Study on multi-center fuzzy C-means algorithm based on transitive closure and spectral clustering

Shan Zeng a, *, Xiaojun Tong b, Nong Sang c

a College of Mathematics and Computer Science, Wuhan Polytechnic University, Wuhan, Hubei 430023, China
b College of Mathematics and Computer Science, Wuhan Textile University, Wuhan, Hubei 430077, China
c Institute for Pattern Recognition and Artificial Intelligence, Huazhong University of Science and Technology, Wuhan, Hubei 430074, China

A R T I C L E   I N F O

Article history:
Received 31 December 2012
Received in revised form 10 October 2013
Accepted 25 November 2013
Available online 12 December 2013

Keywords:
Fuzzy C-means algorithm
Multi-center
Lattice similarity
Spectral clustering

A B S T R A C T

Fuzzy C-means (FCM) clustering has been widely used successfully in many real-world applications. However, the FCM algorithm is sensitive to the initial prototypes, and it cannot handle non-traditional curved clusters. In this paper, a multi-center fuzzy C-means algorithm based on transitive closure and spectral clustering (MFCM-TSCS) is provided. In this algorithm, the initial guesses of the locations of the cluster centers or the membership values are not necessary. Multi-centers are adopted to represent the non-spherical shape of clusters. Thus, the clustering algorithm with multi-center clusters can handle non-traditional curved clusters. The novel algorithm contains three phases. First, the dataset is partitioned into some subclusters by FCM algorithm with multi-centers. Then, the subclusters are merged by spectral clustering. Finally, based on these two clustering results, the final results are obtained. When merging subclusters, we adopt the lattice similarity method as the distance between two subclusters, which has explicit form when we use the fuzzy membership values of subclusters as the features. Experimental results on two artificial datasets, UCI dataset and real image segmentation show that the proposed method outperforms traditional FCM algorithm and spectral clustering obviously in efficiency and robustness.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

Clustering plays an important role in pattern recognition, image processing, and computer vision [1]. With a clustering technique, a collection of objects or feature vectors is partitioned into clusters. In the past few decades, many clustering algorithms have been developed, which mainly contain hierarchical clustering (such as Single Link and Complete Link), partitional clustering (such as k-means, fuzzy C-means, Gaussian Mixture and Density Estimation) and spectral clustering. Moreover, in the last few years, fuzzy C-means (FCM) clustering and spectral clustering algorithms are research focus [2,3].

Unlike the hard clustering techniques (each object is assigned to one and only one cluster), fuzzy C-means clustering allows an object to belong to a cluster with a grade of membership. Moreover, when there is not enough information about the structure of the data, fuzzy C-means clustering algorithm can handle this uncertainly better, and has been widely applied to the data clustering area. However, there are still some open problems in FCM algorithm [4–6]. (1) The number of clusters needs to be known in advance. But we have to cluster objects without knowing the number of clusters in a lot of cases. (2) It can only recognize spherical clusters, but not irregular or elongated shaped clusters. (3) It is easy to fall into local optimum in the iterative process. Spectral clustering algorithm (SC) transforms the original dataset into a new one in a lower-dimensional eigenspace by utilizing eigenvalues and eigenvectors of a similarity matrix derived from the dataset. Then the traditional clustering algorithms can be performed on this new dataset to obtain the final clustering result. In spite of these successes in spectral clustering algorithm, there are still some problems urgently needing to be solved, which are as follows [7–9]: (1) The construction of similarity matrix. Up to now, there is not a method of wide application or high validity. (2) There is not a theory which can solve the selection (including how many and what kind of eigenvectors are needed as spectral features) and the usage of the eigenvectors. (3) The high load of computation makes the spectral clustering act poorly in large-scaled learning.

To solve the problem of partitional clustering algorithm that one center of each class in the irregular shaped datasets, such as non-convex and elongated shaped classes, cannot represent the class well in clustering, the multi-center clustering algorithm [10] proposed by Chaudhuri and Chaudhuri. Based on the multi-center clustering algorithm, to solve the problems of FCM algorithm, an unsupervised fuzzy clustering with multi-centers algorithm [11] is provided by Tao. In the multi-center clustering algorithm, the most widely used method in merging subclusters is DBSCAN algorithm.
[12] based on density (proposed by Martin Ester). The DBSCAN algorithm considers that a class contains the samples which has more than a specified threshold number of samples in a specified neighborhood. The class is constructed by finding a maximum set of the connected samples which satisfy the density (samples in the neighborhood) condition. That is, DBSCAN expands the class in a point-wise manner. Although DBSCAN can find the classes with arbitrary shape, a priori information (specified neighborhood and density threshold) is necessary, and the errors are high when the samples' distribution density is unbalanced.

To achieve a successful clustering without the subclusters-merging difficulty by the multi-center fuzzy C-means clustering algorithm, the spectral clustering algorithm is introduced in this paper. Spectral clustering algorithm has problems which are the construction of similarity matrix and the high load of computation. In order to solve these problems, we choose the fuzzy membership values of subclusters as the features in the lattice similarity method to construct the similarity matrix. At the same time, the Floyd algorithm developed from the Zadeh’s operations is used to modify the similarity, generate the transitive closure of subclusters' fuzzy relations, which is notably block symmetric. The similarity matrix among subclusters instead of the similarity matrix among all the samples, the novel method reduces the computational cost brought by spectral clustering.

In order to handle the non-spherical datasets classification, in this paper, we design a novel fuzzy clustering method, named multi-center fuzzy C-means algorithm based on transitive closure and spectral clustering (MFCM-TSCS). Large classes or non-spherical shaped datasets are first divided into lots of subclusters by using the FCM algorithm. The fuzzy membership values, which represent the degrees that samples belongs to different subclusters, are used as the features to compute the similarity among subclusters via the lattice similarity method. The Floyd algorithm developed from the Zadeh’s operations is used to modify the similarity, generate the transitive closure of subclusters’ fuzzy relations. The transitive closure is regarded as the similarity matrix to generate the Laplacian matrix used in spectral decomposition. The eigenvectors corresponding to the second minimal eigenvalues and the third minimal eigenvalues are chosen to be the spectral features. At last, the FCM algorithm is applied again to cluster the spectral features, thereby, to merge the subclusters. The main highlights in the novel algorithm: (1) the multi-center initialization method can get rid of the problems that the FCM algorithm is sensitive to the initial prototypes, and it can handle non-traditional curved clusters. (2) The similarity matrix generated by the Floyd algorithm is notably block symmetric, which ensure the effective extraction of spectral features. (3) The problem of clustering samples is transformed to a problem of merging subclusters, the computational load is low, and has strong robustness.

