J Inf Process Syst, Vol.15, No.2, pp.359~373, April 2019
https://doi.org/10.3745/JIPS.01.0040

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Incremental Fuzzy Clustering Based on a Fuzzy Scatter Matrix

Yongli Liu*, Hengda Wang*, Tianyi Duan*, Jingli Chen*, and Hao Chao*

Abstract

For clustering large-scale data, which cannot be loaded into memory entirely, incremental clustering algorithms are very popular. Usually, these algorithms only concern the within-cluster compactness and ignore the between-cluster separation. In this paper, we propose two incremental fuzzy compactness and separation (FCS) clustering algorithms, Single-Pass FCS (SPFCS) and Online FCS (OFCS), based on a fuzzy scatter matrix. Firstly, we introduce two incremental clustering methods called single-pass and online fuzzy C-means algorithms. Then, we combine these two methods separately with the weighted fuzzy C-means algorithm, so that they can be applied to the FCS algorithm. Afterwards, we optimize the within-cluster matrix and between-cluster matrix simultaneously to obtain the minimum within-cluster distance and maximum between-cluster distance. Finally, large-scale datasets can be well clustered within limited memory. We implemented experiments on some artificial datasets and real datasets separately. And experimental results show that, compared with SPFCM and OFCM, our SPFCS and OFCS are more robust to the value of fuzzy index m and noise.

Keywords

Fuzzy Clustering, Incremental Clustering, Scatter Matrix

1. Introduction

The development of modern information technology changes every passing day. And thus, data abundance and information overload replace information indigence as the new problem puzzling users, for whom it is more difficult to find valuable information. Gradually, people realize that structure and knowledge behind data is much more important. Therefore, we have to manage to organize the massive data effectively. Data mining is a process of knowledge discovery from huge amount of data, and hence becomes a hot topic in many fields. Among the many techniques of data mining, the well-known clustering technique aims at grouping objects into clusters so that the objects in the same cluster are relatively similar, while the objects in different clusters are relatively dissimilar.

So far many clustering algorithms have been proposed [1]. These algorithms mainly include two types: hard clustering algorithms and soft clustering algorithms, whose representatives are K-means [2] and fuzzy C-means (FCM) [3], respectively. The former algorithm partitions each object into just one cluster, and the latter algorithm allows each object to belong to more than one cluster.

Manuscript received August 4, 2017; first revision September 29, 2017; second revision November 15, 2017; accepted December 12, 2017.
Corresponding Author: Yongli Liu (yongli.buaa@gmail.com)
* School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo, Henan, China (yongli.buaa@gmail.com, chananwhd @126.com, dtyhpu@sina.com, jinglichen_hpu@163.com, chaohao@hpu.edu.cn)

However, both K-means and FCM devote their effort to minimizing the within-cluster scatter matrix trace, which can be interpreted as a compactness measure with a within-cluster variation [4,5]. It means these algorithms just try to make objects in the same cluster as similar as possible. As we know, the within-cluster similarity is only one aspect of clustering. In the other aspect, a clustering algorithm should be able to ensure that objects in different clusters are as dissimilar as possible. In other words, apart from minimizing the within-cluster scatter matrix trace, a clustering algorithm should maximizing the between-cluster scatter matrix trace which can be interpreted as a separation measure with a between-cluster variation [5].

Based on a fuzzy scatter matrix, Wu et al. [5] proposed the fuzzy compactness and separation (FCS) clustering algorithm. In FCS, the compactness is measured using a fuzzy within-cluster variation, and the separation is measured using a fuzzy between-cluster variation. Thus, the FCS could simultaneously consider the within-cluster compactness and the between-cluster separation. Their experimental results showed that this algorithm was efficient and robust.

With the development of information technology, data is growing exponentially. When processing large-scale data, the FCS could be unavailable and inefficient. In this case, an incremental clustering algorithm becomes extremely essential. Incremental methods process data elements one at a time and typically use much less space than needed to store the whole dataset. Nowadays, many methods [6-17] have been designed to solve large-scale data clustering problems. But it is still a challenge to apply fuzzy clustering algorithms to get well-separated clusters in a computation-saved way. Hore et al. [18] proposed two novel incremental clustering approaches, namely single-pass fuzzy C-means (SPFCM) and online fuzzy C-means (OFCM) [19], which treated large-scale datasets as streaming data. Their performances are very close to what you could get if all the data is clustered at one time.

Motivated by above analysis, we propose two incremental FCS algorithms in this paper, namely SPFCS and OFCS. Based on FCS, these two algorithms could simultaneously consider within-cluster compactness and between-cluster separation, and are more robust and efficient. At the same time, because of the 'single-pass' and 'online' incremental strategies, these two algorithms could process large-scale data easily.

In conclusion, the characteristics of our SPFCS and OFCS include: (i) these two algorithms could not consider only within-cluster compactness, but also between-cluster separation, (ii) they could process large-scale data by employing 'single-pass' and 'online' incremental strategies, and (iii) they are more robust to noise and the value of fuzzy parameter.

The remainder of this paper is organized as follows. In Section 2, we provide a literature review of the FCS algorithm and incremental FCM-type clustering algorithms. Section 3 introduces our algorithms in details. Section 4 presents our experiments and discusses the experimental results. Finally, we conclude our work.

2. Related Work

2.1 FCS Algorithm

Before introducing FCS clustering, we list the explanations on the mathematical notations used in this paper in Table 1.

