Knowledge-Based Systems 74 (2015) 69–88

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Median evidential *c*-means algorithm and its application to community detection

Kuang Zhou [a,b,*], Arnaud Martin [b], Quan Pan [a], Zhun-ga Liu [a]

[a] School of Automation, Northwestern Polytechnical University, Xi'an, Shaanxi 710072, PR China
[b] IRISA, University of Rennes 1, Rue E. Branly, 22300 Lannion, France

ARTICLE INFO

Article history:
Received 3 April 2014
Received in revised form 6 November 2014
Accepted 8 November 2014
Available online 17 November 2014

Keywords:
Credal partition
Belief function theory
Median clustering
Community detection
Imprecise communities

ABSTRACT

Median clustering is of great value for partitioning relational data. In this paper, a new prototype-based clustering method, called Median Evidential *C*-Means (MECM), which is an extension of median *c*-means and median fuzzy *c*-means on the theoretical framework of belief functions is proposed. The median variant relaxes the restriction of a metric space embedding for the objects but constrains the prototypes to be in the original data set. Due to these properties, MECM could be applied to graph clustering problems. A community detection scheme for social networks based on MECM is investigated and the obtained credal partitions of graphs, which are more refined than crisp and fuzzy ones, enable us to have a better understanding of the graph structures. An initial prototype-selection scheme based on evidential semi-centrality is presented to avoid local premature convergence and an evidential modularity function is defined to choose the optimal number of communities. Finally, experiments in synthetic and real data sets illustrate the performance of MECM and show its difference to other methods.

1. Introduction

Cluster analysis or clustering is the task of partitioning a set of n objects $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ into c small groups $\Omega = \{\omega_1, \omega_2, \ldots, \omega_c\}$ in such a way that objects in the same group (called a cluster) are more similar (in some sense or another, like characteristics or behavior) to each other than to those in other groups. The clustering can be used in many fields such as privacy preserving [24], information retrieval [5], text analysis [47], etc. It can also be used as the first step of classification problems to identify the distribution of the training set [44]. Among the existing approaches to clustering, the objective function-driven or prototype-based clustering such as *c*-means and Gaussian mixture modeling is one of the most widely applied paradigms in statistical pattern recognition. These methods are based on a fundamentally very simple, but nevertheless very effective idea, namely to describe the data under consideration by a set of prototypes. They capture the characteristics of the data distribution (like location, size, and shape), and classify the data set based on the similarities (or dissimilarities) of the objects to their prototypes [6].

Generally, a c-partition of n objects in X is a set of $n \times c$ values $\{u_{ij}\}$ arrayed as an $n \times c$ matrix $\boldsymbol{U}$. Each element u_{ij} is the membership of $\boldsymbol{x}_i$ to cluster j. The classical *C*-Means (CM) method aims to partition n observations into c groups in which each observation belongs to the class with the nearest mean, serving as a prototype of the cluster. It results in u_{ij} is either 0 or 1 depending whether object i is grouped into cluster j, and thus each data point is assigned to a single cluster (hard partitions). Fuzzy *C*-Means (FCM), proposed by Dunn [11] and later improved by Bezdek [3], is an extension of *c*-means where each data point can be a member of multiple clusters with membership values (fuzzy partitions) [25].

Belief functions have already been used to express partial information about data both in supervised and unsupervised learning [32,41]. Recently, Masson and Denoeux [32] proposed the application of Evidential *C*-Means (ECM) to get credal partitions [9] for object data. The credal partition is a general extension of the crisp (hard) and fuzzy ones and it allows the object not only to belong to single clusters, but also to belong to any subsets of Ω by allocating a mass of belief for each object in X over the power set 2^{Ω}. The additional flexibility brought by the power set provides more refined partitioning results than those by the other techniques allowing us to gain a deeper insight into the data [32].

All of these aforementioned partition approaches are prototype-based. In CM, FCM and ECM, the prototypes of clusters are the

* Corresponding author at: School of Automation, Northwestern Polytechnical University, Xi'an, Shaanxi 710072, PR China.
E-mail addresses: kzhoumath@163.com (K. Zhou), Arnaud.Martin@univ-rennes1.fr (A. Martin).

http://dx.doi.org/10.1016/j.knosys.2014.11.010
0950-7051/

geometric centers of included data points in the corresponding groups. However, this may be inappropriate as it is the case in community detection problems for social networks, where the prototype (center) of one group is likely to be one of the persons (*i.e.* nodes in the graph) playing the leader role in the community. That is to say, one of the points in the group is better to be selected as a prototype, rather than the center of all the points. Thus we should set some constraints for the prototypes, for example, let them be data objects. Actually this is the basic principle of median clustering methods [16]. These restrictions on prototypes can relax the assumption of a metric space embedding for the objects to be clustered [16,18], and only similarity or dissimilarity between data objects is required. There are some clustering methods for relational data, such as Relational FCM (RFCM) [20] and Relational ECM (RECM) [33], but an underlying metric is assumed for the given dissimilarities between objects. However, in median clustering this restriction is dropped [16]. Cottrell et al. [8] proposed Median *C*-Means clustering method (MCM) which is a variant of the classic *c*-means and proved the convergence of the algorithm. Geweniger et al. [16] combined MCM with the fuzzy *c*-means approach and investigated the behavior of the resulted Median Fuzzy *C*-means (MFCM) algorithm.

Community detection, which can extract specific structures from complex networks, has attracted considerable attention crossing many areas from physics, biology, and economics to sociology. Recently, significant progress has been achieved in this research field and several popular algorithms for community detection have been presented. One of the most popular type of classical methods partitions networks by optimizing some criteria. Newman and Girvan [35] proposed a network modularity measure (usually denoted by *Q*) and several algorithms that try to maximize *Q* have been designed [4,7,10,42]. But recent researches have found that the modularity based algorithms could not detect communities smaller than a certain size. This problem is famously known as the resolution limit [14]. The single optimization criteria *i.e.* modularity may not be adequate to represent the structures in complex networks, thus Amiri et al. [1] suggested a new community detection process as a multi-objective optimization problem. Another family of approaches considers hierarchical clustering techniques. It merges or splits clusters according to a topological measure of similarity between the nodes and tries to build a hierarchical tree of partitions [43,27,37]. Also there are some ways, such as spectral methods [40] and signal process method [23,26], to map topological relationship of nodes on networks into geometrical structures of vectors in *n*-dimensional Euclidian space, where classical clustering methods like CM, FCM and ECM could be evoked. However, there must be some loss of accuracy after the mapping process. As mentioned before, for community detection, the prototypes should be some nodes in the graph. Besides, usually only dissimilarities between nodes are known to us. Due to the application of the relaxation on the data objects and the constraints on the prototypes, the median clustering could be applied to the community detection problem in social networks.