The remainder of this paper is organized as follows. The FCM algorithm and spectral clustering are given in Section 2. Section 3 shows the multi-center FCM algorithm based on transitive closure and spectral clustering. Experimental results show the robustness and the complexity of our method are described in Section 4. Finally, we present our conclusions and discuss future work in Section 5.

2. FCM algorithm and spectral clustering

2.1. Fuzzy C-means (FCM) clustering

Let the sample set be \( X = \{x_1, x_2, \ldots, x_N \} \subset \mathbb{R}^{d \times N} \), where \( N \) is the number of elements in the sample set, and \( d \) is the dimension of the feature space. Divide the sample set \( X \) into \( c \) classes, then the fuzzy partition matrix of \( N \) samples belonging to \( c \) classes is \( U = [u_{ij}]_{c \times N} \), where \( u_{ij} \) (\( 1 \leq i \leq c, 1 \leq j \leq N \)) is the fuzzy membership degree of the \( j \)-th sample \( x_j \) belongs to the \( i \)-th class, and \( u_{ij} \) should satisfy the following two constraints:

\[
\sum_{i=1}^{c} u_{ij} = 1, \quad 1 \leq j \leq N
\]

\[
0 \leq u_{ij} \leq 1, \quad 1 \leq i \leq c, \quad 1 \leq j \leq N
\]

Bezdek [14] defines a general objective of FCM clustering algorithm as:

\[
\min_{M} J_{m}(X, U, P) = \sum_{i=1}^{c} \sum_{j=1}^{N} u_{ij}^{m} d_{ij}^{2}
\]

where \( P = (p_1, p_2, \ldots, p_c) \) in Eq. (3) is a \( q \times c \) matrix; \( p_j = (1, 2, \ldots, c) \in \mathbb{R}^q \) is the center of the \( j \)-th class; \( (d_{ij})^2 = ||x_i - p_j||_2^2 = (x_i - p_j)^T M (x_i - p_j) \) shows the general expression of the distance between \( x_i \) and \( p_j \) and \( M \) is the feature weighting matrix; \( m \) is the fuzzy index which controls the fuzzy degree of the fuzzy partition matrix \( U \).

By using the Lagrange multipliers, we can solve for the fuzzy partition matrix \( U \). The fuzzy membership degree \( u_{ij} \) and the center \( p_i \) can be updated by the expectation–maximization (E–M) algorithm:

E-step:

\[
u_{ij} = \frac{1}{\sum_{k=1}^{c} (d_{ij}^2/d_{kj}^2) (1/(m - 1))} \quad \text{for} \quad j = 1, 2, \ldots, N \quad \text{and} \quad i = 1, 2, \ldots, c
\]

M-step:

\[
p_i = \frac{\sum_{j=1}^{N} u_{ij}^{m} \cdot x_j}{\sum_{j=1}^{N} u_{ij}^{m}} \quad \text{for} \quad i = 1, 2, \ldots, c
\]

The E–M algorithm recursively proceeds until a convergence condition is satisfied. The FCM algorithm is an iterative process to minimize the objective function \( J_{m}(X, U, P) \).

2.2. Spectral clustering

Spectral clustering concerns the optimization of graph cut problem in spectral graph theory. In spectral clustering, a neighborhood graph on the data points is first constructed based on some criteria, such as the fully connected graph or the K-nearest neighbor graph. Then, a weighted affinity matrix \( S \in \mathbb{R}^{N \times N} \) is defined, whose \( (i, j) \) element \( s_{ij} \) reflects the similarity between \( x_i \) and \( x_j \). The measuring of the similarity of samples is an open problem in the spectral clustering. There is not a method of wide application or high validity. Gaussian kernel function is adopted in most references to measure the similarity of samples. Specifically

\[
s_{ij} = \begin{cases} 
\exp(-||x_i - x_j||^2/2\sigma^2) & \text{if } i \neq j \\
0 & \text{if } i = j
\end{cases}
\]

where \( || \cdot || \) denotes the Euclidean norm, \( \sigma \) is the Gaussian kernel parameter. Many examples show that this function is sensitive to \( \sigma \), but the artificial selection of \( \sigma \) leads to some certain limitations. Ref. [13] gets the \( \sigma \) automatically by repeating NJW algorithm. It eliminates the artificial factors, but the computation time, on the other hand, increases. Currently, the generation of the similarity matrix in most references relies on knowledge in different fields. There is not a widely used rule.

Given a neighborhood graph with affinity matrix \( S \), taking the bipartitioning of a graph as an example, \( A \) and \( B \) are two disjoint subsets of graph \( G \), subject to \( A \cap B = \emptyset \) and \( A \cup B = V \), a simple but
efficient clustering criterion is the normalized cut (Ncut) [7] which is defined as
\[
\text{Ncut}(A, B) = \frac{\text{assoc}(A, B)}{\text{assoc}(A, V)} + \frac{\text{assoc}(A, B)}{\text{assoc}(B, V)}
\]
where \(\text{assoc}(A, B) = \sum_{i,j,k} A_{ij}B_{jk}S_{ij} \). In other words, Ncut criterion computes the cut cost as a fraction of the total edge connections to all the nodes and achieves a better balance in the cardinality \(A\) and \(B\). Thus the aim is to minimize Ncut. Following some algebraic formulations, it turns out [8] that minimizing normalized cut can be equivalently recast as
\[
\begin{align*}
\text{min} & \quad \text{Tr}(H^T(D - S)H) \\
\text{s.t.} & \quad H^T D H = I
\end{align*}
\]
where \(A_1, \ldots, A_c\) is a partition of \(X\) \((A_1 \cup \cdots \cup A_c = X, A_i \cap A_j = \emptyset, i \neq j\) and \(A_i \neq \emptyset, i = 1, \ldots, c\), \(c\) is the number of classes, \(D\) is an \(N \times N\) diagonal matrix with \(D_{ii} = \sum_{j} S_{ij}\), \(I\) is the identity matrix, \(H \in \mathbb{R}^{N \times c}\) is a specific discrete matrix, and \(\text{Tr}\) denotes the trace of a matrix.