Table 1. List of mathematical notations

Notation	Description
C, N	Numbers of clusters, objects
u_{ci}	Fuzzy object partitioning membership
v_c	Centroid/Medoid of the c^{th} cluster
x_i	The i^{th} object
m	FCM user-defined parameters
w	Weights of objects

For describing within-cluster compactness and between-cluster separation at the same time, Wu et al. [5] proposed the fuzzy total scatter matrix S_{FT}, the fuzzy within-cluster scatter matrix S_{FW} and fuzzy between-cluster scatter matrix S_{FB}. These three matrices are defined as:

$$S_{FT} = \sum_{c=1}^{C}\sum_{i=1}^{N} u_{ci}^{m}(x_i - \bar{x})(x_i - \bar{x})^T \tag{1}$$

$$S_{FW} = \sum_{c=1}^{C}\sum_{i=1}^{N} u_{ci}^{m}(x_i - v_c)(x_i - v_c)^T \tag{2}$$

$$S_{FB} = \sum_{c=1}^{C}\sum_{i=1}^{N} u_{ci}^{m}(v_c - \bar{x})(v_c - \bar{x})^T \tag{3}$$

where m is the weighting exponent (m>1), $\bar{x} = \sum_{i=1}^{N} x_i / N$, and we restrict $\sum_{c=1}^{C} u_{ci} = 1$, $u_{ci} \in [0,1]$. Furthermore, S_{FT}=S_{FW}+S_{FB} is satisfied among S_{FT}, S_{FW} and S_{FB}. For minimizing the within-cluster compactness measure and simultaneously maximizing the between-cluster separation measure, the objective function of FCS is defined as:

$$\begin{aligned} J_{FCS} &= \mathrm{tr}(S_{FW}) - \eta_c \mathrm{tr}(S_{FB}) \\ &= \sum_{c=1}^{C}\sum_{i=1}^{N} u_{ci}^{m} \| x_i - v_c \|^2 - \sum_{c=1}^{C}\sum_{i=1}^{N} \eta_c u_{ci}^{m} \| v_c - \bar{x} \|^2 \end{aligned} \tag{4}$$

where $\|x_i$-$v_c\|^2$ is the square of *Euclidean* distance between the i^{th} data point and the centroid of the c^{th} cluster. To guarantee that no two of the cluster kernels will overlap, η_c is chosen as Eq. (5) such that the parameter will control the influence of between-cluster separation.

$$\eta_c = \frac{(\beta/4)\min_{c' \neq c} \| v_c - v_{c'} \|^2}{\max_k \| v_k - \bar{x} \|^2} \tag{5}$$

where $0 \leq \beta \leq 1.0$, k=1,...,C.

For minimizing the objective function as Eq. (4), we make some restrictions on it.

$$v_c = \frac{\sum_{i=1}^{N} u_{ci}^{m}(x_i - \eta_c \bar{x})}{\sum_{i=1}^{N} u_{ci}^{m}(1 - \eta_c)} \tag{6}$$

$$u_{ci} = \frac{(\| x_i - v_c \|^2 - \eta_c \| v_c - \bar{x} \|^2)^{\frac{-1}{m-1}}}{\sum_{k=1}^{C} (\| x_i - v_k \|^2 - \eta_k \| v_k - \bar{x} \|^2)^{\frac{-1}{m-1}}} \tag{7}$$

For the given data point x_i, if $\| x_i - v_c \|^2 \leq \eta_c \| v_c - \bar{x} \|^2$, then $u_{ci} = 1$, and $u_{c'i} = 0$. That is, each cluster in FCS will have a crisp boundary such that all data points inside this boundary will have a crisp membership value $u_{ij} \in \{0,1\}$ and other data points outside this boundary will have fuzzy membership values $u_{ij} \in [0,1]$, as shown in Fig. 1.

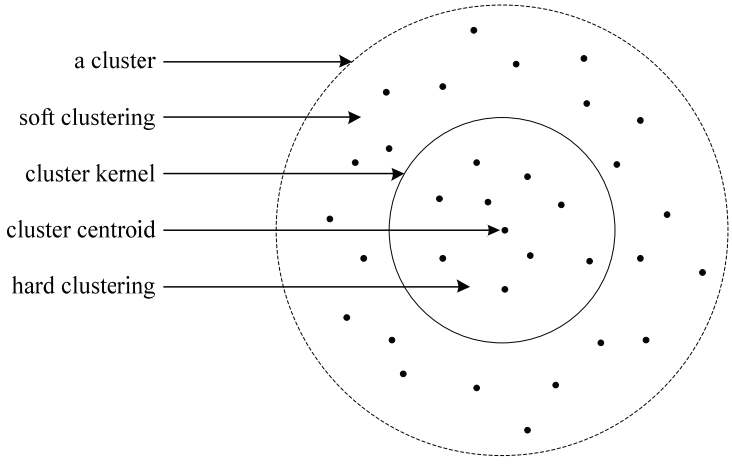


Fig. 1. A sample cluster obtained by FCS.

2.2 Incremental Fuzzy Clustering Algorithm Based on FCM

In this section, we introduce two incremental fuzzy clustering algorithms based on FCM, called SPFCM and OFCM. These two algorithms combine single-pass clustering and online clustering methods separately with the fundamental algorithm weighted fuzzy C-means (WFCM).

2.2.1 WFCM algorithm

WFCM algorithm is a modification of FCM, which weights the centroids obtained by each iteration in the FCM algorithm to ensure that centroids with higher weights are more representative than those with lower weights. For a given dataset $X=[x_1,...,x_n]$, which needs to be clustered into C groups, the goal of WFCM is to minimize the objective function J_{WFCM},

$$J_{WFCM} = \sum_{c=1}^{C} \sum_{i=1}^{N} w_i u_{ci}^m \left\| x_i - v_c \right\|^2 \tag{8}$$

where w_i is the weight of the i-th data point.

The restrictions below are needed so that Eq. (8) could achieve the minimum value.

$$u_{ci} = \frac{\| x_i - v_c \|^{\frac{-2}{m-1}}}{\sum_{k=1}^{C} \| x_i - v_k \|^{\frac{-2}{m-1}}} \tag{9}$$

$$v_c = \frac{\sum_{i=1}^{N} w_i u_{ci}^m x_i}{\sum_{i=1}^{N} w_i u_{ci}^m} \tag{10}$$

2.2.2 Single-pass and online incremental clustering algorithm

Traditional clustering algorithms calculate the entire dataset directly, but they will be not available if the capacity of a single memory is not enough to store the dataset. To solve this problem, Hore et al. [18] proposed two incremental algorithms, SPFCM and OFCM.