In this paper, we extend the median clustering methods in the framework of belief functions theory and put forward the Median Evidential *C*-Means (MECM) algorithm. Moreover, a community detection scheme based on MECM is also presented. Here, we emphasize two key points different from those earlier studies. Firstly, the proposed approach could provide credal partitions for data set with only known dissimilarities. The dissimilarity measure could be neither symmetric nor fulfilling any metric requirements. It is only required to be of intuitive meaning. Thus it expands application scope of credal partitions. Secondly, some practical issues about how to apply the method into community detection problems such as how to determine the initial prototypes and the optimum community number in the sense of credal partitions are discussed. This makes the approach appropriate for graph partitions and gives us a better understanding of the analyzed networks, especially for the uncertain and imprecise structures.

The rest of this paper is organized as follows: Section 2 recalls the necessary background related to this paper. In Section 3, the median *c*-means algorithm is presented and in Section 4, we show how the proposed method could be applied in the community detection problem. In order to show the effectiveness of our approach, in Section 5 we test our algorithm on artificial and real-world data sets and make comparisons with different methods. The final section makes the conclusions.

2. Background

2.1. Theory of belief functions

Let $\Omega = \{\omega_1, \omega_2, \ldots, \omega_c\}$ be the finite domain of X, called the discernment frame. The mass function is defined on the power set $2^\Omega = \{A : A \subseteq \Omega\}$.

Definition 1. The function $m : 2^\Omega \to [0,1]$ is said to be the Basic Belief Assignment (bba) on 2^Ω, if it satisfies:

$$\sum_{A \subseteq \Omega} m(A) = 1. \tag{1}$$

Every $A \in 2^\Omega$ such that $m(A) > 0$ is called a focal element. The credibility and plausibility functions are defined in Eqs. (2) and (3).

$$Bel(A) = \sum_{B \subseteq A, B \neq \emptyset} m(B), \forall A \subseteq \Omega, \tag{2}$$

$$Pl(A) = \sum_{B \cap A \neq \emptyset} m(B), \forall A \subseteq \Omega. \tag{3}$$

Each quantity $Bel(A)$ measures the total support given to A, while $Pl(A)$ can be interpreted as the degree to which the evidence fails to support the complement of A. The function $pl : \Omega \to [0,1]$ such that $pl(\omega_i) = Pl(\{\omega_i\})$ $(\omega_i \in \Omega)$ is called the contour function associated with m. A belief function on the credal level can be transformed into a probability function by Smets method. In this algorithm, each mass of belief $m(A)$ is equally distributed among the elements of A [39]. This leads to the concept of pignistic probability, *BetP*, defined by

$$BetP(\omega_i) = \sum_{\omega_i \in A \subseteq \Omega} \frac{m(A)}{|A|(1 - m(\emptyset))}, \tag{4}$$

where $|A|$ is the number of elements of Ω in A.

Pignistic probabilities, which play the same role as fuzzy membership, can easily help us make a decision. In fact, belief functions provide us many decision-making techniques not only in the form of probability measures. For instance, a pessimistic decision can be made by maximizing the credibility function, while maximizing the plausibility function could provide an optimistic one [31]. Another criterion [31] considers the plausibility functions and consists in attributing the class A_j for object i if

$$A_j = \arg\max_{X \subseteq \Omega} \{m_i^b(X) Pl_i(X)\}, \tag{5}$$

where

$$m_i^b(X) = K_i^b \lambda_X \left(\frac{1}{|X|^r}\right). \tag{6}$$

In Eq. (5) $m_i^b(X)$ is a weight on $Pl_i(X)$, and r is a parameter in $[0,1]$ allowing a decision from a simple class $(r = 1)$ until the total ignorance Ω $(r = 0)$. The value λ_X allows the integration of the lack of knowledge on one of the focal sets $X \subseteq \Omega$, and it can be set to be 1

simply. Coefficient K_i^b is the normalization factor to constrain the mass to be in the closed world:

$$K_i^b = \frac{1}{1 - m_i(\emptyset)}. \tag{7}$$

2.2. Median c-means and median fuzzy c-means

Median c-means is a variant of the traditional c-means method [8,16]. We assume that n (p-dimensional) data objects $\boldsymbol{x}_i = \{x_{i1}, x_{i2}, \ldots, x_{ip}\}$ $(i = 1, 2, \ldots, n)$ are given. The object set is denoted by $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$. The objective function of MCM is similar to that in CM:

$$J_{\mathrm{MCM}} = \sum_{j=1}^{c}\sum_{i=1}^{n} u_{ij} d_{ij}^2, \tag{8}$$

where c is the number of clusters. As MCM is based on crisp partitions, u_{ij} is either 0 or 1 depending whether $\boldsymbol{x}_i$ is in cluster j. The value d_{ij} is the dissimilarity between $\boldsymbol{x}_i$ and the prototype vector $\boldsymbol{v}_j$ of cluster j $(i = 1, 2, \ldots, n, j = 1, 2, \ldots, c)$, which is not assumed to be fulfilling any metric properties but should reflect the common sense of dissimilarity. Due to these weak assumptions, data object x_i itself may be a general choice and it does not have to live in a metric space [16]. The main difference between MCM and CM is that the prototypes of MCM are restricted to the data objects.