Unfortunately, solving the above discrete optimization problem is NP-hard. To make it tractable, an efficient relaxation is adopted so as to solve a real-valued problem instead of a discrete-valued one. This is done by computing the first \(k\) generalized eigenvectors \(z_1, \ldots, z_k\), corresponding to the \(k\) smallest eigenvalues, of the generalized eigenvectors problem
\[
(D - S)z = \lambda Dz
\]
where \(\lambda\) is the eigenvalue and \(z\) is the corresponding eigenvector. Furthermore, \(D - S\) is called as the normalized Laplacian matrix which is symmetric positive semi-definite.

Finally, fuzzy C-means method is performed on the row vectors of \(Z = [z_1, \ldots, z_k] \in \mathbb{R}^{N \times k}\) to obtain the clusters.

3. The new multi-center FCM algorithm

Many datasets are non-convex shaped in practical clustering problems. As is shown in Fig. 1(a), red stars and green stars represent samples belonging to different classes. The direct application of the FCM algorithm cannot achieve good results, because the FCM algorithm is only suitable in spherical clusters, it cannot recognize irregular or elongated shaped clusters, and it is sensitive to the initial centers of clusters. For the purpose of solving these problems, we propose the multi-center FCM algorithm based on transitive closure and spectral clustering (MFCM-TCSC). This method divides irregular classes into lots of subclusters (multi-cluster), transforms the classification problem into a problem of merging subclusters. There is no need to consider the initial number of clusters, whether the distribution of datasets is spherical, nor the local optimum problem in clustering.

3.1. The merge criteria of subclusters in the multi-center fuzzy C-means clustering

In order to recognize the irregular classes, one class is represented by multiple subclusters. That is, partitioning the irregular class into some subclusters (multi-clusters) at first, hence the classification problem can be converted into a subcluster merging problem. As is shown in Fig. 1(a), the dataset is made up of two classes. When the multi-center FCM algorithm is adopted, the dataset is clustered into 6 subclusters. The first class is partitioned into 4 subclusters, and the second is partitioned into 2 subclusters. Merge those subclusters and get the results showed in Fig. 1(c).

In the merging the subclusters process, two criteria are acquired from analyzing the results in Fig. 1: (1) the subclusters of the same class should satisfy the similarity and nearness relations (such as the subclusters 1 and 2 in Fig. 1(a)); (2) the similarity can be transitive between the subclusters of the same class (in Fig. 1(a)), the similarity between subcluster 1 and subcluster 4 can be obtained by taking the maximal similarity value through the path: subcluster 1 \(\rightarrow\) subcluster 2 \(\rightarrow\) subcluster 3 \(\rightarrow\) subcluster 4.

In this paper, the number of subclusters is \(\sqrt{N}\), which is the upper limit of the number of cluster based on the Literature [5,14,15]. After the large classes or non-spherical datasets are divided into lots of subclusters by using the FCM algorithm, then merge the subclusters under those two criteria.

3.2. The representation of the subclusters’ features

After the multi-center FCM algorithm clustering, the centers and the fuzzy membership values of subclusters will be derived. The representation of the features of subclusters can be studied in two aspects: the centers of subclusters and the information of fuzzy membership matrix. (1) Replace subclusters with features of centers. That is, the features of the center \(p_i (i = 1, 2, \ldots, c) \in \mathbb{R}^d\) as the features of the ith subcluster. It is the sparsification of samples, which can reduce the computational load of the similarity measure and the spectral clustering. But the position of a center is related closely to the quantity of samples in this cluster which the center represents. If one class is partitioned sparsely and another densely,
the clustering results may be wrong when the features of centers are used as the features of subclusters. (2) Regard the values of the fuzzy membership matrix as the features of subclusters. That is, the values of \( u_{ij} \) (\( i = 1, 2, \ldots, c \)), which is the fuzzy membership values of all samples belongs to the \( i \)th subcluster (the \( i \)th row of the fuzzy membership matrix \( U \)), as the features of the \( i \)th subcluster. For the reason that the fuzzy membership matrix is a fuzzy matrix, the features compose a fuzzy sequence, so that the similarity among subclusters can be measured with the similarity measure of the fuzzy sets.

In this paper, the values of the fuzzy membership matrix \( U = [u_{ij}]_{i \times N} \) are used as the features of subclusters:

\[
U = [u_{ij}]_{i \times N} = \begin{bmatrix}
    u_{11} & u_{12} & \cdots & u_{1N} \\
    u_{21} & u_{22} & \cdots & u_{2N} \\
    \vdots & \vdots & \ddots & \vdots \\
    u_{i1} & u_{i2} & \cdots & u_{iN}
\end{bmatrix} \quad \text{the features of subcluster } i
\]

where \( u_{ij} \) (\( i \leq j \leq c, 1 \leq j \leq N \)) is the fuzzy membership values of the \( j \)th sample \( x_j \) belongs to the \( i \)th subcluster.

### 3.3. The similarity measure among subclusters based on transitive closure

According to the Criterion 1 of the merge subclusters showed in Section 3.1, we want to find a similarity measure which can show the similarity and nearness relations among subclusters. When the values of the fuzzy membership matrix are used as the features of subclusters, plenty of experiments demonstrate that the lattice similarity method will be better while both subcluster \( i \) (the feature is \( u_{ij} \)) and subcluster \( j \) (the feature is \( u_{ij} \)) have the elements belonging to them and unbelonging to them. In this paper, the lattice similarity is used as the similarity measure among subclusters. The formula of lattice similarity is:

\[
r_{ij} = \frac{1}{2} [(u_{ij} \lor u_{ji}) + (1 - u_{ij} \land u_{ji})]
\]

where \( r_{ij} \) is the similarity degree between subclusters \( i \) and \( j \), \( u_{ij} \lor u_{ji} = \bigvee_{k=1}^{N} \{u_{ik} \land u_{jk}\} \) is the inner product of \( u_{ij} \) and \( u_{ji} \), \( u_{ij} \land u_{ji} = \bigwedge_{k=1}^{N} \{u_{ik} \lor u_{jk}\} \) is the exterior product of them. The formula of lattice similarity listed above is given due to its definition. Actually, the lattice similarity does not satisfy the reflexivity. In this paper, we set the elements on its diagonal zero.

