International Journal of Automation and Computing 14(1), February 2017, 33-44
DOI: 10.1007/s11633-016-1033-z

A Novel Minkowski-distance-based Consensus Clustering Algorithm

De-Gang Xu Pan-Lei Zhao Chun-Hua Yang Wei-Hua Gui Jian-Jun He

College of Information Science and Engineering, Central South University, Changsha 410083, China

Abstract: Consensus clustering is the problem of coordinating clustering information about the same data set coming from different runs of the same algorithm. Consensus clustering is becoming a state-of-the-art approach in an increasing number of applications. However, determining the optimal cluster number is still an open problem. In this paper, we propose a novel consensus clustering algorithm that is based on the Minkowski distance. Fusing with the Newman greedy algorithm in complex networks, the proposed clustering algorithm can automatically set the number of clusters. It is less sensitive to noise and can integrate solutions from multiple samples of data or attributes for processing data in the processing industry. A numerical simulation is also given to demonstrate the effectiveness of the proposed algorithm. Finally, this consensus clustering algorithm is applied to a froth flotation process.

Keywords: Minkowski distance, consensus clustering, similarity matrix, process data, froth flotation.

1 Introduction

In recent years, with the rapid development of the processing industry, the internet, cloud computing, mobile communication and web of things, data clustering has become an important research field that involves data security, data analysis and data mining. For example, complex systems in industry accumulate data that have the characteristics of enormous quantity, continuous sampling, multiple sources, and sparse values[1−3]. Higher requirements have been proposed for data processing in real-time, while research on fast data processing technologies has encountered many challenges.

Clustering is a very useful technique for mining large data sets because it divides the data into smaller groups that are easier to address. Many different clustering methods have been investigated, including hierarchical agglomerative clustering, graph partitioning, mixture densities, and spectral clustering[4, 5]. Most of the clustering methods focus on finding a single optimal or near-optimal clustering according to some specific clustering criterion[6−10]. Consensus clustering is an important extension of classical clustering[11, 12]. Consensus clustering is used to find a compromise that provides a trade-off among different clustering information about the same data set. However, as one of the effective methods, consensus clustering is less sensitive to noise and can integrate solutions from multiple distributed sources of data or attributes. It solves the problem of reconciling clustering information about the same data set that arises from different sources. Many different approaches have been developed to solve the consensus clustering problem over recent years[7]. In consensus clustering algorithms, pairwise similarity often does not reflect a good measure of similarity between data points[13−15]. Addressing a large number of dimensions and a large number of data items is problematic due to time complexity. The effectiveness of the clustering methods depends on the definition of the similarity distance. In addition, the selection of the similarity matrix and the determination of the number of clusters are very difficult problems, and they must be solved urgently.

Motivated by the above discussion, in this paper, we propose a novel consensus clustering algorithm. Based on the Minkowski distance[16, 17], the proposed algorithm can determine the number of clusters automatically and obtain the clustering results. This approach is also less sensitive to noise and can integrate solutions from multiple-sample data or the attributes of data in the process industry.

2 Consensus clustering problem

2.1 Clustering analysis and clustering problem

Cluster analysis is to divide a set of objects into different groups in such a way that objects in the same group are more similar to each another than to objects in the other groups. Cluster analysis can be performed by various algorithms that can efficiently find different clusters. Clustering can be regarded as a multi-objective optimization problem[18]. The appropriate clustering algorithm and parameter settings (including the distance function or the

Research Article
Special Issue on Emergent Control and Computing Techniques for Industrial Applications
Manuscript received January 16, 2016; accepted May 16, 2016; published online December 29, 2016
This work was supported by National High Technology Research and Development Program (863Program) (No. 2013AA040301-3), National Natural Science Foundation of China (Nos. 61473319 and 61104135), the Key Project of National Natural Science Foundation of China (Nos. 61621062 and 61134006), and the Innovation Research Funds of Central South University (No. 2016CX014).
Recommended by Guest Editor Dongbing Gu

number of expected clusters) are very important, and they must be modified until the result achieves the desired properties.

Clustering analysis is one of the key steps in processing large-scale data problems. Different researchers employ different clustering models and different algorithms. The notion of a cluster, as found by the different algorithms, varies significantly in its properties. Clustering algorithms can be categorized based on their clustering model. In recent years, considerable effort has been made to improve the performance of the clustering algorithms. For larger data sets, the development of pre-clustering methods can process enormous data sets efficiently[9].

For high-dimensional data, many of the existing methods cannot obtain satisfactory results due to the curse of dimensionality. New clustering algorithms that focus on subspace clustering have been proposed by giving a correlation of data attributes[19, 20]. However, most of the clustering methods focus on finding a single optimal or near-optimal clustering according to a specific clustering criterion.

2.2 Consensus clustering algorithm

Consensus clustering is also called an aggregation of clustering, which refers to the method of finding the correct clustering from different clustering results that are obtained for a specific dataset by means of specifying different cluster numbers. In terms of the determination of the number of clusters, the consensus clustering method has shown its own characteristics and it provides an effective method for solving gene microarray data and text data clustering problems[4, 21]. However, for the determination of the number of clusters, there are different standards, and the application fields of the methods are also different. There exist the following two algorithms: one algorithm consists of consensus clustering combined with resampling or cross-validation techniques, and the other algorithm is iterative consensus clustering. The consensus clustering combines resampling or cross-validation techniques to simulate the disturbance of the original data by multiple runs of a clustering algorithm, to obtain stable clustering. This algorithm can provide a visual means of observing the number of clusters, cluster members and cluster boundaries. However, certain problems still remain: Most of the samples used in this algorithm are sampled randomly. We do not know the sampling frequency and the sampling ratio. It should be verified whether the rule of determining the number of clusters can be effective for all of the relevant types of data.

Iterative consensus clustering adopts the basic idea of consensus clustering with resampling or cross-validation[13, 14]. The difference is the method of obtaining the consensus matrix. This algorithm combines several clustering algorithms on the same sample for data clustering, without using resampling or cross-validation techniques. Additionally, it introduces the random walk strategy into the analysis of the consensus matrix and obtains the transition probability matrix. The number of clusters is determined by analyzing the transition probability matrix′s eigenvalues. If the result of the eigenvalues cannot reflect the clustering information, then this algorithm will use the consensus matrix as the similarity matrix for multiple iterations. The overall computing process of this algorithm is shown in Fig. 1.