Median fuzzy c-means (MFCM) merges MCM and the standard fuzzy c-means (FCM). As in MCM, it requires the knowledge of the dissimilarity between data objects, and the prototypes are restricted to the objects themselves [16]. MFCM also performs a two-step iteration scheme to minimize the cost function

$$J_{\mathrm{MFCM}} = \sum_{j=1}^{c}\sum_{i=1}^{n} u_{ij}^{\beta} d_{ij}^2, \tag{9}$$

subject to the constrains

$$\sum_{k=1}^{c} u_{ik} = 1, \forall i \in \{1, 2, \ldots, n\}, \tag{10}$$

and

$$\sum_{i=1}^{n} u_{ik} > 0, \forall k \in \{1, 2, \ldots, c\}, \tag{11}$$

where each number $u_{ik} \in [0, 1]$ is interpreted as a degree of membership of object i to cluster k, and $\beta > 1$ is a weighting exponent that controls the fuzziness of the partition. Again, MFCM is preformed by alternating update steps as for MCM:

- Assignment update:

$$u_{ij} = \frac{d_{ij}^{-2/(\beta-1)}}{\sum_{k=1}^{c} d_{ik}^{-2/(\beta-1)}}. \tag{12}$$

- Prototype update: the new prototype of cluster j is set to be $\boldsymbol{v}_j = \boldsymbol{x}_{l^*}$ with

$$\boldsymbol{x}_{l^*} = \arg \min_{\{\boldsymbol{v}_j : \boldsymbol{v}_j = \boldsymbol{x}_l (\in X)\}} \sum_{i=1}^{n} u_{ij}^{\beta} d_{ij}^2. \tag{13}$$

2.3. Evidential c-means

Evidential c-means [32] is a direct generalization of FCM in the framework of belief functions, and it is based on the credal partition first proposed by Denœux and Masson [9]. The credal partition takes advantage of imprecise (meta) classes [30] to express partial knowledge of class memberships. The principle is different from another belief clustering method put forward by Schubert [38], in which conflict between evidence is utilized to cluster the belief functions related to multiple events. In ECM, the evidential membership of an object $\boldsymbol{x}_i = \{\boldsymbol{x}_{i1}, \boldsymbol{x}_{i2}, \ldots, \boldsymbol{x}_{ip}\}$ is represented by a bba $\boldsymbol{m}_i = (m_i(A_j) : A_j \subseteq \Omega)$ over the given frame of discernment $\Omega = \{\omega_1, \omega_2, \ldots, \omega_c\}$. The optimal credal partition is obtained by minimizing the following objective function:

$$J_{\mathrm{ECM}} = \sum_{i=1}^{n} \sum_{A_j \subseteq \Omega, A_j \neq \emptyset} |A_j|^{\alpha} m_i(A_j)^{\beta} d_{ij}^2 + \sum_{i=1}^{n} \delta^2 m_i(\emptyset)^{\beta} \tag{14}$$

constrained on

$$\sum_{A_j \subseteq \Omega, A_j \neq \emptyset} m_i(A_j) + m_i(\emptyset) = 1, \tag{15}$$

where $m_i(A_j) \triangleq m_{ij}$ is the bba of x_i given to the nonempty set A_j, while $m_i(\emptyset) \triangleq m_{i\emptyset}$ is the bba of x_i assigned to the emptyset, and $|\cdot|$ is the cardinality of the set. Parameter α is a tuning parameter allowing to control the degree of penalization for subsets with high cardinality, parameter β is a weighting exponent and δ is an adjustable threshold for detecting the outliers. It is noted that for credal partitions, j is not from 1 to c as before, but ranges in $[0, f]$ with $f = 2^c$. Here d_{ij} denotes the Euclidean distance between x_i and the barycenter (*i.e.* prototype, denoted by $\overline{\boldsymbol{v}}_j$) associated with A_j:

$$d_{ij}^2 = \|x_i - \overline{\boldsymbol{v}}_j\|^2, \tag{16}$$

where $\overline{\boldsymbol{v}}_j$ is defined mathematically by

$$\overline{\boldsymbol{v}}_j = \frac{1}{|A_j|} \sum_{k=1}^{c} s_{kj} \boldsymbol{v}_k, \text{ with } s_{kj} = \begin{cases} 1 & \text{if } \omega_k \in A_k \\ 0 & \text{else} \end{cases}. \tag{17}$$

The notation $\|\cdot\|$ denotes the Euclidean norm of a vector, and $\boldsymbol{v}_k$ is the geometrical center of all the points in cluster k. The update for ECM is given by the following two alternating steps and the update formulas can been obtained by Lagrange multipliers method.

- Assignment update:

$$m_{ij} = \frac{|A_j|^{-\alpha/(\beta-1)} d_{ij}^{-2/(\beta-1)}}{\sum_{A_k \neq \emptyset} |A_k|^{-\alpha/(\beta-1)} d_{ik}^{-2/(\beta-1)} + \delta^{-2/(\beta-1)}}, \quad \forall i = 1, 2 \ldots, n, \forall j / A_j \subseteq \Omega, A_j \neq \emptyset \tag{18}$$

$$m_{i\emptyset} = 1 - \sum_{A_j \neq \emptyset} m_{ij}, \forall i = 1, 2, \ldots, n. \tag{19}$$

- Prototype update: The prototypes (centers) of the classes are given by the rows of the matrix $v_{c \times p}$, which is the solution of the following linear system:

$$\boldsymbol{H}\boldsymbol{V} = \boldsymbol{B}, \tag{20}$$

where $\boldsymbol{H}$ is a matrix of size $(c \times c)$ given by

$$\boldsymbol{H}_{lk} = \sum_{i} \sum_{A_j \supseteq \{\omega_k, \omega_l\}} |A_j|^{\alpha-2} m_{ij}^{\beta}, k, l = 1, 2, \ldots, c, \tag{21}$$

and $\boldsymbol{B}$ is a matrix of size $(c \times p)$ defined by

$$\boldsymbol{B}_{lq} = \sum_{i=1}^{n} x_{iq} \sum_{A_j \ni \omega_l} |A_j|^{\alpha-1} m_{ij}^{\beta}, l = 1, 2, \ldots, c, \ q = 1, 2, \ldots, p. \tag{22}$$

2.4. Some concepts for social networks

In this work we will investigate how the proposed clustering algorithm could be applied to community detection problems in

social networks. In this section some concepts related to social networks will be recalled.