In SPFCM, WFCM algorithm is used for clustering. In the starting point, the weight of each point is set to 1, $w_{data}=[1,1,...,1]^T$. The dataset is then divided into several chunks, $X=[X_1,...,X_h]$. When the first chunk comes (t=1), the X_1, the centroid is obtained as $\Delta=[v_1,...,v_c]$ and SPFCM calculates its weight as $w_c = \sum_{i=1}^{n}(u_{ci})w_{data}$, where n is the number of data points in the first chunk. After the first chunk is processed (t>1), a new chunk is generated by merging the previous centroid into the next one, $X^{t'} = [\Delta^{t-1}, X^t]$, and the weight of the new centroid is updated as $w_c^t = \sum_{i=1}^{n}(u_{ci})[w_c^{t-1}, w_{data}]$.

Different from SPFCM, the OFCM algorithm classifies each chunk of data individually using FCM, and then centroids of all the chunks are collected and grouped by performing clustering again. Then the centroids of each chunk are updated to $w=[w_1^1,..,w_c^1,w_1^2,...,w_c^2,...,w_1^h,...,w_c^h]$.

3. Incremental Fuzzy Clustering Algorithms Based on a Fuzzy Scatter Matrix

In this section, we introduce two incremental fuzzy clustering algorithms based on a fuzzy scatter matrix, called SPFCS and OFCS, respectively.

For applying single-pass and online clustering methods to FCS, the data points in FCS should be weighted. First, the weighted within-cluster and between-cluster scatter matrices are defined as

$$S_{IFW} = \sum_{c=1}^{C}\sum_{i=1}^{N} w_i u_{ci}^m \| x_i - v_c \|^2 \tag{11}$$

$$S_{IFB} = \sum_{c=1}^{C}\sum_{i=1}^{N} w_i u_{ci}^m \| v_c - \bar{x} \|^2 \tag{12}$$

Based on above analysis, the objective function of incremental FCS algorithm is designed as:

$$\begin{aligned} J_{IFCS}(U,V) &= \mathrm{tr}(S_{IFW}) - \eta_c \mathrm{tr}(S_{IFB}) \\ &= \sum_{c=1}^{C}\sum_{i=1}^{N} w_i u_{ci}^m \| x_i - v_c \|^2 - \sum_{c=1}^{C}\sum_{i=1}^{N} \eta_c w_i u_{ci}^m \| v_c - \bar{x} \|^2 \end{aligned} \tag{13}$$

restricted by the following constraints:

$$\begin{cases} \min J_{IFCS} \\ s.t. \quad \sum_{c=1}^{C} u_{ci} = 1 \end{cases} \tag{14}$$

where

$$\eta_c = \frac{(\beta/4)\min_{c' \neq c} \| v_c - v_{c'} \|^2}{\max_k \| v_k - \bar{x} \|^2} \tag{15}$$

The constrained optimization of IFCS in Eq. (13) can be solved by applying Lagrange multiplier method, constrained by Eq. (14), yielding the new objective function as,

$$J(U,V) = \sum_{c=1}^{C}\sum_{i=1}^{N} w_i u_{ci}^m \| x_i - v_c \|^2 - \sum_{c=1}^{C}\sum_{i=1}^{N} \eta_c w_i u_{ci}^m \| v_c - \bar{x} \|^2 + \sum_{i=1}^{N} \lambda_i \left(\sum_{c=1}^{C} u_{ci} - 1 \right) \tag{16}$$

Taking the partial derivative of $J(U,V)$ in Eq. (16) with respect to U, and setting the gradient to zero we have

$$\frac{\partial J}{\partial U} = m u_{ci}^{m-1} w_i \| x_i - v_c \|^2 - m u_{ci}^{m-1} \eta_c w_i \| v_c - \bar{x} \|^2 - \lambda_i = 0 \tag{17}$$

Solving the above Eq. (17), subject to the constraints in Eq. (14), yield the formula for u_{ci} as

$$u_{ci} = \frac{(\| x_i - v_c \|^2 - \eta_c \| v_c - \bar{x} \|^2)^{\frac{-1}{m-1}}}{\sum_{k=1}^{C} (\| x_i - v_k \|^2 - \eta_k \| v_k - \bar{x} \|^2)^{\frac{-1}{m-1}}} \tag{18}$$

Likewise, taking the partial derivative of $J(U, V)$ in Eq. (16) with respect to V, and setting the gradient to zero we have

$$\frac{\partial J}{\partial V} = \sum_{i=1}^{N} u_{ci}^m w_i \| x_i - v_c \| + \sum_{i=1}^{N} \eta_c w_i u_{ci}^m \| v_c - \bar{x} \| = 0 \tag{19}$$

Solving the above equation, we get v_c as

$$v_c = \frac{\sum_{i=1}^{N} u_{ci}^m w_i (x_i - \eta_c \bar{x})}{\sum_{i=1}^{N} u_{ci}^m w_i (1 - \eta_c)} \tag{20}$$

The incremental method for SPFCS and OFCS is similar to SPFCM and OFCM, using IFCS as the clustering method.

4. Experiments

4.1 Datasets

We implement experiments on six datasets, including two artificial datasets and four real datasets, whose information is shown as Table 2. In Table 2, the first two datasets are artificial datasets, and the others are real datasets. The two artificial datasets are constructed like the work of Wu et al. [5], and all

the four real datasets are from UCI database. The four UCI datasets contain 150, 2310, 3498 and 20000 sample objects, respectively, and can help to show clustering performance of our algorithms under different data scale.