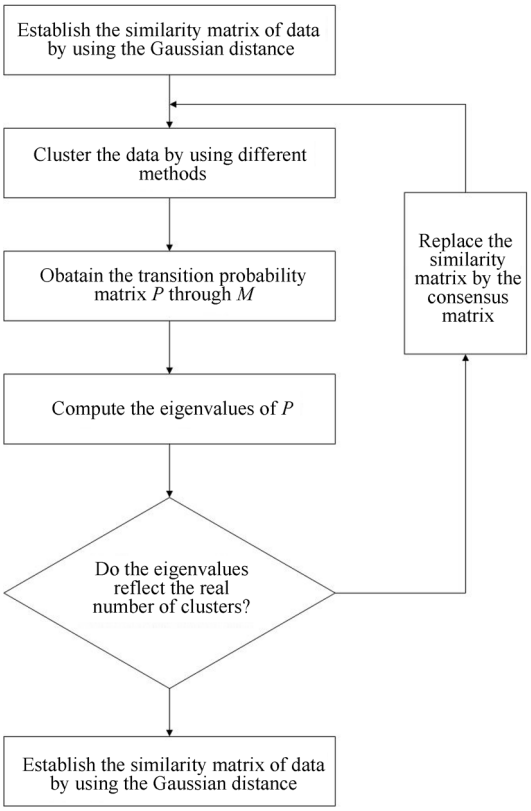

Fig. 1 Flow graph of iterative consensus clustering

Iterative consensus clustering adopts the basic idea of consensus clustering with resampling or cross-validation[13, 14]. The difference is the method of obtaining the consensus matrix. This algorithm combines several clustering algorithms on the same sample for data clustering, without using resampling or cross-validation techniques. Additionally, it introduces the random walk strategy into the analysis of the consensus matrix and obtains the transition probability matrix. The number of clusters is determined by analyzing the transition probability matrix′s eigenvalues. If the result of the eigenvalues cannot reflect the clustering information, then this algorithm will use the consensus matrix as the similarity matrix for multiple iterations. The overall computing process of this algorithm is shown in Fig. 1.

The iterative consensus clustering algorithm also has some problems. For example, it is not very easy to determine the number of iterations and specify the termination condition; the construction of the similarity matrix is relatively simple, but the Gaussian distance formula is not always useful. Combining the advantages of the above two algorithms, in this paper, we propose a new consensus clus-

tering algorithm that is based on the Minkowski distance. This method can quickly acquire the number of clusters accurately without iterating over the data.

3 Consensus clustering numbers based on the Minkowski distance

Compared with the two above consensus clustering algorithms, this novel algorithm differs in the similarity matrix construction. It does not use the resampling technique, but it uses the Minkowski distance to measure the input data. The similarity matrix does not adopt one specific Minkowski distance, but makes use of the different forms of Minkowski distances by adjusting the parameters in the equation. Then, it performs consensus clustering based on the different similarity matrix constructions and obtains different consensus matrices. Finally, it selects the best consensus matrix from the different clustering results. The detailed algorithmic process is described as follows.

3.1 Building the similarity matrix

The similarity matrices are usually constructed by the Euclidean distance or the Gaussian distance. In this paper, we use the Minkowski distance formulas (1) and (2) to construct the similarity matrix.

$$M_p(x,y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{\frac{1}{p}} \tag{1}$$

$$S_{M_p}(x,y) = \mu \exp(-\mu M_p(x,y)). \tag{2}$$

When $p = 1$, equation (1) is the Manhattan distance, which reflects the sum of the absolute value of the difference between data i and data j. When $p = 2$, formula (1) is the Euclidean distance, which reflects the shortest distance between data i and data j, namely, the diagonal distance. If $p \to \infty$, then formula (1) is the Chebyshev distance, which reflects the maximum difference between data i and data j in a certain dimension. Additionally, p could have a value that is less than 1, and the clustering method, by taking different values of p, will yield different results. Here, μ is an adjustable parameter. Because there are two parameters, the distance formula can reflect similar information about the data for different aspects by adjusting the parameters. Although it is unknown which form of the Minkowski distances can describe the similarity of the specific data in advance, we can attempt to use this formula to establish different similarity matrices and obtain the real similarity information selected from different clustering effects. This method is suitable for the idea of consensus clustering, namely, selecting the appropriate matrix from the different results. For the p value, we can separately sample from $\{1, 2, 3\}$ and select μ values from $\{0.1, 0.2, 0.5, 0.8, 0.9\}$. Thus, we can obtain 15 different similarity matrices. Experiments in the following have shown that the 15 different similarity matrices are comprehensive for obtaining the clustering information from the real data.

3.2 Fusion of clustering algorithms

The used clustering algorithms are spectral clustering algorithms[11], which are constructed by two different Laplacian matrices and the improved Newman greedy algorithm in complex network theory[12, 22]. Because this algorithm does not require a determination of the number of clusters, the improved Newman greedy algorithm is set to terminate when the cluster arrives at a specific number. The above three algorithms have their own characteristics. The first two algorithms are similar: Their main idea is spectral clustering, and their difference is the construction of a Laplacian matrix. The third algorithm is the improved Newman greedy algorithm in complex networks, which has the advantage of having a fast convergence speed. These distances can be described by the formulas (3) and (4).

$$L_{sym} = D^{-\frac{1}{2}} L D^{\frac{1}{2}} \tag{3}$$

$$L_{rm} = D^{-1} L \tag{4}$$

where D is a diagonal matrix whose diagonal elements are the sum of the corresponding row of the similarity matrix, the first matrix by L_{sym} is a symmetric matrix, the second matrix by L_{rw} is closely related to a random walk and L is the similarity matrix.

3.3 Description of the number of clusters

1) Mathematical representation of the number of clusters

Any similarity matrix can be regarded as an adjacency matrix that involves the nodes of an undirected graph: The number of samples can be considered to be the number of nodes, and the weights of the similarity matrix can be regarded as the edges between the nodes. We can take advantage of the strengths of the edges to represent the values of the weights. We introduce the random walk strategy to an undirected graph and obtain the transition probability matrix U, where $U = D^{-1}S$, and S is the similarity matrix. $D = \mathrm{diag}\{Sv\}$, where v is a vector whose elements are all 1. We let $\lambda(U) = \{1 = \lambda_1 \geq \lambda_1 \geq \cdots \geq \lambda_n\}$, which is the spectral distribution of P (also called the eigenvalue distribution). It has been proven that the first k eigenvalues will be close to 1 $\{1 = \lambda_1 \geq \lambda_1 \geq \cdots \geq \lambda_n\}$. The relative spacing between the eigenvalues λ_k and λ_{k+1} can determine the number of data clusters, which is the mathematical basis for the representation of the number of clusters.

2) Consensus similarity matrix

First, we determine the sequence of the selected number of clusters, $\pi = [\pi_1, \pi_2, \cdots, \pi_n]$, where n is the number of all selected categories. We use the above three clustering algorithms to cluster the data, and we obtain $3 \times n$ results of clustering. Second, we construct the consensus clustering matrix M. (If the i-th and j-th nodes are assigned to the same class, then M_{ij} is 1, otherwise, it is 0.) Finally, we use the consensus similarity matrix M instead of the similarity matrix S to acquire the transition probability matrix U and obtain the eigenvalue distribution.