Get the similarity of all the subclusters by the similarity measure based on the lattice similarity, build the relationship graph of the similarity which is showed in Fig. 1(b). In this figure, there are six subclusters, and we can know that there are more than one similarity path from the 4th subcluster to the 6th subcluster from the graph.

According to the Criterion 2 of the merge subclusters showed in Section 3.1, we want to construct a similarity measure which can be transitive among the subclusters of the same class. To solve this problem, we need to modify the original lattice similarity. We define that the real similarity of two subclusters is the optimum similarity relation values in the relationship graph. Obviously, to find the real similarity of two subclusters is an optimum path problem, and it embodies the transitivity theory.

Floyd algorithm is a classical method to get the shortest path. The method of Floyd algorithm is introduced as follows. Insert vertex \( v_k \) as an intermediate (all vertices in the path except for the starting vertex and the terminating vertex) between any two vertices and compare the known shortest distance from \( v_j \) to \( v_l \) with the possible distance from \( v_j \) to \( v_l \) which comes out while the existence of the inserted vertex \( v_k \). Displace the original value with the smaller one to get the new weighted adjacency matrix. Repeat it, when all the vertices have been in turn inserted between and as the intermediate, the new weighted adjacency matrix can tell every shortest distance from one vertex to another. It is the distance matrix of graph \( G \).

In this paper, Floyd algorithm is adopted to find the real similarity among subclusters. That is, apply max and min operators of Zadeh’s operations between two subclusters to get the optimum similarity between them. For example, the real similarity between subclusters 1 and 2 in Fig. 1(b) can be formulated as

\[
\begin{align*}
    r_{12} &= \max(r_{12}, \min(r_{13}, r_{32}), \ldots, \min(r_{13}, r_{34}, r_{42}), \ldots) \\
    &= \min(r_{13}, r_{34}, r_{45}, r_{56}, r_{62}, \ldots)
\end{align*}
\]

We can know it from formula (12) that the essence of the computation progress of Floyd algorithm by using max and min operators to get the real similarity is finding the maximum among all the possible paths as the similarity of subclusters and choosing the minimum in a path as the weight of itself.

When we use Floyd algorithm to find the optimum similarity (real similarity), we first consider the condition of inserting only one vertex. The condition of inserting multiple vertices is actually a computational composition of inserting only one vertex. The detailed computational process of finding the real similarity is as follows:

To subclusters \( x \) and \( z \), \( y \) is the set of all the subclusters, \( R_1 \) is the original similarity between \( x \) and \( y \), \( R_2 \) is the original similarity between \( y \) and \( z \). If an optional element of \( y \) is set as the intermediate vertex, the real similarity of \( x \) and \( z \) is \( R_1 \circ R_2 \). The formula of the real similarity is

\[
(R_1 \circ R_2)(x, z) = \bigvee_{y \in Y} (R_1(x, y) \land R_2(y, z))
\]

If the number of subclusters is \( n \) the optimum similarity matrix \( R_{\text{opt}} \) of inserting one intermediate vertex can be derived from original similarity matrix \( R \). Let \( R = (r_{ij})_{n \times n} \), \( R_{\text{opt}} = (b_{ij})_{n \times n} \), then \( R_{\text{opt}} \) is

\[
R_{\text{opt}} = R \circ R = (b_{ij})_{n \times n}
\]

where \( b_{ij} = \bigvee_{k=1}^{n} (r_{ik} \land r_{kj}) \).

The condition of inserting multiple vertices is actually a computational composition of inserting only one vertex, which can be computed with iteration on the fundamental of the former condition. When the similarity matrix converges and does not change any more, it is the real similarity matrix. Let \( R = r_{mn} \), it is the original similarity matrix. Similar to getting the fuzzy equivalent matrix in transitive closure, there is a minimal natural number \( h(h \leq n) \) satisfying \( t(R) = R^h \). To all the natural numbers \( l \) larger than \( h \), there is \( R^l = R^h \), and \( t(R) \) is the real similarity matrix. In this paper, the square method is adopted to get the real similarity matrix \( t(R) \). Square one by one from original similarity matrix \( R \):

\[
R \rightarrow R^2 \rightarrow R^4 \rightarrow \cdots \rightarrow R^{2^l} \rightarrow \cdots
\]

The first time \( R^{2^l} = R^{2^l} \) is the real similarity matrix \( t(R) \).

After \( l \) times squares, we can get the real similarity matrix \( t(R) = R^{2^l} \) of an \( n \times n \) matrix \( R \). It satisfies \( R^{2^l} \leq R^n \), \( 2^l \leq n \). The complexity of this method is \( O(\log_2 n + 1) \).

For example, we can get the real similarity matrix (transitive closure) among subclusters in Fig. 1(b) by our method, which are given in Fig. 2.

Under the guidance of the two criteria of merging subclusters which showed in Section 3.1, we adopt Floyd algorithm to get the transitive closure of subclusters, which is of similarity and nearness relations, also transitivity. In this paper, we define that the real
similarity matrix of the subclusters is the transitive closure of the subclusters.