- In the *Unequal Sample Size* dataset, there are 100 data points with two attributes. The samples can be divided into two groups, one of which has 97 samples and the other has 3 samples.
- The *Noise* dataset is a two-dimensional dataset with the sample size n=400. The data points can be divided into two clusters, one of which has a noisy point.
- The *Iris* dataset is a four-dimensional dataset with 150 data points consisted of 50 points from each of three clusters. Each cluster is linearly separable with the other two clusters.
- The *Statlog Segmentation* dataset was drawn randomly from a database of 7 outdoor images. The images were hand-segmented to create a classification for every pixel. It's a 19-dimensional dataset with 2310 data points. All these samples can be divided into 7 clusters, each of which contains 330 points.
- The *Pen-Based Recognition of Handwritten Digits* dataset was created by collecting 250 samples from 44 writers. We randomly select 3498 samples for the experiment. These 16-dimensional samples can be divided into 10 clusters.
- The *Letter Recognition* dataset is used to identify 26 letters. It's a 16-dimensional dataset with 20,000 samples which can be divided into 26 clusters.

Table 2. Datasets

Datasets	Sample size	Attributes	Clusters
Unequal Sample Size (USS)	100	2	2
Noise	400	2	2
Iris	150	4	3
Statlog Segmentation (SS)	2310	19	7
Pen Digits (PD)	3498	16	10
Letter Recognition (LR)	20000	16	26

4.2 Evaluation Criteria

The clustering results of the algorithm are evaluated by F-measure [20] and entropy.

4.2.1 F-measure

F-measure, also known as F-score, is the weighted harmonic mean of Precision and Recall. The value of F-measure is in [0,1]. The greater the value is, the better the clustering performs. The term is defined as

$$F = \frac{2PR}{P+R} \tag{21}$$

where P denotes the precision, which is the probability that a (randomly selected) retrieved document is relevant, and R denotes the recall, which is the probability that a (randomly selected) relevant document is retrieved in a search.

4.2.2 Entropy

The concept of entropy was originally proposed by the German physicist, Clausius [21] in 1865. It is used to represent the degree of internal chaos of the system. The entropy is defined as

$$E = \sum_{i=1}^{N} p_i \log_2 p_i \tag{22}$$

where p_i represents the occurrence probability of the ^{ith} sample. The smaller the entropy values, the better the clustering results.

4.3 Experimental Results

For the USS dataset, we implement SPFCM, SPFCS, OFCM and OFCS algorithms separately with different values combinations of m for (2, 6) and β for 0.1 [10]. The initial sample size of each chunk is set to 75 points. This group of experimental results are illustrated as Fig. 2. In Fig. 2, the blue circles denote objects of one cluster, the red crosses denote objects of the other cluster, and the dark squares are centroids of the corresponding cluster. The two attributes of this dataset correspond to abscissa valuing from 20 to 100 and ordinate valuing from 20 to 70, respectively. Fig. 2(a) and (b) display clustering results of SPFCM with m=2 and m=6, respectively. We can intuitively see that there should be two clusters, however some objects that should be in one cluster are incorrectly divided into the other. In Fig. 2(a) and (b), objects in the right cluster are sparse, and they are more like noisy data in clustering. In this perspective, SPFCM is easily affected by noisy data. Furthermore, by comparing Fig. 2(a) and (b), SPFCM with different values of m generates results with great difference, because centroid of the second cluster has an apparent displacement. Fig. 2(c) and (d) illustrate clustering results of SPFCS with m=2 and m=6, respectively. The results are obviously better than results of SPFCM. In Fig. 2(c) and (d), objects are grouped into two clusters, in accordance with human's cognition. In other words, the so-called noisy data has very little influence on the SPFCS. Simultaneously, Fig. 2(c) is very similar to Fig. 2(d), which shows that different values of m also have very little influence on the SPFCS.

Fig. 2(e) and (f) show clustering results of OFCM with m=2 and m=6, respectively. We observe that clustering performance of OFCM is very similar to that of spFCM. And therefore, the similar conclusion can be drawn that OFCM is sensitive to noisy data and the value of m. In Fig. 2(g) and (h), clustering results of OFCS with m=2 and m=6 are introduced, respectively. In terms of clustering performance in this group of experiments, OFCS is very similar to SPFCS. Different from SPFCM and OFCM, SPFCS and OFCS are more robust to noise and insensitive to the fuzzy index m.

In the *Noise* dataset, objects have two attributes that correspond to abscissa valuing from -3 to 7 and ordinate valuing from -2 to 2, respectively. These objects are divided into two clusters, separated by 2 in the horizontal axis, shown as Fig. 3. Based on this dataset, we add a noisy data whose coordinate is (100, 0). Fig. 3(a), (c), (e), and (g) illustrate experimental results of SPFCM, SPFCS, OFCM, and OFCS with m=2, respectively, and Fig. 3(b), (d), (f), and (h) display clustering results of SPFCM, SPFCS, OFCM, and OFCS with m=6, respectively. The experimental results show that when m=2, SPFCM and OFCM algorithms obtain accurate clustering results, while when m=6, there exists one data point that is put into wrong cluster for the noisy data. However, SPFCS and OFCS both obtain accurate clustering separation when m is equal to either 2 or 6. It shows that SPFCS and OFCS algorithms are more robust to noise and more insensitive to the value of m than SPFCM and OFCM.