As an illustrative simple example, we divide eleven data points (1–11) into five classes with 2 different algorithms. The result is shown in Figs. 2 and 3, where the real clustering is that the points (1–4) are in the same class, and the points (5–9) are in the same class. Different classes are shown in different colors. The consensus similarity matrix that was constructed by two different dividing algorithms is shown in Table 1. The eigenvalue distributions of the consensus similarity matrix are shown in Fig. 4, which shows that there is a gap between the 4th eigenvalue and the first 3. Thus, this result can correctly reflect the number of clusters.

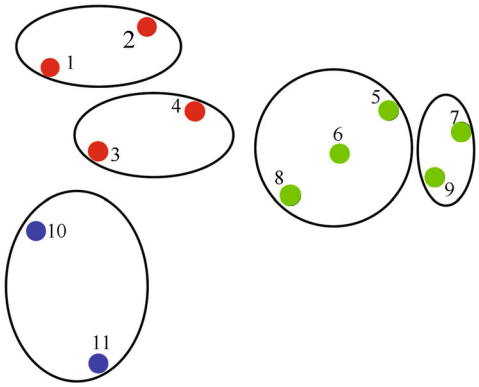

Fig. 2 Clustering algorithm 1

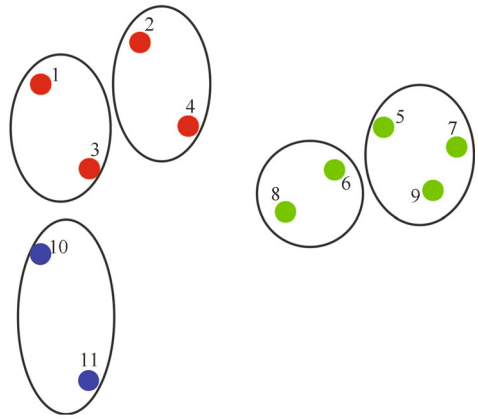

Fig. 3 Clustering algorithm 2

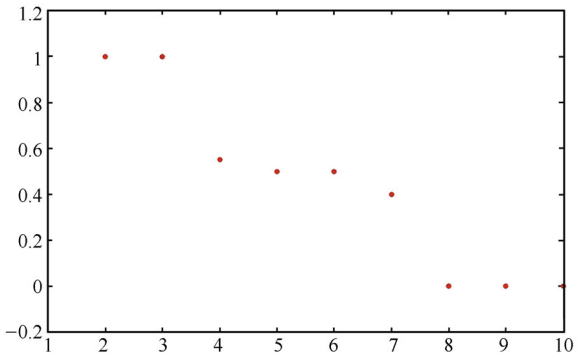

Fig. 4 Eigenvalue distributions

Table 1 Consensus similarity matrix

M	1	2	3	4	5	6	7	8	9	10	11
1	2	1	1	0							
2	1	2	0	1							
3	1	0	2	1							
4	0	1	1	2							
5					2	1	1	1	1		
6					1	2	0	2	0		
7					1	0	2	0	2		
8					1	2	0	2	0		
9					1	0	2	0	2		
10										2	2
11										2	2

4 Consensus clustering algorithm based on the Minkowski distance

4.1 Consensus clustering algorithm

The consensus clustering algorithm based on the Minkowski distance to determine the number of clusters is shown in the following.

Algorithm 1. Obtaining the number of clusters:

Step 1. Establish the different similarity matrices based on the Minkowski distance with different parameters (p and μ).

Step 2. Cluster one of the above similarity matrices with the three clustering algorithms and the clustering numbers (n numbers), and obtain the $3 \times n$ matrices.

Step 3. Obtain the consensus similarity matrix M by analyzing the $3 \times n$ matrices (which reflect the information of the clustering).

Step 4. Obtain the transition probability matrix U from the consensus matrix M.

Step 5. Acquire the transition probability matrices that correspond to the other similarity matrices by using the above methods.

Step 6. Obtain the number of clusters by analyzing the eigenvalues of the different transition probability matrices.

The detailed steps are as follows:

1) Establish the similarity matrix S of the samples using the Minkowski distance. To cover the values of the parameters as much as possible, we set $p \in [1, 2, 3]$ and $\mu \in [0.1, 0.2, 0.5, 0.8, 0.9]$. There is a total of 15 types of situations with respect to the similarity information.

2) For each group p and μ, we set $\pi \in [\pi_1, \pi_2, \cdots, \pi_n]$. For each value π, we use the above three types of clustering algorithms to cluster the similarity matrix and obtain $3 \times n$ clustering results, and then, we obtain the consensus similarity matrices $M_i (i \in [1, 2, \cdots, 15])$.

3) Reset the values for p and μ, and repeat the second step to obtain 15 similarity matrices $M = [M_1, M_2, \cdots M_{15}]$, and then, we obtain the corresponding transition probability matrix using the method described above, $M = [M_1, M_2, \cdots, M_{15}]$.

4) Obtain the eigenvalues of the transition probability

matrices $U_i(i \in [1, 2, \cdots, 15])$, and select the eigenvalue distribution that can reflect the number of clusters best according to certain discrimination rules (the sum of the difference between the adjacency eigenvalues except for the gap eigenvalues being the least), and ultimately obtain the number of clusters from the distribution.

The proposed algorithm has been analyzed by a simulation for four types of data (Table 2), where aggregation and R15 are the graphed data, and breast and yeast are selected from the data at the UCI repository. The clustering number is shown in Fig 5.

Table 2 Four types of typical data

Name	Number of objects	Dimension	Number of clusters
Aggregation	788	2	7
R15	600	2	15
Breast	699	9	2
Yeast	1 484	8	10

Remark 1. When the number of clusters is large due to the fusing of the different clustering algorithms, different algorithms usually do not produce the same error. Therefore, we introduce a parameter δ, and when $M_{ij} < 3 \times 8 \times \delta$, $M_{ij} = 0$, where the constant 3 is the number of clustering algorithms that we used, and the constant 8 is the total number of selected clusters.

From the above graphs, we can obtain the eigenvalue information, which can reflect the correct number of clusters.

4.2 Analysis by the consensus clustering algorithm

In combination with building the similarity matrix and the algorithm for determining the number of clusters, the process of the consensus clustering algorithms combined with multiple clustering algorithms based on Minkowski distances is shown in the following.

Algorithm 2. The proposed consensus clustering algorithm:

1) Data preprocessing.

2) Establish the different similarity matrices based on the Minkowski distance with different parameters (p and μ).

3) Determine the clustering number by using the method introduced in Part 3.