2.4.1. Centrality and dissimilarity

The problem of assigning centrality values to nodes in graphs has been widely investigated as it is important for identifying influential nodes [48]. Gao et al. [15] put forward Evidential Semi-local Centrality (ESC) and pointed out that it is more reasonable than the existing centrality measures such as Degree Centrality (DC), Betweenness Centrality (BC) and Closeness Centrality (CC). In the application of ESC, the degree and strength of each node are first expressed by bbas, and then the fused importance is calculated using the combination rule in belief function theory. The higher the ESC value is, the more important the node is. The detail computation process of ESC can be found in [15].

The similarity or dissimilarity index signifies to what extent the proximity between two vertices of a graph is. The dissimilarity measure considered in this paper is the one put forward by Zhou [49]. This index relates a network to a discrete-time Markov chain and utilizes the mean first-passage time to express the distance between two nodes. One can refer to [49] for more details.

2.4.2. Modularity

Recently, many criteria were proposed for evaluating the partition of a network. A widely used measure called modularity, or Q was presented by Newman and Girvan [35]. Given a hard partition with c groups $(\omega_1, \omega_2, \ldots, \omega_c)$ $\boldsymbol{U} = (u_{ik})_{n\times c}$, where u_{ik} is one if vertex i belongs to the k_{th} community, 0 otherwise, and let the c crisp subsets of vertices be $\{V_1, V_2, \ldots, V_c\}$, then the modularity can be defined as [13]:

$$Q_h = \frac{1}{\|\boldsymbol{W}\|}\sum_{k=1}^{c}\sum_{i,j\in V_k}\left(w_{ij} - \frac{k_i k_j}{\|\boldsymbol{W}\|}\right), \tag{23}$$

where $\|\boldsymbol{W}\| = \sum_{i,j=1}^{n} w_{ij}, k_i = \sum_{j=1}^{n} w_{ij}$. The node subsets $\{V_k, k = 1, 2, \ldots, c\}$ are determined by the hard partition $\boldsymbol{U}_{n\times c}$, but the role of $\boldsymbol{U}$ is somewhat obscured by this form of modularity function. To reveal the role played by the partition $\boldsymbol{U}$ explicitly, Havens et al. [21] rewrote the equations in the form of $\boldsymbol{U}$. Let $\boldsymbol{k} = (k_1, k_2, \ldots, k_n)^T, \boldsymbol{B} = \boldsymbol{W} - \boldsymbol{k}^T\boldsymbol{k}/\|W\|$, then

$$\begin{aligned} Q_h &= \frac{1}{\|\boldsymbol{W}\|}\sum_{k=1}^{c}\sum_{i,j=1}^{n}\left(w_{ij} - \frac{k_i k_j}{\|\boldsymbol{W}\|}\right)u_{ik}u_{jk} \\ &= \frac{1}{\|\boldsymbol{W}\|}\sum_{k=1}^{c}\boldsymbol{u}_k\boldsymbol{B}\boldsymbol{u}_k^{\mathrm{T}} \\ &= \mathrm{trace}\left(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{U}\right)/\|\boldsymbol{W}\|, \end{aligned} \tag{24}$$

where $\boldsymbol{u}_k = (u_{1k}, u_{2k}, \ldots, u_{nk})^{\mathrm{T}}$.

Havens et al. [21] pointed out that an advantage of Eq. (24) is that it is well defined for any partition of the nodes not just crisp ones. The fuzzy modularity of $\boldsymbol{U}$ was derived as

$$Q_f = \mathrm{trace}\left(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{U}\right)/\|\boldsymbol{W}\|, \tag{25}$$

where $\boldsymbol{U}$ is the membership matrix and u_{ik} represents the membership of community k for node i. If u_{ik} is restricted in $[0, 1]$, the fuzzy partition degrades to the hard one, and so Q_f equals to Q_h at this time.

3. Median Evidential C-Means (MECM) approach

We introduce here median evidential c-means in order to take advantages of both median clustering and credal partitions. Like all the prototype-based clustering methods, for MECM, an objective function should first be found to provide an immediate measure of the quality of clustering results. Our goal then can be characterized as the optimization of the objective function to get the best credal partition.

3.1. The objective function of MECM

To group n objects in $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ into c clusters $\omega_1, \omega_2, \ldots, \omega_c$, the credal partition $\boldsymbol{M} = \{\boldsymbol{m}_1, \boldsymbol{m}_2, \ldots, \boldsymbol{m}_n\}$ defined on $\Omega = \{\omega_1, \omega_2, \ldots, \omega_c\}$ is used to represent the class membership of the objects, as in [9,32]. The quantities $m_{ij} = m_i(A_j)$ $(A_j \neq \emptyset, A_j \subseteq \Omega)$ are determined by the dissimilarity between object x_i and focal set A_j which has to be defined first.

Let the prototype set of specific (singleton) cluster be $\boldsymbol{V} = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_c\}$, where $\boldsymbol{v}_i$ is the prototype vector of cluster ω_i $(i = 1, 2, \ldots, c)$ and it must be one of the n objects. If $|A_j| = 1$, *i.e.*, A_j is associated with one of the singleton clusters in Ω (suppose to be ω_j with prototype vector $\boldsymbol{v}_j$), then the dissimilarity between $\boldsymbol{x}_i$ and A_j is defined by

$$\overline{d}_{ij}^2 = d^2(\boldsymbol{x}_i, \boldsymbol{v}_j), \tag{26}$$

where $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$ represents the known dissimilarity between object $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. When $|A_j| > 1$, it represents an imprecise (meta) cluster. If object $\boldsymbol{x}_i$ is to be partitioned into a meta cluster, two conditions should be satisfied. One is the dissimilarity values between $\boldsymbol{x}_i$ and the included singleton classes' prototypes are similar. The other is the object should be close to the prototypes of all these specific clusters. The former measures the degree of uncertainty, while the latter is to avoid the pitfall of partitioning two data objects irrelevant to any included specific clusters into the corresponding imprecise classes. Let the prototype vector of the imprecise cluster associated with A_j be $\overline{\boldsymbol{v}}_j$, then the dissimilarity between $\boldsymbol{x}_i$ and A_j can be defined as:

$$\overline{d}_{ij}^2 = \frac{\gamma\frac{1}{|A_j|}\sum_{\omega_k\in A_j} d^2(\boldsymbol{x}_i, \boldsymbol{v}_k) + \rho_{ij}\min\{d(\boldsymbol{x}_i, \boldsymbol{v}_k) : \omega_k \in A_j\}}{\gamma + 1}, \tag{27}$$

with

$$\rho_{ij} = \frac{\sum_{\omega_x,\omega_y\in A_j}\sqrt{\left(d(\boldsymbol{x}_i, \boldsymbol{v}_x) - d(\boldsymbol{x}_i, \boldsymbol{v}_y)\right)^2}}{\eta\sum_{\omega_x,\omega_y\in A_j} d(\boldsymbol{v}_x, \boldsymbol{v}_y)}. \tag{28}$$