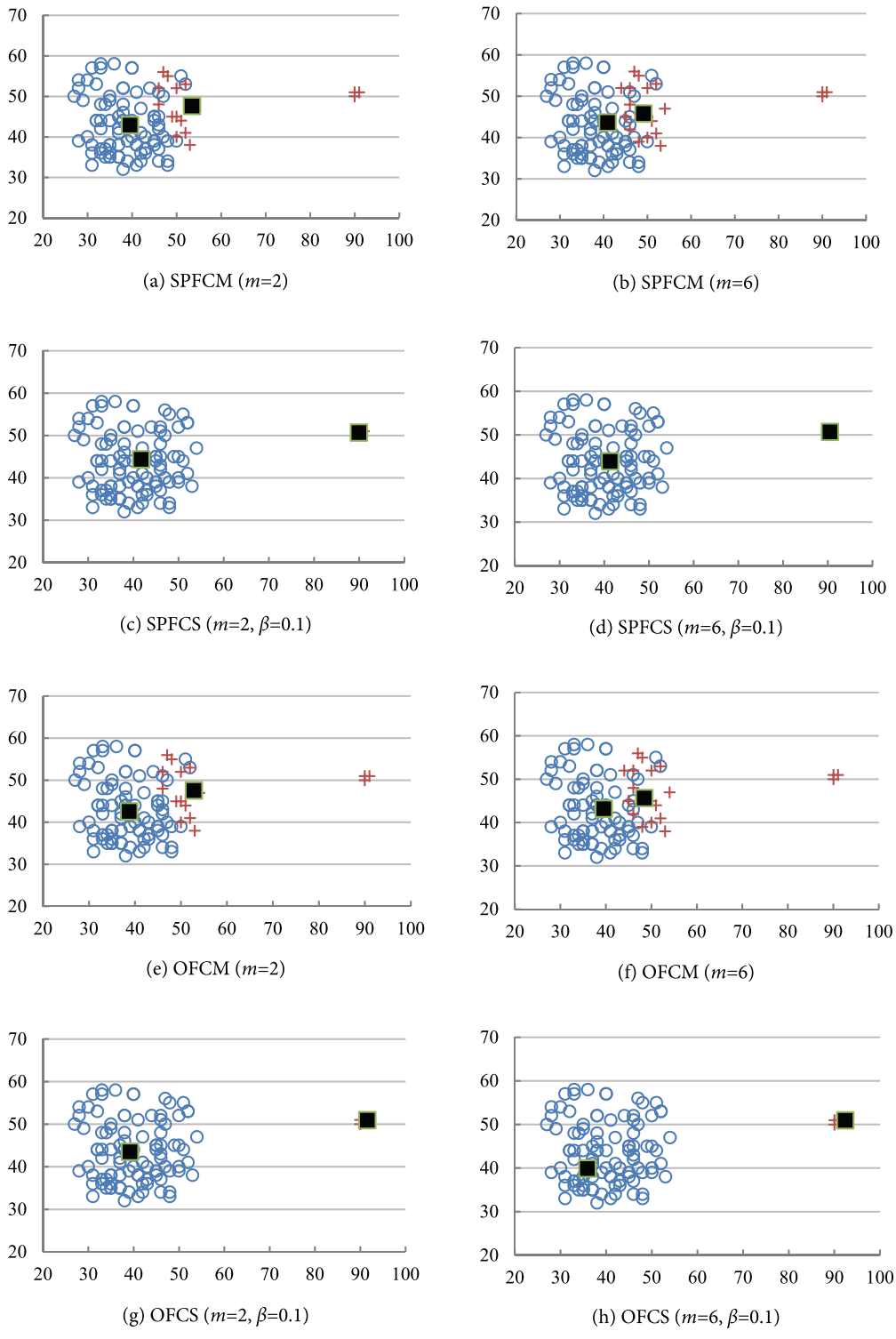


Fig. 2. SPFCM, SPFCS, OFCM, and OFCS clustering results on the USS dataset.

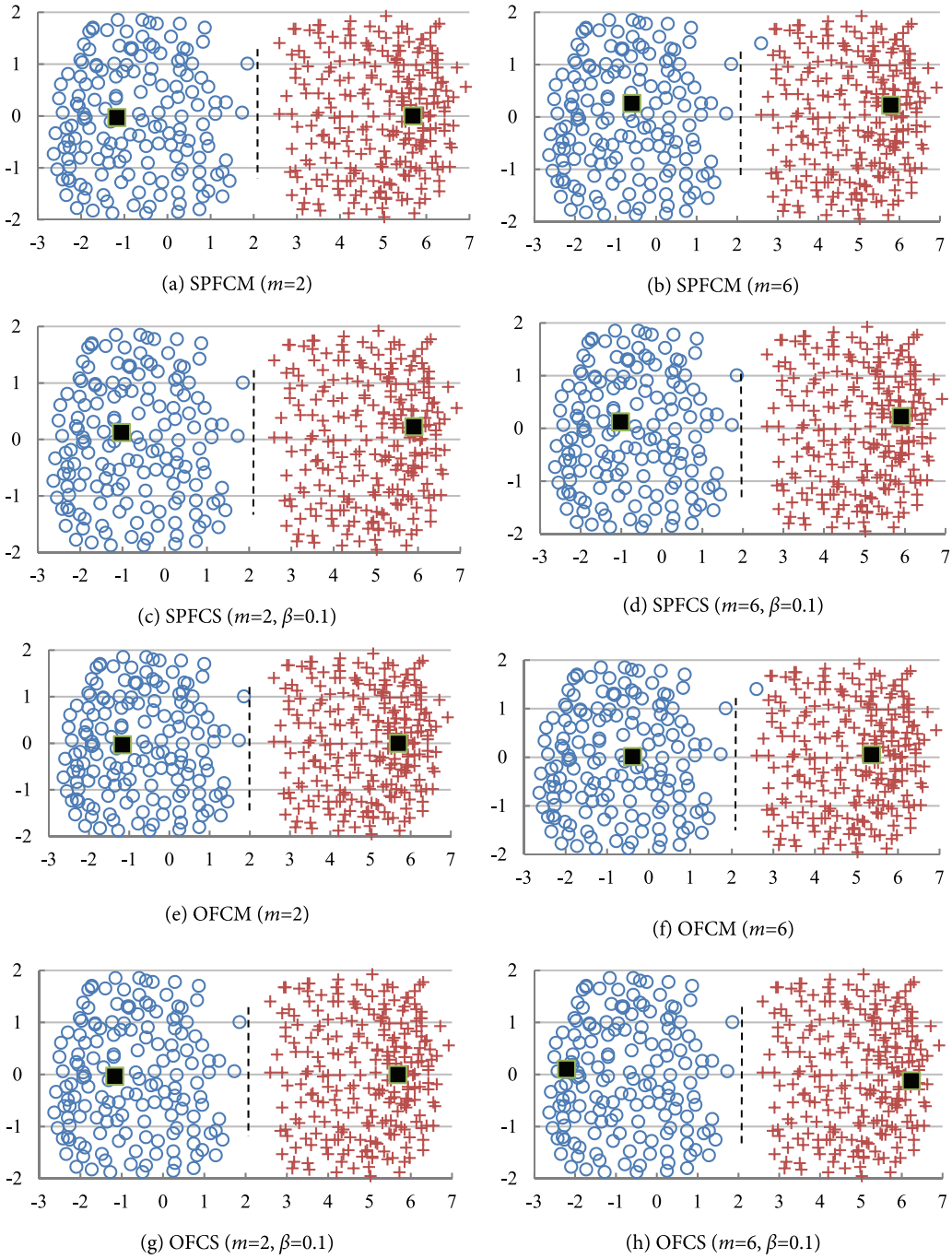


Fig. 3. SPFCM, SPFCS, OFCM, and OFCS clustering results on the Noise dataset.