3.4. The new multi-center FCM algorithm based on transitive closure and spectral clustering

When the transitive closure got from the original similarity of subclusters, the key is to find an algorithm to merge those subclusters by using the transitive closure. Among so many algorithms, spectral clustering algorithm is a method based on the similarity matrix of samples to cluster, so we use spectral clustering to merge those subclusters. In our method, the transitive closure as the similarity matrix is used in spectral clustering to merge those subclusters, which can get the transitive closure to get the selection of parameters in generating the similarity matrix. Furthermore, the transitive closure can reflect the similarity among subclusters more objectively than the matrix got from the Gaussian kernel function. In spectral clustering, Merge subclusters by using FCM algorithm again upon the proper spectral features. In the graph Laplacians theory, the eigenvectors corresponding to the k smallest eigenvalues contains the main classification information. In this paper, we select the eigenvectors corresponding to the second minimal eigenvalues (issues in connection with curve segmentation) and the third minimal eigenvalues (issues in connection with surface segmentation) as the spectral features based on the Literature [7,8].

The novel algorithm contains three phases. First, the data set is partitioned into some subclusters with spherical distribution by using the FCM algorithm with multi-centers, and the fuzzy membership values of subclusters are used as the features to construct the similarity matrix among subclusters via the lattice similarity method. Then, the Floyd algorithm developed from the Zadeh’s operations is used to modify the similarity, generate the transitive closure of subclusters’ fuzzy relations, and the subclusters obtained in the first phase are merged by spectral clustering. Finally, based on these two clustering results, the final results are obtained. The main steps of the proposed method are given as follows:

Algorithm 1 (MFCM-TSCS). Input: Given the number of subclusters $\sqrt{N}$, the number of classes $c$, the distance function $||.||$, the termination criterion $\epsilon$.

Output: The fuzzy membership matrix $U$, the cluster centers $P$ of the classes and the partition of the dataset.

1. Perform the FCM algorithm with multi-centers to get the fuzzy membership matrix and cluster centers of the subclusters. The number of subclusters is $\sqrt{N}$.
2. Regard the values of the fuzzy membership matrix as the features of subclusters. Get the similarity matrices of the subclusters $R$ via the lattice similarity by using formulas (11).
3. Adopt Floyd algorithm in formula (15) to modify the original similarity matrix $R$ into transitive closure $t(R)$.
4. Regard the transitive closure $t(R)$ as the similarity matrix $S$ which is used in spectral decomposition. Build the Laplacian matrix according to formula (9). Decompose the Laplacian matrix to get the spectral features.
5. Select spectral features. Generally, the eigenvectors corresponding to the second minimal eigenvalues and the third minimal eigenvalues are chosen to be the spectral features.
6. Cluster spectral features by using the FCM algorithm to get the results of merging the subclusters to implement the clustering of the dataset.

4. Experiments and analysis

4.1. Experimental setup

In this section, we verify the effectiveness of the MFCM-TCS algorithm. We first apply it to two non-convex clustering problems on artificial datasets. The results are compared with fuzzy C-means
algorithm (FCM) [6], spectral clustering algorithm (SC) [8] and the multi-center FCM algorithm based on the original lattice similarity matrix (MFCM-OLS). In order to show the performance visually, the relationship graphs of the similarity matrix based on the lattice similarity method and the transitive closure of subclusters are shown. In all the algorithms, the desired number of clusters was set in advance.

In the second experiment, we solve seven sub-datasets of the UCI datasets [16] classification problems using our method. The results are compared with those of the fuzzy C-means algorithm (FCM), a modified K-means algorithm using the manifold-distance-based dissimilarity measure (DSKM) [17], the genetic-algorithm-based clustering technique (GAC) proposed by Maulik and Bandyopadhyay [18], DBSCAN algorithm based on density proposed by
Martin Ester [12], and expectation-maximization clustering based on Gaussian mixture models (EM) [19]. Then we make IRIS dataset classification problems by our method as the typical analysis. In all the algorithms, the desired number of clusters was set in advance.

In the third experiment, we solve the segmentation problems of road image by using our method, and compared with the gray-level histogram, FCM, and the EM method.

In the first two experiments, because of the true partitioning is known, we use normalized mutual information (NMI) as the clustering validation measure. NMI is an external clustering validation metric that estimates the quality of the clustering with respect to the given true labels of the datasets: it measures how closely the clustering algorithm could reconstruct the underlying label distribution in the data [20]. If \( C \) is the random variable denoting the cluster assignments of the instances and \( Y \) is the random variable denoting the underlying class labels on the instances, then the NMI measure is defined as:

\[
NMI(C, Y) = \frac{I(C; Y)}{(H(C) + H(Y))/2} \tag{16}
\]

where \( I(C; Y) = H(Y) - H(Y|C) \) is the mutual information between the random variables \( C \) and \( Y \), \( H(C) \) is the Shannon entropy of \( C \), and \( H(Y|C) \) is the conditional entropy of \( Y \) given \( C \). NMI effectively measures the amount of statistical information shared by the random variables representing the cluster assignments and the user-labeled class assignments of the instances. The range of NMI values is 0 to 1. In general, the larger the NMI value is, the better the clustering quality is. NMI is better than other external clustering validation measures such as purity and entropy, since it does not necessarily increase when the number of clusters increases [21].

We implemented the MFCM-TSC method in Matlab 7.0 and conducted our experiments on a PC with PIV 2.33G CPU and 1.95G main memory running under the Microsoft Windows 7 operating system. For each test dataset, we repeated experiments for 20 runs. The parameter settings in the UCI datasets experimental study, for GAC, the maximum iterative number is set to 100, the population size is 20, the crossover probability is 0.8, and the mutation probability is 0.1. For FCM, DSKM and EM, the maximum iterative number is set to 100, and the stop threshold is \( 10^{-4} \). Also, the final result is the average of the results from the 20 runs.

In the last, the noise immunity and computational time of our method is compared with spectral clustering algorithm, so that the superiority of robustness and computational complexity of our method can be distinctly seen.

4.2. Experiments and analysis on artificial datasets, UCI datasets and real Images

4.2.1. Classification on the artificial datasets

4.2.1.1. Classification on the pattern of two moons. The two-moon pattern show in Fig. 3(a) is a typical sample set in studies on non-convex clustering algorithm. Two intertwining moons are in this pattern, and there are 400 samples in each of them. If we use the FCM algorithm to cluster, we will get the clustering result given

![Fig. 4. Classification of 3D ring dataset.](image-url)
in Fig. 3(b). The result is not good. Fig. 3(c) shows the spectral clustering results of two-moon datasets when the Gaussian kernel parameter $\sigma = 0.6$. Note that if $\sigma$ gets a no-proper value, the clustering results can be bad. In this experiment, the spectral clustering results can get the optimum, when $\sigma$ is from 0.45 to 0.7.

