., & Kolby, N. (n.d.). Datasets. *The UCI Machine Learning Repository*. https://archive.ics.uci.edu/

Materum, L., & Teologo Jr., A. T. (2021). An improved K-power means technique using Minkowski distance metric and dimension weights for clustering wireless multipaths in indoor channel scenarios. *Journal of Information and Communication Technology*, *20*(4), 541–563. https://doi.org/10.32890/jict2021.20.4.4

Park, C. W., & Eom, I. K. (2024). Underwater image enhancement using adaptive standardisation and normalisation networks. *Engineering Applications of Artificial Intelligence*, *127*, 107445. https://doi.org/10.1016/j.engappai.2023.107445

Radzi, A. R., Farouk, A. M., Romali, N. S., Farouk, M., Elgamal, M., & Rahman, R. A. (2024). Assessing environmental management plan implementation in water supply construction projects: Key performance indicators. *Sustainability*, *16*(2), 600. https://doi.org/10.3390/su16020600

Roszkowska, E., & Wachowicz, T. (2024). Impact of normalisation on entropy-based weights in Hellwig's method: A case study on evaluating sustainable development in the education area. *Entropy*, *26*(5), 365. https://doi.org/10.3390/e26050365

Rovatti, R., & Fantuzzi, C. (1996). S-norm aggregation of infinite collections. *Fuzzy Sets and Systems*, *84*(3), 255–269. https://doi.org/10.1016/0165-0114(95)00320-7

Sainin, M. S., Alfred, R., & Ahmad, F. (2021). Ensemble meta classifier with sampling and feature selection for data with imbalance multiclass problem. *Journal of Information and Communication Technology*, *20*(2), 103–133. https://doi.org/10.32890/jict2021.20.2.1

Seman, A., & Mohd Sapawi, A. (2018). Extensions to the K-Amh algorithm for numerical clustering. *Journal of Information and Communication Technology*, *17*(4), 587–599. https://doi.org/10.32890/jict2018.17.4.8272

Sharif, N. A. M., Harun, N. H., & Yusof, Y. (2024). Colour image enhancement model of retinal fundus image for diabetic retinopathy recognition. *Journal of Information and Communication Technology*, *23*(2), 293-334.

Singh, A., & Kumar, M. (2023). Bayesian fuzzy clustering and deep CNN-based automatic video summarisation. *Multimedia Tools and Applications*, *83*(1), 963–1000. https://doi.org/10.1007/s11042-023-15431-9

Song, J., Lee, H., & Kwon, O.-Y. (2023). Investigating job mismatch in software industry through news big data. *Journal of Information and Communication Technology*, *22*(1), 31–48. https://doi.org/10.32890/jict2023.22.1.2

Sutranggono, A. N., Riyanarto Sarno, & Imam Ghozali. (2024). Multi-class multi-level classification of mental health disorders based on textual data from social media. *Journal of Information and Communication Technology*, *23*(1), 77–104. https://doi.org/10.32890/jict2024.23.1.4

Takamiya, K., Iwamoto, Y., Nonaka, M., & Chen, Y. (2023). CT brain image synthesization from MRI brain images using CycleGAN. *2023 IEEE International Conference on Consumer Electronics (ICCE)*. https://doi.org/10.1109/icce56470.2023.10043572

Tyler, D. E. (2008). Robust statistics: Theory and methods. *Journal of the American Statistical Association*, *103*(482), 888–889. https://doi.org/10.1198/jasa.2008.s239

Wang, Y., & Xue, Q. (2024). Fault identification of product design using fuzzy clustering generative adversarial network (FCGAN) model. *Soft Computing*, *28*(4), 3725–3742. https://doi.org/10.1007/s00500-024-09636-9

Wu, C. W. (2022). On rearrangement inequalities for T-norm logics. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2204.06051

Yang, A., Peng, B., Lu, C., He, Z., Chen, E. T., & Sheng, F. (2024b). A novel method for multiple targets localisation based on normalised cross-correlation adaptive variable step-size dynamic template matching. *AIP Advances*, *14*(4). https://doi.org/10.1063/5.0194376

Yin, H., Aryani, A., Petrie, S., Nambissan, A., Astudillo, A., & Cao, S. (2024). A rapid review of clustering algorithms. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2401.07389

Yongli, L., Hengda, W., Jing, L., & Lishen, Y. (2019). Feature weighted fuzzy C-ordered-means clustering algorithm. *Journal of Henan Polytechnic University* (*Natural Science*), *38*(3), 123-130. https://doi.org/10.16186/j.cnki.1673-9787.2019.3.17