On the following four real datasets, we implemented six algorithms—SPFCM, single-pass hyperspherical fuzzy C-means (SPHFCM) [10], SPFCS, OFCM, online hyperspherical fuzzy C-means (OHFCM) [10], and OFCS. On the *Iris* dataset, the values of m in these six algorithms value 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, and 6. The sample size of each chunk is set to 30 points. And we choose F-measure as the criteria to

evaluate the performance of the clustering algorithms. As shown in Fig. 4, when the value of *m* becomes larger, the performance of SPFCS and OFCS remain steady, while the F-measure of other algorithms drops down. The extreme deviation values of F-Measure of SPFCS, SPFCM, and SPHFCM are 0.047, 0.086, and 0.148, respectively, and their standard deviation values are 0.018, 0.035, and 0.047, respectively. In the OFCS, OFCM, and OHFCM group, the extreme deviation values are 0.011, 0.173, and 0.068, respectively, and the standard deviation values are 0.004, 0.066, and 0.023, respectively. Obviously, in these two experimental groups, the fluctuation of SPFCS and OFCS with different *m* value is the lowest. Hence, SPFCS and OFCS are more robust to the value of *m* than the others.

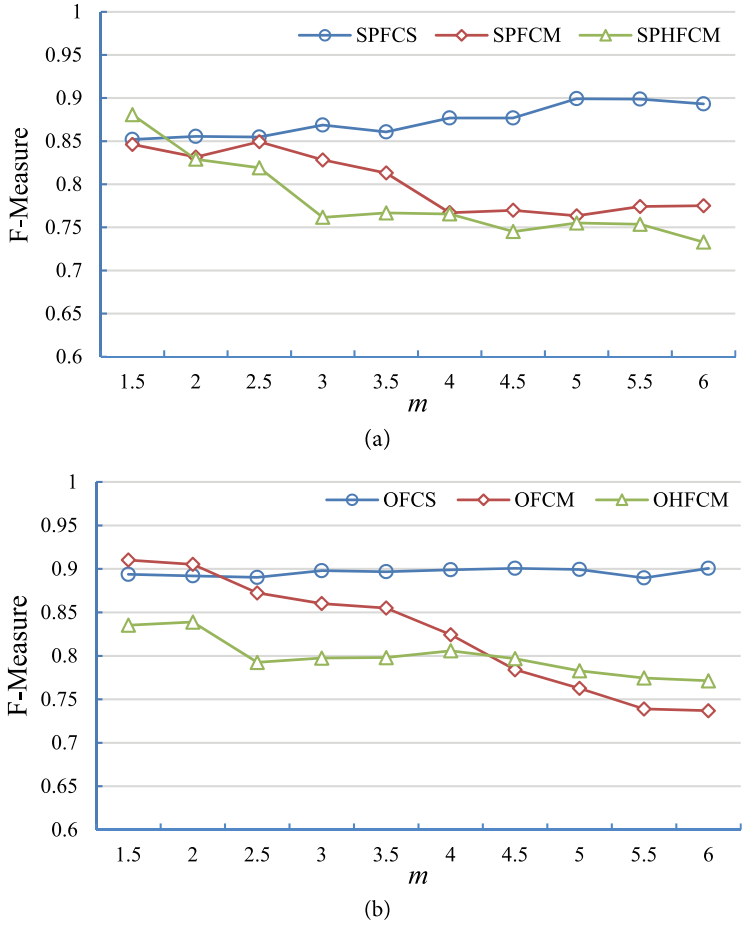


Fig. 4. Influence of *m* on SPFCM, SPHFCM, SPFCS, OFCM, OHFCM, OFCS algorithms for the *Iris* dataset: (a) single-pass algorithm and (b) online algorithm.

Tables 3 and 4 show the performances of SPFCM, SPHFCM, SPFCS, OFCM, OHFCM and OFCS on the four real datasets, in terms of F-measure and entropy, respectively. It can be seen from these two table that SPFCS and OFCS perform better than the other four algorithms in most cases. Mostly, the F-measure values of SPFCS and OFCS are larger than SPFCM, SPHFCM and OFCM, OHFCM separately and their entropy values are relatively lower. Especially in the PD dataset, the average F-measure value of SPFCS is larger than SPFCM and SPHFCM by 70.21% and 118.70% separately. And the average entropy value of

SPFCS is smaller than SPFCM and SPHFCM by 81.38% and 61.14% separately. In the OFCS, OFCM, OHFCM group, the average deviation is 47.04%, 54.99% and 48.08%, 42.48%. So, we conclude that our SPFCS and OFCS algorithms perform better than the other four algorithms in terms of clustering accuracy on the four real datasets.