When we adopt the FCM algorithm with multi-centers to divide the sample set into 28 subclusters (based on $\sqrt{3450.34} \approx 28$), the result is shown in Fig. 3(d). The adoption of the lattice similarity can get the original similarity relationship graph of the subclusters, as is shown in Fig. 3(e). Then get the relationship graph of the transitive closure among all the subclusters as is shown in Fig. 3(g) by using Floyd algorithm. In Fig. 3(g) we can see that the relationship graph is notable block symmetric so that the clustering information is more outstanding.

According to the graph spectral partitioning theory, the eigenvector corresponding to the second minimal eigenvalue contains the graph partitioning information. In this experiment, the eigenvectors corresponding to the second minimal eigenvalues are chosen to be the spectral features. Fig. 3(f) and (h) corresponds to the spectral features of using original similarity matrix and transitive closure in spectral decomposition. In Fig. 3(f) and (h), the red and black dots represent subclusters’ feature of two different classes; the information on the $x$-coordinate represents the subclusters’ order, the values on the $y$-coordinate are the spectral features of the subclusters.

As is shown in Fig. 3(f), we can know that the result is not good if we use the spectral features got from the original similarity matrix. The result by spectral clustering which based on the transitive closure is showed in Fig. 3(h). Apply the FCM algorithm again, and those subclusters can be merged correctly based on the spectral features, which are derived from the transitive closure. It illustrates that the transitive closure can reflect the similarity and nearness relations among subclusters in a better way.

### 4.2.1.2. Classification of 3D ring shaped dataset

In order to demonstrate the generalization of our algorithm, we now discuss the case of three-dimensional dataset. In Fig. 4(a), there is a dataset of three twining 3D rings, the red, blue, and black dots represent three different classes. In this dataset, the class 1 contains 500 samples, the class 2 contains 900 samples and the class 3 contains 800 samples.

The application of the FCM algorithm divided the dataset into three clusters. The result is poor, as is shown in Fig. 4(b). If the Gaussian kernel parameter $\sigma = 1.8$, the spectral clustering results is shown in Fig. 4(c). The result is very good. In this experiment, the spectral clustering results can get the optimum, when $\sigma$ is from 1.75 to 2.4. Fig. 4(d) shows the results of dividing the sample sets into 47 subclusters (based on $\sqrt{2206 = 47}$) by FCM algorithm with multi-centers. In Fig. 4(d), the class 1 is partitioned into 18 subclusters, the class 2 is partitioned into 16 subclusters and the class 3 is partitioned into 13 subclusters. The relationship graphs of the original lattice similarity and the transitive closure are shown in Fig. 4(e) and (f).

The spectral features by the original lattice similarity matrix and the transitive closure are shown in Table 1. In Table 1, if the original similarity matrix, which is generated by the lattice similarity method, as the Laplacian matrix in spectral decomposition, the spectral features can hardly distinguish the subclusters. Fig. 4(f) shows that the transitive closure modify the original similarity matrix is notably block symmetric. Its corresponding spectral features showed in Table 1 are good. Apply the FCM algorithm again, subclusters can be merged correctly based on those spectral features. In this experiment, the computational time of spectral clustering is 430.34 s, and the computational time of our method is 6.59 s. So our method outperforms spectral clustering obviously in terms of computational time.
Table 2
Comparison our method with FCM, spectral clustering and MFCM-OLS algorithm on artificial datasets.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>FCM</th>
<th>Spectral clustering</th>
<th>MFCM-OLS</th>
<th>MFCM-TCSC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Instance</td>
<td>Class</td>
<td>Number of cluster</td>
<td>NMI</td>
</tr>
<tr>
<td>Two-moon</td>
<td>800</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3D-ring</td>
<td>2200</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 3
Performance comparison of six algorithm in solving the UCI datasets.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Instance</th>
<th>Class</th>
<th>q</th>
<th>NMI (standard deviation)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td></td>
<td></td>
<td></td>
<td>FCM</td>
</tr>
<tr>
<td>Iris</td>
<td>150</td>
<td>3</td>
<td>4</td>
<td>0.7420 (5.34E−04)</td>
</tr>
<tr>
<td>Breast</td>
<td>277</td>
<td>2</td>
<td>9</td>
<td>0.1767 (3.28E−04)</td>
</tr>
<tr>
<td>Zoo</td>
<td>101</td>
<td>7</td>
<td>16</td>
<td>0.6018 (5.27E−04)</td>
</tr>
<tr>
<td>Balance</td>
<td>625</td>
<td>3</td>
<td>4</td>
<td>0.1875 (2.37E−04)</td>
</tr>
<tr>
<td>Pimaindians</td>
<td>768</td>
<td>2</td>
<td>8</td>
<td>0.2128 (8.76E−05)</td>
</tr>
<tr>
<td>Page</td>
<td>5473</td>
<td>5</td>
<td>10</td>
<td>0.6218 (1.13E−04)</td>
</tr>
<tr>
<td>Wine</td>
<td>178</td>
<td>3</td>
<td>13</td>
<td>0.3872 (6.32E−04)</td>
</tr>
</tbody>
</table>
The results are compared with FCM, spectral clustering and MFCM-OLS, and the average results and the standard deviation of the NMI values in 20 runs are shown in Table 2.

4.2.2. Classification of the UCI dataset

In order to validate the performance of the MFCM-TSC algorithm, we select seven lower-dimensional sub-datasets of the UCI datasets, named Iris, breast, zoo, wine, Pimaindians, balance and page, to study a range of different interesting data properties. The results are compared with FCM, DSKM, GAC, DBSCAN and EM. The average results and the standard deviation of the NMI values in 20 runs are shown in Table 3.