4) Obtain the final clustering by using the clustering number and the consensus matrix (or the similarity matrices by using the local scale, nearest-correlation, or Minkowski distance).

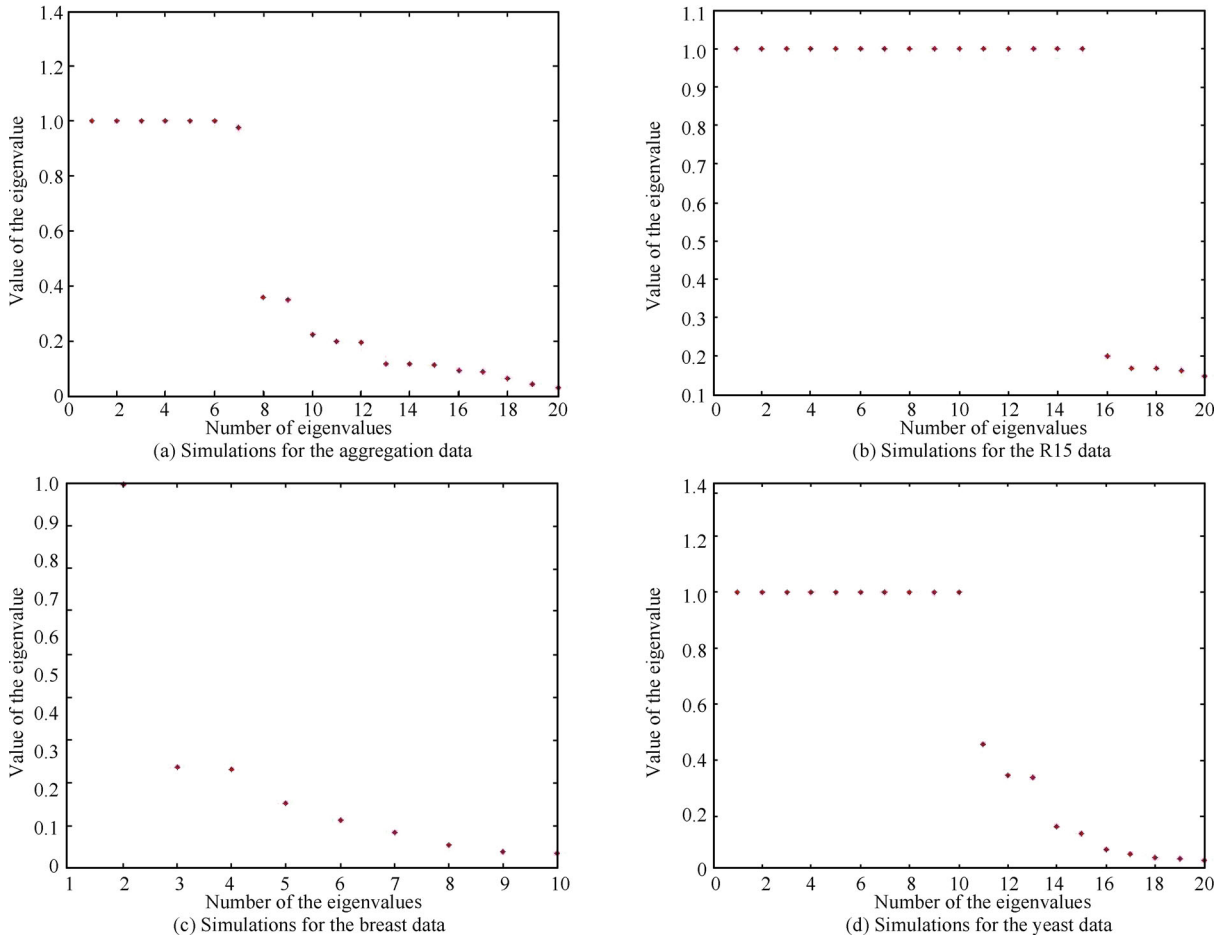

Fig. 5 Simulation results of the aggregation, R15, breast and yeast data

The preprocessing includes data normalization, principal component analysis or singular value decomposition (dimensionality reduction), which are used to extract the main ingredient information. To construct the final similarity matrix, the local scale method[6] and the nearest-correlation (NC) are referred to [7, 8]. When the cluster number has been determined, we can use the following three methods to establish the similarity matrix: 1) Use the consensus matrix obtained in the previous steps as the similarity matrix. 2) Use the similarity matrix established by the Minkowski distance that corresponds to the cluster number k. 3) Use the similarity matrix established by the local scale or nearest-correlation distance. We can flexibly use different methods according to different data features to obtain the corresponding accurate data clustering information. The accuracy of these methods is shown in Section 3.3.

Next, we analyze the computational complexity of the proposed algorithm. Because the major purpose of determining the clustering number is to reduce the computational complexity, it is of interest to see, both theoretically and in practice, how dramatic the improvement could be. Suppose that we have n partitions for each group p and μ. Then, every iteration of the algorithm requires $(n \times p \times \mu)$ evaluations. Suppose that the algorithm takes h steps to converge; then, its computational complexity is $O(p \times m \times n \times h)$. The following are results that arise from simulations for the data (in Table 3) using the proposed algorithm.

Table 3 UCI data, graphic data and artificial random data

Name	Number of samples	Dimension	Number of clusters
Random five	100	2	5
Flame figure	240	2	2
Iris data	150	4	3
Wine data	178	13	3

The test using typical data includes the data sets of Random five clustering data, Flame picture data, Iris data and Wine data. The simulation results are shown in Figs. 6−9. The data in Table 4 shows a small number of samples, and the number of samples in the data in Table 5 is relatively large. The aggregation, R15 and D31 are from the UCI data repository, and the random 4 clusters data are artificial data (the ranges of the values include [−1 −1], [2 2], [−3 3], [−4 −4], the corresponding variances are 0.5, 0.7, 0.1 and 0.6, and the number of samples is 8 000, which contains 4 clusters with 2 000 samples for each). The random 5 clusters data are also artificial data (the ranges of the values include [1 1], [1 6], [6 1], [6 6], [3.5 3.5], the corresponding variances are the same (the value is 0.1), and the number of samples is 10 000, which contains 5 clusters with 2 000 samples in each).

For clustering large-scale data, the algorithm called fast spectral clustering with k-means is used. This algorithm makes use of the k-means method to obtain the initial clustering, which can yield some clustering central points and distributes the original data by calculating the shortest distance from these central points. Then, it clusters the central points with the spectral clustering algorithm and redistributes the original data by means of the final clustering central points. The process is shown in the following.

Algorithm 3. Fast spectral clustering with k-means:

Step 1. Cluster the data with k-means and obtain k_1 clusters.

Step 2. Calculate the k clustering centers and obtain k_1 representative points.