Table 3. Clustering performance in terms of F-measure on real datasets

Dataset	Sample size	SPFCM	SPHFCM	SPFCS	OFCM	OHFCM	OFCS
Iris	10%	0.81	0.76	0.91	0.89	0.80	0.89
	20%	0.84	0.88	0.91	0.90	0.82	0.90
	50%	0.90	0.68	0.89	0.89	0.90	0.88
SS	10%	0.51	0.42	0.53	0.52	0.42	0.60
	20%	0.50	0.34	0.52	0.46	0.39	0.60
	50%	0.44	0.34	0.56	0.49	0.42	0.61
PD	10%	0.42	0.43	0.65	0.37	0.43	0.69
	20%	0.44	0.28	0.66	0.43	0.44	0.64
	50%	0.34	0.26	0.70	0.34	0.40	0.71
LR	10%	0.14	0.15	0.19	0.15	0.17	0.21
	20%	0.14	0.14	0.18	0.15	0.17	0.20
	50%	0.10	0.10	0.11	0.11	0.15	0.18

Table 4. Clustering performance in terms of entropy on real datasets

Dataset	Sample size	SPFCM	SPHFCM	SPFCS	OFCM	OHFCM	OFCS
Iris	10%	0.16	0.17	0.10	0.12	0.15	0.12
	20%	0.14	0.11	0.11	0.10	0.16	0.10
	50%	0.12	0.27	0.12	0.12	0.10	0.13
SS	10%	0.53	0.62	0.48	0.53	0.62	0.42
	20%	0.54	0.74	0.49	0.56	0.65	0.41
	50%	0.61	0.76	0.45	0.56	0.64	0.40
PD	10%	0.62	0.64	0.35	0.70	0.60	0.34
	20%	0.61	0.84	0.36	0.62	0.60	0.40
	50%	0.76	0.88	0.33	0.75	0.65	0.32
LR	10%	1.25	1.23	1.13	1.25	1.17	1.08
	20%	1.28	1.26	1.16	1.24	1.20	1.09
	50%	1.37	1.38	1.32	1.35	1.26	1.15

The SPFCM, SPHFCM, OFCM and OHFCM are FCM-type clustering algorithms. Although they are all incremental clustering algorithms and able to process large-scale data, they only try to minimize the within-cluster scatter matrix trace like FCM. However, both SPFCS and OFCS try to minimize the within-cluster scatter matrix trace and maximize the between-cluster scatter matrix trace, which concerns the within-cluster compactness and the between-cluster separation simultaneously. Therefore, these two algorithms are easy to get higher clustering accuracy.

In the last part of our experiments, taking the SS dataset as an example, we implement an experiment with SPFCS and OFCS to investigate the influence of noise. By adding 5%, 10%, 20% and 40% noisy points to the dataset separately, we get the results shown as Fig. 5. Note that, when the proportion of noise added into the dataset increases, the clustering performance of SPFCS, OFCS, SPHFCM and OHFCM remains

steady while SPFCM and OFCM declines. By calculating the standard deviation of F-measure value, we can illustrate the fluctuation of these algorithms with different proportion of noise. In Fig. 5(a), the standard deviation values of SPFCS, SPFCM and SPHFCM are 0.018, 0.067 and 0.015 separately. In Fig. 5(b), the values of OFCS, OFCM and OHFCM are 0.015, 0.045 and 0.026 separately. The experimental results show that, compared with SPFCM and OFCM, SPFCS and OFCS are more robust to noise.

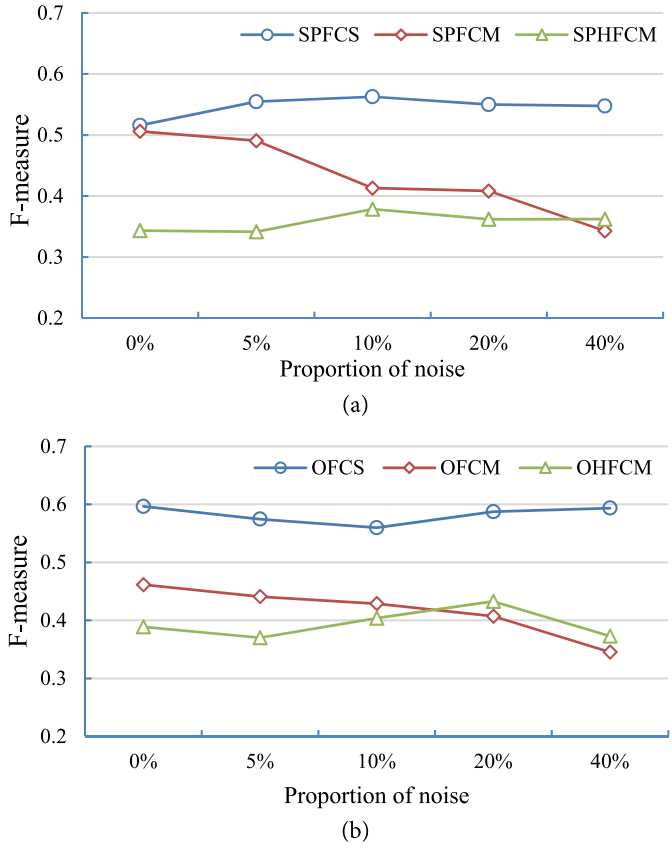


Fig. 5. The influence of noise on clustering: (a) single-pass algorithm and (b) online algorithm.

5. Conclusion

For revealing latent knowledge hidden behind large-scale data, clustering technique has gained wide attention. In the process of clustering, the FCS algorithm considers synthetically within-cluster compactness and between-cluster separation, and therefore easily produces more accurate clustering results. However, this algorithm is difficult to process large-scale data.

Based on a fuzzy scatter matrix, we extend FCS and propose two incremental fuzzy clustering algorithms, SPFCS and OFCS. First, we weight the centroids obtained from each iteration of the FCS algorithm so that the weighted algorithm can be combined with 'single-pass' and 'online' algorithms. Then, we implement experiments with several artificial datasets and real datasets. Experimental results show that, compared with SPFCM, OFCM, SPHFCM and OHFCM, the SPFCS and OFCS algorithms

have better clustering performance. Also, SPFCS and OFCS are more robust to the fuzzy index *m* and noise than the SPFCM and OFCM algorithms.

In clustering processes of SPFCS and OFCS, the number of clustering results needs to be specified in advance. It encourages us to design better K-value prediction algorithms in future studies.