In Eq. (27) γ weights the contribution of the dissimilarity of the objects from the consisted specific clusters and it can be tuned according to the applications. If $\gamma = 0$, the imprecise clusters only consider our uncertainty. Discounting factor ρ_{ij} reflects the degree of uncertainty. If $\rho_{ij} = 0$, it means that all the dissimilarity values between $\boldsymbol{x}_i$ and the included specific classes in A_j are equal, and we are absolutely uncertain about which cluster object $\boldsymbol{x}_i$ is actually in. Parameter η $(\in [0, 1])$ can be tuned to control of the discounting degree. In credal partitions, we can distinguish between "equal evidence" (uncertainty) and "ignorance". The ignorance reflects the indistinguishability among the clusters. In fact, imprecise classes take both uncertainty and ignorance into consideration, and we can balance the two types of imprecise information by adjusting γ. Therefore, the dissimilarity between x_i and $A_j (A_j \neq \emptyset, A_j \subseteq \Omega)$, $\overline{d}_{ij}$, can be calculated by

$$\bar{d}_{ij}^2 = \begin{cases} d^2(\boldsymbol{x}_i, \boldsymbol{v}_j) & |A_j| = 1, \\ \frac{\gamma\frac{1}{|A_j|}\sum_{\omega_k\in A_j} d^2(\boldsymbol{x}_i, \boldsymbol{v}_k) + \rho_{ij}\min\{d(\boldsymbol{x}_i, \boldsymbol{v}_k) : \omega_k \in A_j\}}{\gamma+1} & |A_j| > 1 \end{cases}. \tag{29}$$

Like ECM, we propose to look for the credal partition $\boldsymbol{M} = \{\boldsymbol{m}_1, \boldsymbol{m}_2, \ldots, \boldsymbol{m}_n\} \in \mathcal{R}^{n\times 2^c}$ and the prototype set $\boldsymbol{V} = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_c\}$ of specific (singleton) clusters by minimizing the objective function:

$$J_{\mathrm{MECM}}(\boldsymbol{M},\boldsymbol{V}) = \sum_{i=1}^{n}\sum_{A_j\subseteq\Omega, A_j\neq\emptyset}|A_j|^{\alpha}m_{ij}^{\beta}\overline{d}_{ij}^2 + \sum_{i=1}^{n}\delta^2 m_{i\emptyset}^{\beta}, \tag{30}$$

constrained on

$$\sum_{A_j\subseteq\Omega, A_j\neq\emptyset} m_{ij} + m_{i\emptyset} = 1, \tag{31}$$

where $m_{ij} \triangleq m_i(A_j)$ is the bba of n_i given to the nonempty set A_j, $m_{i\emptyset} \triangleq m_i(\emptyset)$ is the bba of n_i assigned to the empty set, and $\overline{d}_{ij}$ is the dissimilarity between x_i and focal set A_j. Parameters α, β, δ are adjustable with the same meanings as those in ECM. Note that J_{MECM} depends on the credal partition $\boldsymbol{M}$ and the set $\boldsymbol{V}$ of all prototypes.

3.2. The optimization

To minimize J_{MECM}, an optimization scheme via an Expectation–Maximization (EM) algorithm as in MCM [8] and MFCM [16] can be designed, and the alternate update steps are as follows:

Step 1. Credal partition ($\boldsymbol{M}$) update.

- $\forall A_j \subseteq \Omega, A_j \neq \emptyset$,

$$m_{ij} = \frac{|A_j|^{-\alpha/(\beta-1)}\overline{d}_{ij}^{-2/(\beta-1)}}{\sum_{A_k\neq\emptyset}|A_k|^{-\alpha/(\beta-1)}\overline{d}_{ik}^{-2/(\beta-1)} + \delta^{-2/(\beta-1)}} \tag{32}$$

- if $A_j = \emptyset$,

$$m_{i\emptyset} = 1 - \sum_{A_j\neq\emptyset} m_{ij}. \tag{33}$$

Step 2. Prototype ($\boldsymbol{V}$) update.

The prototype v_i of a specific (singleton) cluster ω_i $(i = 1, 2, \ldots, c)$ can be updated first and then the dissimilarity between the object and the prototype of each imprecise (meta) clusters associated with subset $A_j \subseteq \Omega$ can be obtained by Eq. (29). For singleton clusters ω_k $(k = 1, 2, \ldots, c)$, the corresponding new prototypes v_k $(k = 1, 2, \ldots, c)$ are set to be sample x_l orderly, with

$$x_l = \arg\min_{\boldsymbol{v}'_k}\left\{ L(\boldsymbol{v}'_k) \triangleq \sum_{i=1}^{n}\sum_{\omega_k\in A_j}|A_j|^{\alpha}m_{ij}^{\beta}\overline{d}_{ij}^2(\boldsymbol{v}'_k),\ \boldsymbol{v}'_k \in \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}\right\}, \tag{34}$$

The dissimilarity between $\boldsymbol{x}_i$ and A_j, $\overline{d}_{ij}^2$, is a function of $\boldsymbol{v}'_k$, which is the prototype of $\omega_k (\in A_j)$, and it should be one of the n objects in $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$.

The bbas of the objects' class membership are updated identically to ECM [32], but it is worth noting that $\overline{d}_{ij}$ has different meanings and less constraints as explained before. For the prototype updating process the fact that the prototypes are assumed to be one of the data objects is taken into account. Therefore, when the credal partition matrix $\boldsymbol{M}$ is fixed, the new prototypes of the clusters can be obtained in a simpler manner than in the case of ECM application. The MECM algorithm is summarized as Algorithm 1.