Table 4 Data, including the UCI data and artificial data

Name	Samples	Dimension	Number of clusters
Aggregation	788	2	7
R15	600	2	15
D31	3 100	2	31
Rand 4 clusters	8 000	2	4
Rand 5 clusters	10 000	2	5

Table 5 Simulation results of the rand index

Name	Nearest-correlation	Local scale	Minkowski
Aggregation	0.866 3	0.812 8	0.757 6
R15	0.536 1	0.992 8	0.899 3
D31	0.527 2	0.857 2	0.803 3
Rand 4 clusters	0.992 0	0.990 4	0.990 7
Rand 5 clusters	1	1	1

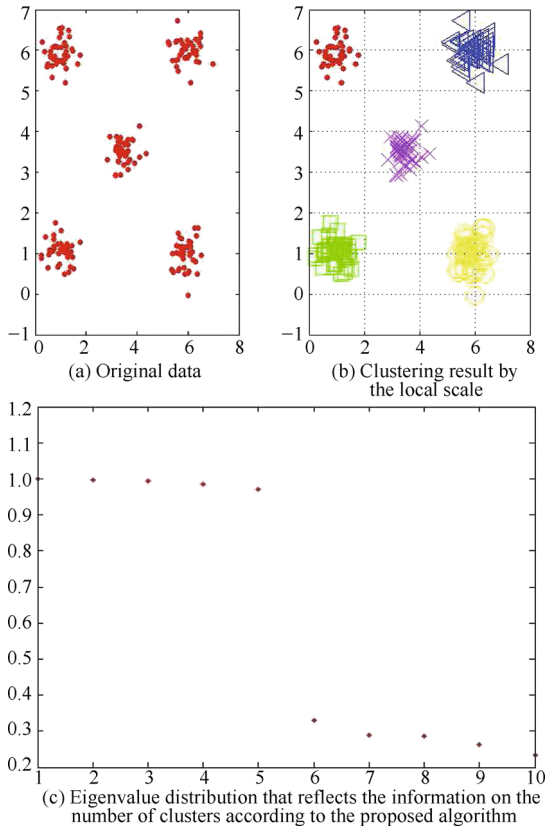

Fig. 6 Simulation results of the random five clustering data by the proposed algorithm

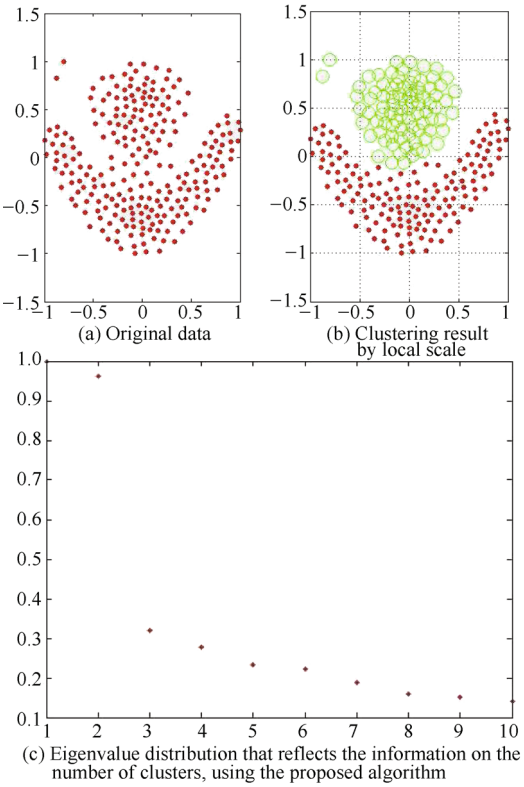

Fig. 7 Simulation results of the Flame figure data by the proposed algorithm

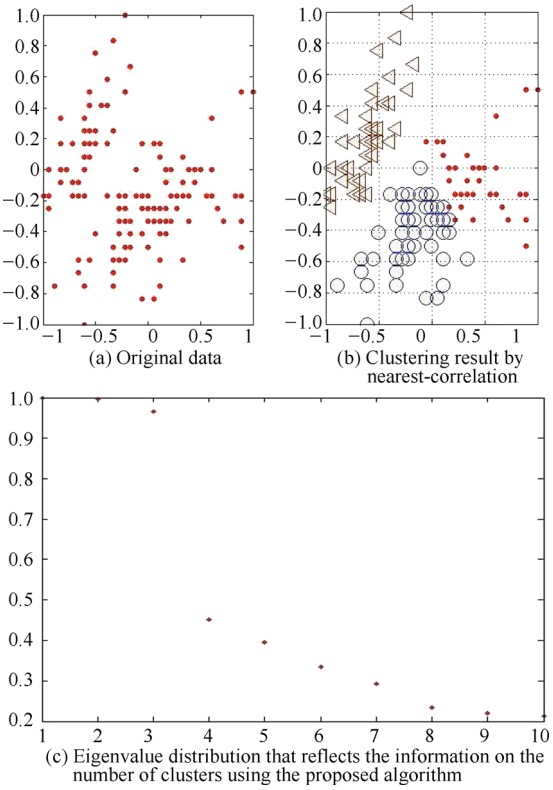

Fig. 8 Simulation results on the Iris data by the proposed algorithm

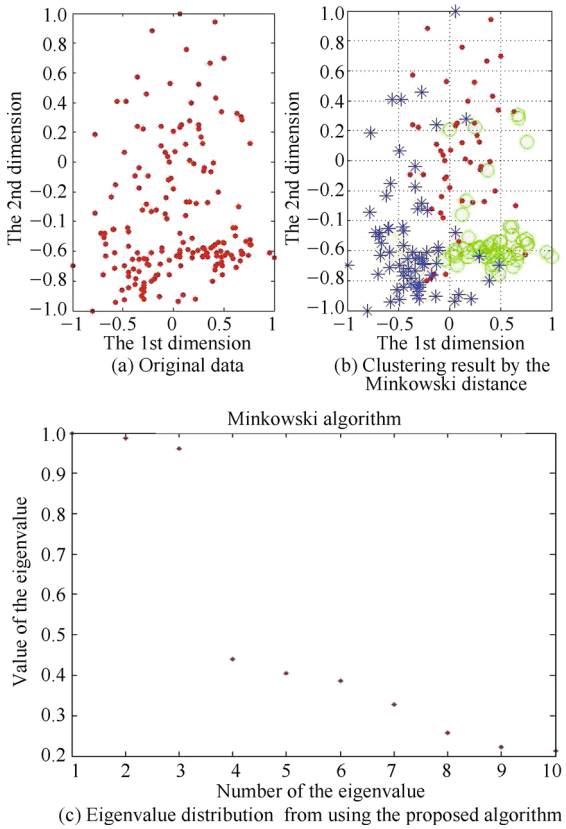

Fig. 9 Simulation results on the wine data by the proposed algorithm

Step 3. Distribute the original data by the shortest distance from the k_1 clustering centers.