From Table 3, we can see clearly that the MFCM-TSC did best on six out of the seven problems, while FCM did best only on the IRIS dataset. DBSCAN, EM and GAC only obtained the desired clustering for the two spheroid datasets, Iris and Page. This is because the structure of the other five datasets does not satisfy convex distribution. On the other hand, the MFCM-TSC can successfully recognize these complex clusters, and DSKM also obtained the true clustering on four problems, iris, zoo, balance and page, which indicates that the manifold distance metrics and the multi-center clustering are very suitable to measure complicated clustering structures. In comparisons between the MFCM-TSC and DSKM, the MFCM-TSC obtained the true clustering on wine and breast. Furthermore, for the Pimaindians problems, the MFCM-TSC did a little better than DSKM in the clustering correct ratio. The main drawback of DSKM is that it has to recalculate the geometrical center of each cluster with the K-means algorithm after cluster assignment, which reduces its ability to reflect global consistency. The MFCM-TSC avoids this drawback by the multi-centers of each class representatives.

From Table 3, we can see clearly that the MFCM-TSC can get rid of the problems that the FCM algorithm is sensitive to the initial prototypes, and it can handle non-traditional curved clusters.

In order to further illustrate our method, we make IRIS dataset classification problems by our method as the typical analysis. IRIS dataset consists of information on 150 iris flowers, 50 each from one of three iris species: Setosa (class 1), Versicolour (class 2), and Virginica (class 3). Each flower is characterized by four numeric attributes, denoted as X1, X2, X3 and X4. By projecting the data onto two dimensional plane, showed in Fig. 5(a) and (b) (the red, blue, and black dots represent three different classes), we can see that there are two large clusters: one consisting of samples of class 1 and the other consisting of samples from class 2 and class 3. The second cluster can further be divided into two subclusters, one composed of samples from class 2 and the other from class 3.

The subclusters constructed by the MFCM-TSC algorithm which has been projected onto X1, X2 and X3, X4 attributes respectively are shown in Fig. 5(a) and (b) (the class 1 is partitioned into subclusters 1st–4th, the class 2 is partitioned into subclusters 5th–8th and the class 3 is partitioned into subclusters 9th–12th; ‘.’, ‘*’, ‘.’, ‘*’ represent the different subcluster). In the MFCM-TSC, the number of subclusters is 12, because of $\sqrt{750} \approx 12$. Then we show the spectral features of all subclusters in Table 4. From the Table 4, we can see these three classes. After merging those 12 subclusters using FCM, we get the final clustering result in Fig. 5(c). From the Table 4 and Fig. 5(c), we can see the resulting clusters are fitting the class label very well, although the samples of subcluster 9 are overlapped samples. There is some overlap between two clusters note that our method might have difficulty in clustering highly overlapped clusters, and the percentage of classification error is because the clustering algorithm is not optimized for classification accuracy but rather for clustering soundness.

4.2.3. Real road segmentation

In this experiment, we discuss the real road segmentation by our method. In Fig. 6(a), there is a picture of a real road, which is made up of street, trees and skies. Grey level values of a pixel and its 8-connected neighborhoods are chosen as the features to partition. Fig. 6(b) shows the gray-level histogram of the real road. Fig. 6(c) shows the segmentation result by the gray-level histogram. The gray-level values used to partition are 70 and 135. Black color represents trees, gray is the street, and white is the sky in Fig. 6(c). We can see it from the result that a large part of the street is partitioned as part of the sky, and some trees are partitioned as the street. Fig. 6(d) shows the clustering result by the FCM algorithm. Green color represents trees, black represents the street, and white is the sky in Fig. 6(d). Though the segmentation result is better than the former one, but some part of the street is partitioned as part of the sky. Fig. 6(e) shows the Gaussian mixture models of the road, when the Gaussian number is three. Fig. 6(f) shows the segmentation result by the EM clustering based on Gaussian mixture models. We can see it from the result that the street and the sky are mixed. In Fig. 6(g), the picture is divided into 7 subclusters given by the FCM algorithm with multi-centers, and the different color represents different subcluster. In this experiment, to illustrate our method briefly, we set the number of subcluster to 7. The transitive closure got from the lattice similarity is regarded as the similarity.
Table 4
The spectral features by the transitive closure.

<table>
<thead>
<tr>
<th>Subclusters</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature 1</td>
<td>0.4183</td>
<td>0.4105</td>
<td>0.3963</td>
<td>0.4183</td>
<td>-0.2023</td>
<td>-0.2056</td>
<td>-0.2023</td>
<td>-0.2056</td>
<td>-0.1913</td>
<td>-0.2044</td>
<td>-0.2044</td>
<td>-0.1947</td>
</tr>
<tr>
<td>Feature 2</td>
<td>0.0004</td>
<td>0.0003</td>
<td>0.0003</td>
<td>0.0004</td>
<td>0.2743</td>
<td>0.3132</td>
<td>0.2743</td>
<td>0.3132</td>
<td>0.1766</td>
<td>-0.4951</td>
<td>-0.4951</td>
<td>-0.3630</td>
</tr>
</tbody>
</table>

Feature 1: the eigenvectors corresponding to the second minimal eigenvalues; feature 2: the eigenvectors corresponding to the third minimal eigenvalues.

Fig. 6. Real road segmentation. (a) The picture of the real road; (b) the gray-level histogram of the road; (c) the segmentation result by gray-level histogram; (d) the segmentation result by the FCM algorithm; (e) the Gaussian mixture models of the road; (f) the segmentation result by the EM clustering based on Gaussian mixture models; (g) the segmentation result of the road into 7 subclusters; (h) the segmentation result by our method.
matrix to generate the Laplacian matrix used in spectral decomposition. The eigenvectors corresponding to the second minimal eigenvalues are chosen to be the spectral features. The eigenvectors corresponding to the second minimal eigenvalues and their corresponding colors in Fig. 6(g) are: $-0.1858$ (green), $-0.3724$ (blue), $0.4682$ (light blue), $0.4682$ (red), $-0.3724$ (pink), $-0.3586$ (black) and $0.3481$ (yellow). Hence, the area in green is a cluster alone, those in blue, pink and black are merged, the left ones in light blue, red, and yellow are merged together. Fig. 6(h) shows the segmentation result given by merging the 7 subclusters into 3 classes, and the green color represents trees, black represents the street, white represents the sky. In spite that a few part of the sky is partitioned as the street, the result is good on the whole.