Algorithm 1. Median evidential c-means algorithm.

Input	dissimilarity matrix $\boldsymbol{D} \triangleq [d(x_i, x_j)]_{n\times n}$ for the n objects $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$
Parameters	c: number clusters $1 < c < n$ α: weighing exponent for cardinality $\beta > 1$: weighting exponent $\delta > 0$: dissimilarity between any object to the emptyset $\gamma > 0$: weight of dissimilarity between data and prototype vectors $\eta \in [0, 1]$: control of the discounting degree
Initialization	Choose randomly c initial cluster prototypes from the objects
Loop	$t \leftarrow 0$ **Repeat** (1). $t \leftarrow t + 1$ (2). Compute $\boldsymbol{M}_t$ using Eq. (32) and (33) and $\boldsymbol{V}_{t-1}$ (3). Compute the new prototype set $\boldsymbol{V}_t$ using Eq. (34) **Until** the prototypes remain unchanged

The convergence of MECM algorithm can be proved in the following lemma, similar to the proof of median neural gas [8] and MFCM [16].

Lemma 1. *The MECM algorithm (Algorithm 1) converges in a finite number of steps.*

Proof. Suppose $\theta^{(t)} = (\boldsymbol{M}_t, \boldsymbol{V}_t)$ and $\theta^{(t+1)} = (\boldsymbol{M}_{t+1}, \boldsymbol{V}_{t+1})$ are the parameters from two successive iterations of MECM. We will first prove that

$$J_{MECM}(\theta^{(t)}) \geqslant J_{MECM}(\theta^{(t+1)}), \tag{35}$$

which shows MECM always monotonically decreases the objective function. Let

$$\begin{aligned} J_{MECM}^{(t)} &= \sum_{i=1}^{n}\sum_{A_j\subseteq\Omega, A_j\neq\emptyset}|A_j|^{\alpha}(m_{ij}^{(t)})^{\beta}(\overline{d}_{ij}^{(t)})^2 + \sum_{i=1}^{n}\delta^2(m_{i\emptyset}^{(t)})^{\beta} \\ &\triangleq \sum_i\sum_j f_1(\boldsymbol{M}_t)f_2(\boldsymbol{V}_t) + \sum_i f_3(\boldsymbol{M}_t), \end{aligned} \tag{36}$$

where $f_1(\boldsymbol{M}_t) = |A_j|^{\alpha}(m_{ij}^{(t)})^{\beta}, f_2(\boldsymbol{V}_t) = (d_{ij}^{(t)})^2$, and $f_3(\boldsymbol{M}_t) = \delta^2(m_{i\emptyset}^{(t)})^{\beta}$. $\boldsymbol{M}_{t+1}$ is then obtained by maximizing the right hand side of the equation above. Thus,

$$J_{MECM}^{(t)} \geqslant \sum_i\sum_j f_1(\boldsymbol{M}_{t+1})f_2(\boldsymbol{V}_t) + \sum_i f_3(\boldsymbol{M}_{t+1}) \tag{37}$$

$$\geqslant \sum_i\sum_j f_1(\boldsymbol{M}_{t+1})f_2(\boldsymbol{V}_{t+1}) + \sum_i f_3(\boldsymbol{M}_{t+1}) \tag{38}$$

$$= J_{MECM}^{(t+1)}. \tag{39}$$

This inequality (37) comes from the fact $\boldsymbol{M}_{t+1}$ is determined by differentiating of the respective Lagrangian of the cost function with respect to $\boldsymbol{M}_t$. To get Eq. (38), we could use the fact that every prototype $\boldsymbol{v}_k$ $(k = 1, 2, \ldots, c)$ in $\boldsymbol{V}_{t+1}$ is orderly chosen explicitly to be

$$\arg\min_{\boldsymbol{v}'_k}\left\{ L(v'_k) \triangleq \sum_{i=1}^{n}\sum_{\omega_k\in A_j}|A_j|^{\alpha}m_{ij}^{\beta}\overline{d}_{ij}^2(\boldsymbol{v}'_k),\ \boldsymbol{v}'_k \in \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}\right\},$$

and thus this formula evaluated at $\boldsymbol{V}_{t+1}$ must be equal to or less than the same formula evaluated at V_t.

Hence MECM causes the objective function to converge monotonically. Moreover, the bba $\boldsymbol{M}$ is a function of the prototypes $\boldsymbol{V}$ and for given $\boldsymbol{V}$ the assignment $\boldsymbol{M}$ is unique. Because MECM assumes that the prototypes are original object data in $\boldsymbol{X}$, so there is a finite number of different prototype vectors $\boldsymbol{V}$ and so is the number of corresponding credal partitions $\boldsymbol{M}$. Consequently we can get the conclusion that the MECM algorithm converges in a finite number of steps. □

Remark 1. Although the objective function of MECM takes the same form as that in ECM [32], we should note that in MECM, it is no longer assumed that there is an underlying Euclidean distance. Thus the dissimilarity measure d_{ij} has few restrictions such as the triangle inequality or the symmetry. This freedom distinguishes the MECM from ECM and RECM, and it leads to the constraint for the prototypes to be data objects themselves. The distinct difference in the process of minimization between MECM and ECM lies in the prototype-update step. The purpose of updating the prototypes is to make sure that the cost function would decrease. In ECM the Lagrange multiplier optimization is evoked directly while in MECM a search method is applied. As a result, the objective function may decline more quickly in ECM as the optimization process has few constraints. However, when the centers of clusters in the data set are more likely to be the data object, MECM may converge with few steps.

Remark 2. Although both MECM and MFCM can be applied to the same type of data set, they are very different. This is due to the fact that they are founded on different models of partitioning. MFCM provides fuzzy partition. In contrast, MECM gives credal partitions. We emphasize that MECM is in line with MCM and MFCM: each class is represented by a prototype which is restricted to the data objects and the dissimilarities are not assumed to be fulfilling any metric properties. MECM is an extension of MCM and MFCM in the framework of belief functions.