Step 4. Analyze the k clustering centers by spectral clustering and obtain k_2 clusters ($k_2 < k_1$).

Step 5. Redistribute the original data by the shortest distance from the k_2 clustering centers.

The fast spectral clustering with k-means has the advantage of completing the calculations quickly, which is reflected in the initial clustering of the k-means and the analysis of the central points by the use of spectral clustering. Thus, the calculation′s complexity is decreased compared with the method of clustering the original data using the spectral clustering only. However, clustering by using the k-means method has the shortcoming of converging to a local optimum, and thus, it is sometimes unstable. The present paper provides one stable and accurate consensus clustering method, for which the accuracy of the simulation results is better than the above algorithm. The process is shown in the following.

Algorithm 4. Consensus clustering based on the Minkowski distance:

Step 1. Select n samples each time for H times.

Step 2. Analyze the selected data every time by means of the consensus clustering combined with the Minkowski distance.

Step 3. Obtain the H transition probability matrix U and analyze the corresponding eigenvalues.

Step 4. Choose the selected data that reflects the clustering number k and obtain k clustering centers using spectral clustering.

Step 5. Distribute the original data to the corresponding cluster by calculating the shortest distance from the k clustering centers.

The accuracy of the proposed algorithm is better than that of fast spectral clustering with k-means because the provided consensus clustering algorithm can find more information on the number of clusters number than k-means.

The rand index is used to measure and compare the accuracy of the two methods. The value of the rand index is between 0 and 1. If the value is closer to 1, then the result is more accurate. The simulation results are shown in Tables 5 and 6.

Table 6 Simulation results on the rand index for the proposed algorithm

	Number from k in the k-means				
Name	100	200	300	500	1 000
Aggregation	0.401 1	0.267 3	0.306 3	0.373 5	
R15	0.440 4	0.568 0	0.676 4	0.823 3	
D31	0.028 7	0.103 2	0.163 3	0.397 7	0.448 7
Rand 4 clusters	0.491 2	0.680 8	0.474 0	0.344 3	0.431 0
Rand 5 clusters	0.402 8	0.303 8	0.399 3	0.300 9	0.309 1

By comparing the values of the rand index, the algorithm in the paper is better than the fast spectral clustering with the k-means method. The value of the rand index obtained by the latter method can increase as the value of k increases, but the operation time also increases.

4.3 Clustering evaluation criteria

To evaluate the consensus clustering method, the paper adopts two evaluation criteria to measure the results of the algorithm, including the compactness and accuracy. The compactness (formula (5)) measures the average pairwise distances between the points in the same cluster.

$$compactness = \frac{1}{N}\sum_{k=1}^{K} n_k \left(\frac{\sum\limits_{x_i, x_j \in C_k} d\left(x_i, x_j\right)}{\frac{n_k\left(n_k-1\right)}{2}} \right) \tag{5}$$

where $d(x_i, x_j)$ is the distance between x_i and x_j, N indicates the number of data points in the cluster, and n_k is the number of data points in the class C_k.

$$accuracy = \frac{\sum\limits_{k=1}^{K} majority\left(C_k | L_k\right)}{N}. \tag{6}$$

The accuracy (formula (6)) measures the consistency between the clustering results of the algorithm and the actual results, in other words, the veracity. In the formula, L_k is the k-th class of the actual classes, $majority(C_k\,|L_k\,)$ is the number of points that have the plurality label in the C_k cluster (If label l appeared in cluster k more often than any other label, then $majority(C_k\,|L_k\,)$ is the number of points in C_k with label l).

As an example application of our clustering method, the true clustering numbers of the random 5 classes of data, flame figure data, iris data and wine data are given by using the consensus clustering algorithm based on the Minkowski distance. The final clustering result is obtained by constructing the similarity matrices using the consensus matrix, Minkowski distance, local-scale and nearest-correlation, and the evaluation of the two clustering criteria is given in Table 7.

Table 7 Accuracy results of the different clustering algorithms

Methods of constructing the similarity matrix	Simulation data	Compactness	Accuracy
Consensus matrix	Random five clusters	0.002 4	1
	Flame figure	0.015 3	0.804 1
	Iris	0.070 3	<0.5
	Wine	0.013 3	0.921 3
Minkowski distance	Random five clusters	0.002 5	1
	Flame figure	0.015 2	0.833
	Iris	0.074 7	<0.5
	Wine	0.008 3	0.966 2
Local scale	Random five clusters	0.092 6	1
	Flame figure	1.571 9	0.987 5
	Iris	2.473 9	0.546 7
	Wine	2.651 8	0.949 4
Nearest-correlation	Random five clusters	0.002 3	1
	Flame figure	0.013 6	0.966 7
	Iris	0.047 3	0.94
	Wine	0.016 5	0.904 4

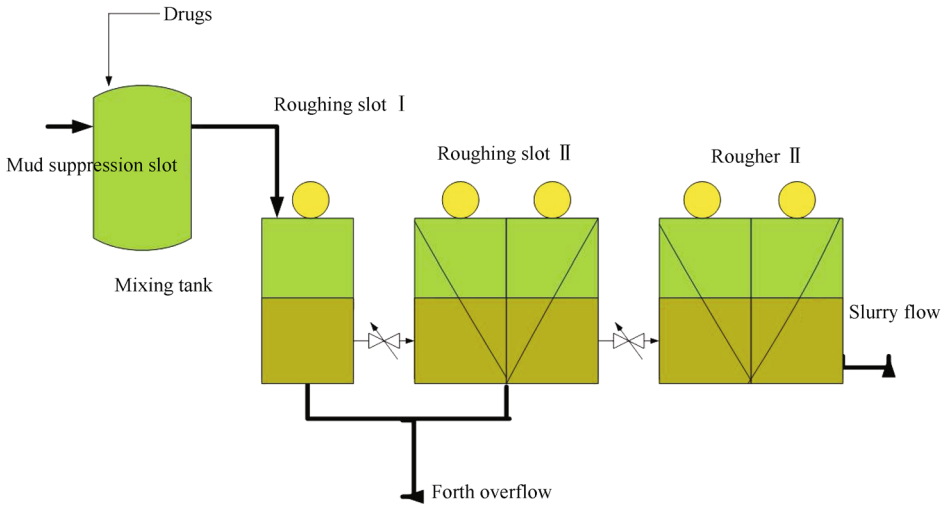

Fig. 10 Copper froth flotation process

Table 7 shows that there are different clustering results. For the simplest data (random five clusters of data), the clustering accuracy is 1 for the four similarity matrices of the methods. For the flame figure data, the method with the similarity matrix has the best clustering accuracy. For the iris data, the result is best for the method based on the similarity matrix constructed by nearest-correlation. For the wine data, the method with the similarity matrix constructed by the Minkowski distance is more accurate. The compactness reflects the average pairwise distances between the points in the same cluster. The values of the compactness are different for the different similarity matrices, and the value of the compactness reflects the compactness degree for the data points in the same cluster. A smaller value reflects a more compact degree, and a larger value reflects a sparser degree. For the test clusters data, the compactness of the algorithm using local scale is larger than that of the other methods. From the comparison, we observe that the consensus clustering algorithms usually obtain better clustering results on all of the datasets. However, different methods for constructing the similarity matrix have different performances for the different datasets. Thus, choosing a good similarity matrix is important, and further research will be focused in this direction. However, the proposed method can still be considered to be a powerful tool that can produce good results.