4.3. Analysis on robustness

In this experiment, noises are added into the two-moon pattern to illustrate the robustness of our method. Fig. 7(a) shows that some bridging noises connecting the two moons. In Fig. 7(a), red dots and black ones represent samples in two classes, blue dots are noise points. To solve this problem, Wang compares results of different algorithms [22]. In order to show the robustness of our method, 10 bridging noises are added into the two-moons pattern (in Fig. 7(a), there are 400 sample points in each class and 10 noise points). Fig. 7(b) shows the classification result by the FCM algorithm, and the clustering result is poor. Fig. 7(c) shows the classification result by spectral clustering. The Gaussian kernel parameter $\sigma$ is 0.6, which is the optimum value when there is no noise. We can see it in Fig. 7(c) that two classes can hardly be divided apart, if the Gaussian kernel parameter is 0.6. Experiments show that if the value of Gaussian kernel parameter changes a little, clustering results will be very bad. In other words, the value should be set again if the datasets change a little. For this reason, spectral clustering algorithm is highly sensitive to noise. Fig. 7(d) shows the result of dividing the datasets into 28 subclusters (based on $\sqrt{800 \approx 28}$) by the FCM algorithm with multi-centers. The spectral

---

**Table 5**

<table>
<thead>
<tr>
<th></th>
<th>Two-moon</th>
<th>Two-moon</th>
<th>Two-moon</th>
<th>Two-moon</th>
<th>3D-ring</th>
<th>3D-ring</th>
<th>3D-ring</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>200</td>
<td>400</td>
<td>600</td>
<td>800</td>
<td>500</td>
<td>900</td>
<td>2200</td>
</tr>
<tr>
<td>Spectral clustering</td>
<td>0.45</td>
<td>2.42</td>
<td>8.61</td>
<td>19.80</td>
<td>4.14</td>
<td>27.90</td>
<td>430.34</td>
</tr>
<tr>
<td>MFCM-TCSC</td>
<td>0.63</td>
<td>0.72</td>
<td>0.80</td>
<td>0.88</td>
<td>0.77</td>
<td>5.09</td>
<td>6.59</td>
</tr>
<tr>
<td>Accelerating rate</td>
<td>0.7</td>
<td>3.4</td>
<td>10.8</td>
<td>22.6</td>
<td>5.38</td>
<td>5.5</td>
<td>65.30</td>
</tr>
</tbody>
</table>
features by the transitive closure modified from the original lattice similarity matrix are shown in Fig. 7(e). The meanings of dots in Fig. 7(e) are the same with those in Fig. 7(h). After merging those 28 subclusters using FCM, we get the final clustering result in Fig. 7(f). The results show that our method can divide two classes apart well, and has strong robustness.

Furthermore, we find that the validity of our method is related to the density of noise points. When the proportion of noise points density to original samples density reaches a certain value, our method will fail. In the two-moon pattern dataset, when the proportion of noise points density to original samples density reaches 50%, our method will fail.

4.4. Analysis on computational complexity

In this experiment, we study the computational complexity of our method. In our method, dataset is divided into multiple subclusters by using the FCM algorithm with multi-centers, and the problem of clustering samples is converted to a problem of merging subclusters, the computational load is low.

The time complexity of the FCM is \(O(N \cdot q \cdot c^2 \cdot i)\), where \(N\) is the number of samples, \(c\) is the number of cluster, \(q\) is the feature dimension, and \(i\) is the number of iteration. The time complexity of transitive closure is \(O(\log_2 N + 1)\). The time complexity of the spectral clustering algorithm based on original dataset is \(O(N^3)\). So our method’s time complexity is \(O(n^3) + O(N \cdot q \cdot n^2 \cdot i) + O(\log_2 n + 1)\), where \(n\) is the number of subclusters. In our method, \(n\) is approximate to \(\sqrt{N}\), the empirical optimum number of cluster. The larger the \(N\), the larger the \(n\). Then the average number of samples in subcluster is more, so that the accelerating effect of our method gets better.

In order to compare the computational times between our method and spectral clustering algorithm, define the proportion of the time of running spectral clustering to the time of running our method as the accelerating rate. This value shows the relative speed of our method to spectral clustering algorithm. Table 5 gives the computational times of spectral clustering algorithm and our method as well as the accelerating rate. The values in Table 5 are the average values got after 50 independent runs on each problem and the experimental parameters are the same as Table 2. The accelerating rate in Table 5 shows that when the quantity of samples is small, the computational times are close. With the increase of samples, the time of running spectral clustering algorithm goes up rapidly, while the time of running our method just rises a little bit. When the number of samples gets 800, the time of running our method is much less than the time of running spectral clustering algorithm, and the accelerating rate is 22.6.

5. Conclusions

In this paper, a new unsupervised fuzzy clustering algorithm is presented. Since multi-centers are used to represent classes, the nonspherical classes can be correctly detected. Moreover, the proposed clustering algorithm can handle non-traditional, curved classes. The experiment shows that our method can get rid of the problems that the FCM algorithm is sensitive to the initial prototypes and the similarity matrix of spectral clustering algorithm is hard to generate. Compared to spectral clustering algorithm, our method greatly improves the noise immunity and computational speed. However, our method might have difficulty in clustering highly overlapped clusters and noisy clusters.

There is still much further work for us to be done. In our experiment, the number of subcluster is \(\sqrt{N}\), lack of theoretical support, how to set the initial number of subclusters is our next major research issue. Moreover, how to generate a widely used, high validated similarity measure method of the subclusters is another important direction of our study. In this paper, regarding the values of the fuzzy membership as the features of subclusters may cause redundancy features. Especially, if samples distribute unbalanced, samples in dense area contribute a lot to the similarity measure among subclusters. So the representation of the features of subclusters is an issue in the further work. There is not a theory which can solve the selection (including how many and what kind of eigenvectors are needed as spectral features) and the usage of the eigenvectors. This problem also is further work for us to be done.

Acknowledgment

This work is supported by National Natural Science Foundation of China under Grants 61303116 and 61072143.

References