3.3. The parameters of the algorithm

As in ECM, before running MECM, the values of the parameters have to be set. Parameters α, β and δ have the same meanings as those in ECM, and γ weighs the contribution of uncertainty to the dissimilarity between nodes and imprecise clusters. The value β can be set to be $\beta = 2$ in all experiments for which it is a usual choice. The parameter α aims to penalize the subsets with high cardinality and control the amount of points assigned to imprecise nodes and meta classes are different, thus different values of α should be taken even for the same data set. But both in ECM and MECM, the higher α is, the less mass belief is assigned to the meta clusters and the less imprecise will be the resulting partition. However, the decrease of imprecision may result in high risk of errors. For instance, in the case of hard partitions, the clustering results are completely precise but there is much more intendancy to partition an object to an unrelated group. As suggested in [32], a value can be used as a starting default one but it can be modified according to what is expected from the user. The choice δ is more difficult and is strongly data dependent [32].

For determining the number of clusters, the validity index of a credal partition defined by Masson and Denoeux [32] could be utilized:

$$N^*(c) \triangleq \frac{1}{n\log_2(c)} \times \sum_{i=1}^{n}\left[\sum_{A\in 2^{\Omega}\setminus\emptyset} m_i(A)\log_2|A| + m_i(\emptyset)\log_2(c)\right], \tag{40}$$

where $0 \leqslant N^*(c) \leqslant 1$. This index has to be minimized to get the optimal number of clusters. When MECM is applied to community detection, a different index is defined to determine the number of communities. We will describe it in the next section.

4. Application and evaluation issues

In this section, we will discuss how to apply MECM to community detection problems in social networks and how to evaluate credal partitions.

4.1. Evidential modular function

Assume the obtained credal partition of the graph is

$$\boldsymbol{M} = [\boldsymbol{m}_1, \boldsymbol{m}_2, \ldots, \boldsymbol{m}_n]^{\mathrm{T}},$$

where $\boldsymbol{m}_i = (m_{i1}, m_{i2}, \ldots, m_{ic})^{\mathrm{T}}$. Similarly to the fuzzy modularity by Havens et al. [21], here we introduce an evidential modularity [50]:

$$Q_e = \frac{1}{\|\boldsymbol{W}\|}\sum_{k=1}^{c}\sum_{i,j=1}^{n}\left(w_{ij} - \frac{k_i k_j}{\|\boldsymbol{W}\|}\right) pl_{ik} pl_{jk}, \tag{41}$$

where $\boldsymbol{pl}_i = (pl_{i1}, pl_{i2}, \ldots, pl_{ic})^{\mathrm{T}}$ is the contour function associated with m_i, which describes the upper value of our belief to the proposition that the i_{th} node belongs to the k_{th} community.

Let $\boldsymbol{PL} = (pl_{ik})_{n\times c}$, then Eq. (41) can be rewritten as:

$$Q_e = \frac{\mathrm{trace}(\boldsymbol{PL}^{\mathrm{T}}\boldsymbol{BPL})}{\|\boldsymbol{W}\|}. \tag{42}$$

Q_e is a directly extension of the crisp and fuzzy modularity functions in Eq. (24). When the credal partition degrades into the hard and fuzzy ones, Q_e is equal to Q_h and Q_f respectively.

4.2. The initial prototypes for communities

Generally speaking, the person who is the center in the community in a social network has the following characteristics: he has relation with all the members of the group and their relationship is stronger than usual; he may directly contact with other persons who also play an important role in their own community. For instance, in Twitter network, all the members in the community of the fans of Real Madrid football Club (RMC) are following the official account of the team, and RMC must be the center of this community. RMC follows the famous football player in the club, who is sure to be the center of the community of his fans. In fact, RMC has 10,382,777 followers and 30 followings (the data on March 16, 2014). Most of the followings have more than 500,000 followers. Therefore, the centers of the community can be set to the ones not only with high degree and strength, but also with neighbors who also have high degree and strength. Thanks to the theory of belief functions, the evidential semi-local centrality ranks the nodes considering all these measures. Therefore the initial c prototypes of each community can be set to the nodes with largest ESC values.

Note that there is usually more than one center in one community. Take Twitter network for example again, the fans of RMC who follow the club official account may also pay attention to Cristiano Ronaldo, the most popular player in the team, who could be another center of the community of RMC's fans to a great extent. These two centers (the accounts of the club and Ronaldo) both have large ESC values but they are near to each other. This situation violates the rule which requires the chosen seeds as far away from each other as possible [2,26].

The dissimilarity between the nodes could be utilized to solve this problem. Suppose the ranking order of the nodes with respect to their ESCs is $n_1 \geqslant n_2 \geqslant \cdots \geqslant n_n$. In the beginning n_1 is set to be the first prototype as it has the largest ESC, and then node n_2 is considered. If $d(n_1, n_2)$ (the dissimilarity between node 1 and 2) is larger than a threshold μ, it is chosen to be the second prototype. Otherwise, we abandon n_2 and turn to check n_3. The process continues until all the c prototypes are found. If there are not enough prototypes after checking all the nodes, we should decrease μ moderately and restart the search from n_1. In this paper we test the approach with the dissimilarity measure proposed in [49]. Based on our experiments, $[0.7, 1]$ is a better experiential range of the threshold μ. This seed choosing strategy is similar to that in [26].

4.3. The community detection algorithm based on MECM

The whole community detection algorithm in social networks based on MECM is summarized in Algorithm 2.

Algorithm 2. Community detection algorithm based on MECM.

Input: $\boldsymbol{A}$, the adjacency matrix; $\boldsymbol{W}$, the weight matrix (if any); μ, the threshold controlling the dissimilarity between the prototypes; $c_{\min}$, the minimal number of communities; $c_{\max}$, the maximal of communities; the required parameters in original MECM algorithm
Initialization: Calculate the dissimilarity matrix of the nodes in the graph.
repeat
(1). Set the cluster number c in MECM be $c = c_{\min}$.
(2). Choose the initial c prototypes using the strategy proposed in Section 4.2.
(3). Run MECM with the corresponding parameters and the initial prototypes got in (2).
(4). Calculate the evidential modularity using Eq. (42).
(5). Let $c = c + 1$.
until c reaches at $c_{\max}$.
Output: Choose the number of communities at around which the modular function peaks, and output the corresponding credal partition of the graph.