5 Application to the froth flotation process

In this section, the proposed consensus clustering is applied to the mineral flotation process that is used to monitor the product composition (ore grade). Froth flotation is widely used in mineral processing plants. Froth flotation utilizes the differences in the physicochemical surface properties of various minerals to achieve specific separation. Hydrophobic particles attached to air bubbles are transported upward into a froth layer at the top of the flotation cell, while the hydrophilic particles remain in the slurry that forms the tail. Flotation plant operators usually have heuristic rules with regard to the visual appearances of froth and corresponding corrective operating actions because it is a well-known fact that the performance of the flotation circuits is related to the froth visual characteristics[23, 24]. It has been proven that froth appearances are essentially process outputs that respond to process inputs such as reagent flow rates[22−25].

Recently, some methods have been developed for the modeling and control of flotation processes based on the froth features and corresponding process data. The typically detailed copper froth flotation process used in this case study is described in Fig. 10. In this process, after grinding first, the slurry flow passes through the mud suppression slot, followed by the mixing tank, and then, it flows through the roughing of the first slot (slot I) and the rougher II.

In practical industrial applications, enormous volumes of flotation froth visual feature data can be obtained by an image collecting system, such as data on the froth color, size, texture, and speed. We utilize online sensors to obtain a large amount of process operating data, which form dynamic process data. How to process these data in time and extract useful information to reflect the froth flotation state is a challenging problem. At present, many researchers adapt artificial neural networks to process the data. However, these algorithms rely heavily on the choice of training samples, which are very sensitive to the initial weight and can easily converge to a local minimum. Moreover, the presence of problems such as over-fitting, over-training and other factors leads to errors and poor capacity for improvement. To avoid these problems, in this paper, we analyze the correlation between the data from the viewpoint of consensus clustering. Then, we can obtain the status information of the flotation process on the basis of the clustering results, which can be used to identify and monitor the production of the froth flotation.

In this paper, we analyze real data from the above-

mentioned copper froth flotation from May 24, 2011 to May 26, 2011, which includes a total of 1056 sets of data (collected every 5 minutes). Each set of data is 13-dimensional feature data that includes the bubble speed, stability, gray value, and load factors. After addressing the noise in the data, we select representative data every 20 minutes and obtain 264 sets of data. According to the proposed consensus clustering algorithm, the number of clusters is shown in Fig. 11. Table 8 is the mean grade of the copper during the morning shift, middle shift and night shift on May 24−26, 2013. Every shift lasts eight working hours.

Table 8 Different copper froth grades for three shifts

	Time	Morning shift	Middle shift	Night shift
Copper	5-24	15.72	16.78	14.97
Froth	5-25	14.06	17.38	15.47
Grade	5-26	14.06	17.38	15.47

It can be determined from Fig. 11 that the 264 sets of sample data can be divided into 3 clusters: Cluster 1 (which includes the production data derived from 0:00 AM on May 24th to 9:00 AM on May 25th and 4 AM to 10 AM on the 26th). Cluster 2 (which includes the production data from 10 AM on May 25th to 4 AM on May 26th). And Cluster 3 (which includes the production data from 10 AM on 26th to 12 PM on 26th). According to the copper grade, it is obvious that there are differences between the three production operation shifts. In fact, the clustering results from the data can reflect the correlation for the sample data, which is shown in Fig. 12.

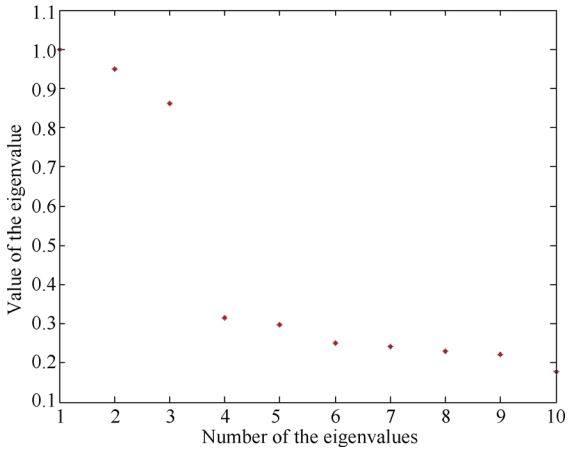

Fig. 11 Eigenvalue of the Minkowski matrix

6 Conclusions

Consensus clustering can solve the problem of reconciling clustering information about the same data set that arises from different runs of the same algorithm. Then, it can find a single consensus clustering that is better than the existing clusters. In this paper, we propose a novel consensus clustering algorithm that considers the consensus clustering partition distance and similarity matrix. Based on the Minkowski distance, the proposed clustering algorithm can automatically set the number of clusters and obtain better clustering results, which can find a compromise in the different clustering information about the same data set. Numerical simulation results are provided to demonstrate the effectiveness of the presented algorithm. This real application also verifies the effectiveness of the proposed consensus clustering algorithm.

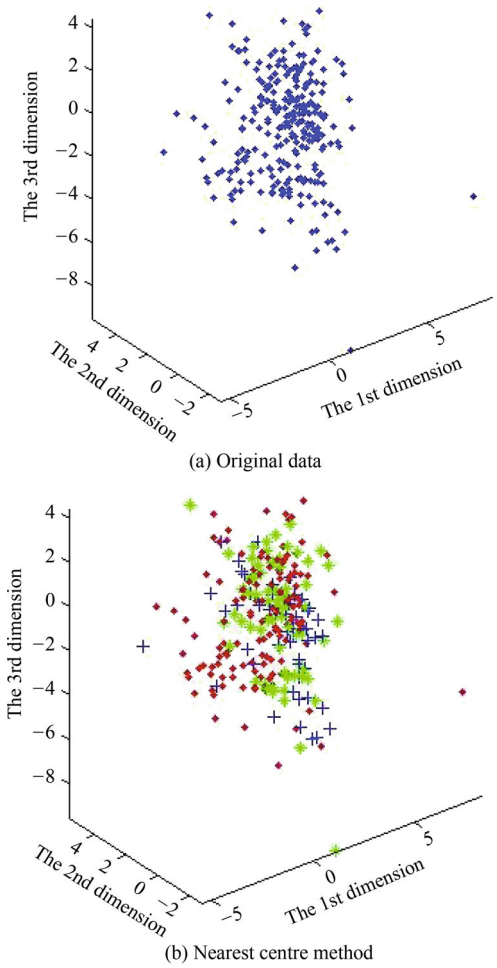

Fig. 12 Clustering results for the sample data (Different colors represent different clusters)

References

[1] A. L. Barabási, R. Albert. Emergence of scaling in random networks. *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[2] V. Mayer-Schönberger, K. Cukier. *Big Data: A Revolution that will Transform How We Live, Work, and Think*, Boston, USA: Houghton Mifflin Harcourt, 2013.