Acknowledgement

The authors would like to thank the support of Foundation for University Key Teacher by Henan Province (No. 2015GGJS-068), Fundamental Research Funds for the Universities of Henan Province (No. NSFRF1616), and Foundation for scientific and technological project of Henan Province (No. 172102210279).

References

[1] Y. Liu and X. Wan, "Information bottleneck based incremental fuzzy clustering for large biomedical data," *Journal of Biomedical Informatics*, vol. 62, pp. 48-58, 2016.

[2] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651-666, 2009.

[3] J. C. Bezdek, R. Ehrlich, and W. Full, "FCM: the fuzzy c-means clustering algorithm," *Computers & Geosciences*, vol. 10, no. 2-3, pp. 191-203, 1984.

[4] Y. Zhou, H. F. Zuo, and J. Feng, "A clustering algorithm based on feature weighting fuzzy compactness and separation," *Algorithms*, vol. 8, no. 2, pp. 128-143, 2015.

[5] K. L. Wu, J. Yu, and M. S. Yang, "A novel fuzzy clustering algorithm based on a fuzzy scatter matrix with optimality tests," *Pattern Recognition Letters*, vol. 26, no. 5, pp. 639-652, 2005.

[6] C. Y. Chen, S. C. Hwang, and Y. J. Oyang, "An incremental hierarchical data clustering algorithm based on gravity theory," in *Advances in Knowledge Discovery and Data Mining*. Heidelberg: Springer, 2002, pp. 237-250.

[7] S. Young, I. Arel, T. P. Karnowski, and D. Rose, "A fast and stable incremental clustering algorithm," in *Proceedings of 2010 7th International Conference on Information Technology: New Generations (ITNG)*, Las Vegas, NV, 2010, pp. 204-209.

[8] S. Chakraborty and N. K. Nagwani, "Analysis and study of incremental k-means clustering algorithm," *High Performance Architecture and Grid Computing*. Heidelberg: Springer, 2011, pp. 338-341.

[9] L. E. Aik and T. W. Choon, "An incremental clustering algorithm based on Mahalanobis distance," in *AIP Conference Proceedings*, vol. 1635, pp. 788-793, 2014.

[10] J. P. Mei, Y. Wang, L. Chen, and C. Miao, "Incremental fuzzy clustering for document categorization," in *Proceedings of 2014 IEEE International Conference on Fuzzy Systems*, Beijing, China, 2014, pp. 1518-1525.

[11] Y. Wang, L. Chen, and J. P. Mei. "Incremental fuzzy clustering with multiple medoids for large data," *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 6, pp. 1557-1568, 2014.

[12] M. F. K. Minhas, R. A. Abbasi, N. R. Aljohani, A. A. Albeshri, M. Mushtaq, "INTWEEMS: a framework for incremental clustering of tweet streams," in *Proceedings of the 17th International Conference on Information Integration and Web-based Applications & Services*, Brussels, Belgium, 2015.

[13] L. Pradeep and A. M. Sowjanya, "Multi-density based incremental clustering," *International Journal of Computer Applications*, vol. 116, no. 17, pp. 6-9, 2015.

[14] O. Shmueli and L. Shnaiderman, "Incremental clustering of indexed XML data," U.S. Patent 8930407, Jan 6, 2015.

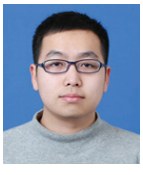
[15] F. Cambi, P. Crescenzi, and L. Pagli, "Analyzing and comparing on-line news sources via (two-layer) incremental clustering," in *Proceedings of the 8th International Conference on Fun with Algorithms*, La Maddalena, Italy, 2016.
[16] L. Chen, M. Liu, C. Wu, and A. Xu, "A novel clustering algorithm and its incremental version for large-scale text collection," *Information Technology and Control*, vol. 45, no. 2, pp. 136-147, 2016.
[17] D. Wang and A. H. Tan, "Self-regulated incremental clustering with focused preferences," in *Proceedings of 2016 International Joint Conference on Neural Networks (IJCNN)*, Vancouver, Canada, 2016, pp. 1297-1304.
[18] P. Hore, L. O. Hall, and D. B. Goldgof, "Single pass fuzzy c means," in *Proceedings of 2007 International Fuzzy Systems Conference*, London, UK, 2007, pp. 1-7.
[19] P. Hore, L. O. Hall, D. B. Goldgof, and W. Cheng, "Online fuzzy c means," in *Proceedings of 2008 Annual Meeting of the North American Fuzzy Information Processing Society*, New York, NY, 2008, pp. 1-5.
[20] A. Maratea, A. Petrosino, and M. Manzo, "Adjusted F-Measure and kernel scaling for imbalanced data learning," *Information Sciences*, vol. 257, pp. 331-341, 2014.
[21] R. Clausius, "Ueber verschiedene für die Anwendung bequeme Formen der Hauptgleichungen der mechanischen Wärmetheorie," *Annalen der Physik*, vol. 201, no. 7, pp. 353-400, 1865.

Yongli Liu https://orcid.org/0000-0002-0540-865X

He received his Ph.D. degree in computer science and engineering from Beihang University in 2010. He is currently an associate professor in Henan Polytechnic University. His current research interests include data mining and information retrieval.

Hengda Wang https://orcid.org/0000-0003-3405-8860

He is currently a master student in Henan Polytechnic University. His current research interests include data mining.

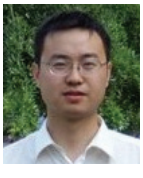


Tianyi Duan https://orcid.org/0000-0001-6258-4089

He is currently a master student in Henan Polytechnic University. His current research interests include data mining and information retrieval.

Jingli Chen https://orcid.org/0000-0002-8866-0219

She is currently a master student in Henan Polytechnic University. Her current research interests include data mining.



Hao Chao https://orcid.org/0000-0001-6700-9446

He received his Ph.D. degree in pattern recognition and intelligent system from Institute of Automation, Chinese Academy of Sciences in 2012. He is currently a lecturer in Henan Polytechnic University. His current research interests include data mining and speech recognition.