[3] V. R. Radhakrishnan, A. R. Mohamed. Neural networks for the identification and control of blast furnace hot metal quality. *Journal of Process Control*, vol. 10, no. 6, pp. 509–524, 2000.

[4] S. Monti, P. Tamayo, J. Mesirov, T. Golub. Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. *Machine Learning*, vol. 52, no. 1–2, pp. 91–118, 2003.

[5] S. Race, C. Meyer, K. Valakuzhy. Determining the number of clusters via iterative consensus clustering. arXiv: 1408.0967, 2014.

[6] M. E. J. Newman. Fast algorithm for detecting community structure in networks. *Physical Review E*, vol. 69, no. 6, pp. 066133, 2004.

[7] P. Deuflhard, W. Huisinga, A. Fischer, C. Schütte. Identification of almost invariant aggregates in reversible nearly uncoupled Markov chains. *Linear Algebra and its Applications*, vol. 315, no. 1–3, pp. 39–59, 2000.

[8] W. J. Stewart. *Probability, Markov Chains, Queues, and Simulation: The Mathematical Basis of Performance Modeling*, Princeton, USA: Princeton University Press, 2009.

[9] K. Fujiwara, M. Kano, S. Hasebe. Development of correlation-based clustering method and its application to software sensing. *Chemometrics & Intelligent Laboratory Systems*, vol. 101, no. 2, pp. 130–138, 2010.

[10] K. Fujiwara, M. Kano, S. Hasebe. Correlation-based spectral clustering for flexible process monitoring. *Journal of Process Control*, vol. 21, no. 10, pp. 1438–1448, 2011.

[11] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.

[12] B. Yang, D. Y. Liu, J. M. Liu, D. Jin, H. B. Ma. Complex network clustering algorithms. *Journal of Software*, vol. 20, no. 1, pp. 54–66, 2009. (in Chinese)

[13] A. Strehl, J. Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *The Journal of Machine Learning Research*, vol. 3, pp. 583–617, 2003.

[14] T. Li, C. Ding, M. I. Jordan. Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization. In *Proceedings of the 7th IEEE International Conference on Data Mining*, IEEE, Omaha, USA, pp. 577–582, 2007.

[15] X. H. Hu, I. Yoo, X. D. Zhang, P. Nanavati, D. Das. Wavelet transformation and cluster ensemble for gene expression analysis. *International Journal of Bioinformatics Research and Applications*, vol. 1, no. 4, pp. 447–460, 2005.

[16] R. C. De Amorim, B. Mirkin. Minkowski metric, feature weighting and anomalous cluster initializing in K-Means clustering. *Pattern Recognition*, vol. 45, no. 3, pp. 1061–1075, 2012.

[17] R. C. De Amorim, C. Hennig. Recovering the number of clusters in data sets with noise features using feature rescaling factors. *Information Sciences*, vol. 324, pp. 126–145, 2015.

[18] B. S. Everitt, S. Landau, M. Leese, D. Stahl. *Cluster Analysis*, 5th ed., UK: Wiley, 2011.

[19] B. Auffarth. Clustering by a genetic algorithm with biased mutation operator. In *Proceedings of the IEEE Congress on Evolutionary Computation*, IEEE, Barcelona, Spain, pp. 1–8, 2010.

[20] B. J. Frey, D. Dueck. Clustering by passing messages between data points. *Science*, vol. 315, no. 5814, pp. 972–976, 2007.

[21] H. G. Ayad, M. S. Kamel. On voting-based consensus of cluster ensembles. *Pattern Recognition*, vol. 43, no. 5, pp. 1943–1953, 2010.

[22] M. E. J. Newman. Detecting community structure in networks. *The European Physical Journal B*, vol. 38, no. 2, pp. 321–330, 2004.

[23] D. G. Xu, X. Chen, Y. F. Xie, C. H. Yang, W. H. Gui. Complex networks-based texture extraction and classification method for mineral flotation froth images. *Minerals Engineering*, vol. 83, pp. 105–116, 2015.

[24] J. Zhang, Z. H. Tang, J. P. Liu, Z. Tan, P. F. Xu. Recognition of flotation working conditions through froth image statistical modeling for performance monitoring. *Minerals Engineering*, vol. 86, pp. 116–129, 2016.

[25] N. Barbian, J. J. Cilliers, S. H. Morar, D. J. Bradshaw. Froth imaging, air recovery and bubble loading to describe flotation bank performance. *International Journal of Mineral Processing*, vol. 84, no. 1–4, pp. 81–88, 2007.

De-Gang Xu received the Ph. D. degree in control science and engineering from Zhejiang University, China in 2007. He is currently a professor with College of Information Science and Engineering, Central South University, China.

His research interests include intelligent control, process control, machine learning and computation algorithms.

E-mail: dgxu@csu.edu.cn (Corresponding author).
ORCID iD: 0000-0003-1730-9410

Pan-Lei Zhao received the M. Sc. degree in control science and engineering from Central South University, China in 2014. He is currently an engineer with China Railway Rolling Stock Corporation Limited.

His research interests include intelligent control, process control and computation algorithms.

E-mail: 876377835@qq.com

Chun-Hua Yang received the Ph. D. degree in control science and engineering from Zhejiang University, China in 2002. She is currently a professor with College of Information Science and Engineering, Central South University, China.

Her research interests include intelligent control, process control, machine learning and dispatching control system.

E-mail: ychh@csu.edu.cn

Wei-Hua Gui received the M. Sc. degree in control science and engineering from Zhejiang University, China in 1984. He is currently a professor with College of Information Science and Engineering, Central South University, China.

His research interests include large-scale control, process control and computer control system.

E-mail: gwh@csu.edu.cn

Jian-Jun He received the Ph. D. degree in control science and engineering from Zhejiang University, China in 2003. He is currently a professor with the School of Information Science and Engineering.

His research interests include large-scale control, process control and computer control system.

E-mail: jjhe@csu.edu.cn