In the algorithm, $c_{\min}$ and $c_{\max}$ can be determined based on the original graph. Note that $c_{\min} \geqslant 2$. It is an empirical range of the community number of the network. If c is given, we can get a credal partition based on MECM and then the evidential modularity can be derived. As we can see, the modularity is a function of c and it should peak at around the optimal value of c for the given network.

4.4. Performance evaluation

The objective of the clustering problem is to partition a similar data pair to the same group. There are two types of correct decisions by the clustering result: a true positive (TP) decision assigns two similar objects to the same cluster, while a true negative (TN) decision assigns two dissimilar objects to different clusters. Correspondingly, there are two types of errors we can commit: a false positive (FP) decision assigns two dissimilar objects to the same cluster, while a false negative (FN) decision assigns two similar objects to different clusters. Let a (respectively, b) be the number of pairs of objects simultaneously assigned to identical classes (respectively, different classes) by the stand reference partition and the obtained one. Actually a (respectively, b) is the number of TP (respectively, TN) decisions. Similarly, let c and d be the numbers of FP and FN decisions respectively. Two popular measures that are typically used to evaluate the performance of hard clusterings are precision and recall. Precision (P) is the fraction of relevant instances (pairs in identical groups in the clustering benchmark) out of those retrieved instances (pairs in identical groups of the discovered clusters), while recall (R) is the fraction of relevant instances that are retrieved. Then precision and recall can be calculated by

$$\mathrm{P} = \frac{a}{a+c} \quad \text{and} \quad \mathrm{R} = \frac{a}{a+d}, \tag{43}$$

respectively. The Rand index (RI) measures the percentage of correct decisions and it can be defined as

$$\mathrm{RI} = \frac{2(a+b)}{n(n-1)}, \tag{44}$$

where n is the number of data objects. In fact, precision measures the rate of the first type of errors (FP), recall (R) measures another type (FN), while RI measures both.

For fuzzy and evidential clusterings, objects may be partitioned into multiple clusters with different degrees. In such cases precision would be consequently low [34]. Usually the fuzzy and evidential clusters are made crisp before calculating the measures, using for instance the maximum membership criterion [34] and pignistic probabilities [32]. Thus in the work presented in this paper, we have hardened the fuzzy and credal clusters by maximizing the corresponding membership and pignistic probabilities and calculate precision, recall and RI for each case.

The introduced imprecise clusters can avoid the risk to group a data into a specific class without strong belief. In other words, a data pair can be clustered into the same specific group only when we are quite confident and thus the misclassification rate will be reduced. However, partitioning too many data into imprecise clusters may cause that many objects are not identified for their precise groups. In order to show the effectiveness of the proposed method in these aspects, we use the evidential precision (EP) and evidential recall (ER):

$$\mathrm{EP} = \frac{n_{er}}{N_e}, \quad \mathrm{ER} = \frac{n_{er}}{N_r}. \tag{45}$$

In Eq. (45), the notation N_e denotes the number of pairs partitioned into the same specific group by evidential clusterings, and n_{er} is the number of relevant instance pairs out of these specifically clustered pairs. The value N_r denotes the number of pairs in the same group of the clustering benchmark, and ER is the fraction of specifically retrieved instances (grouped into an identical specific cluster) out of these relevant pairs. When the partition degrades to a crisp one, EP and ER equal to the classical precision and recall measures respectively. EP and ER reflect the accuracy of the credal partition from different points of view, but we could not evaluate the clusterings from one single term. For example, if all the objects are partitioned into imprecise clusters except two relevant data object grouped into a specific class, $\mathrm{EP} = 1$ in this case. But we could not say this is a good partition since it does not provide us with any information of great value. In this case $\mathrm{ER} \approx 0$. Thus ER could be used to express the efficiency of the method for providing valuable partitions. Certainly we can combine EP and ER like RI to get the evidential rank index (ERI) describing the accuracy:

$$\mathrm{ERI} = \frac{2(a^* + b^*)}{n(n-1)}, \tag{46}$$

where a^* (respectively, b^*) is the number of pairs of objects simultaneously clustered to the same specific class (*i.e.*, singleton class, respectively, different classes) by the stand reference partition and the obtained credal one. Note that for evidential clusterings, precision, recall and RI measures are calculated after the corresponding hard partitions are got, while EP, ER and ERI are based on hard credal partitions [32].

Example 1. In order to show the significance of the above performance measures, an example containing only ten objects from two groups is presented here. The three partitions are given in Fig. 1b–d. The values of the six evidential indices (P,R,RI,EP,ER,-ERI) are listed in Table 1. We can see that if we simply partition the nodes in the overlapped area, the risk of misclassification is high in terms of precision. The introduced imprecise cluster $\omega_{12} \triangleq \{\omega_1, \omega_2\}$ could enable us to make soft decisions, as a result the accuracy of the specific partitions is high. However, if too many objects are clustered into imprecise classes, which is the case of partition 3, it is pointless although EP is high. Generally, EP denotes the accuracy of the specific decisions, while ER represents the

efficiency of the approach. We remark that the evidential indices degrade to the corresponding classical indices (*e.g.*, evidential precision degrades to precision) when the partition is crisp.

5. Experiments

In this section a number of experiments are performed on classical data sets in the distance space and on graph data for which only the dissimilarities between nodes are known. The obtained credal partitions are compared with hard and fuzzy ones using the evaluation indices proposed in Section 4.4 to show the merits of MECM.

5.1. Overlapped data set

Clustering approaches to detect overlap objects which leads to recent attentions are still inefficiently processed. Due to the introduction of imprecise classes, MECM has the advantage to detect overlapped clusters. In the first example, we will use overlapped data sets to illustrate the behavior of the proposed algorithm.

We start by generating 2×100 points uniformly distributed in two overlapped circles with a same radius $R = 30$ but with different centers. The coordinates of the first circle's center are $(0, 0)$ while the coordinates of the other circle's center are $(30, 30)$. The data set is displayed in Fig. 2a.

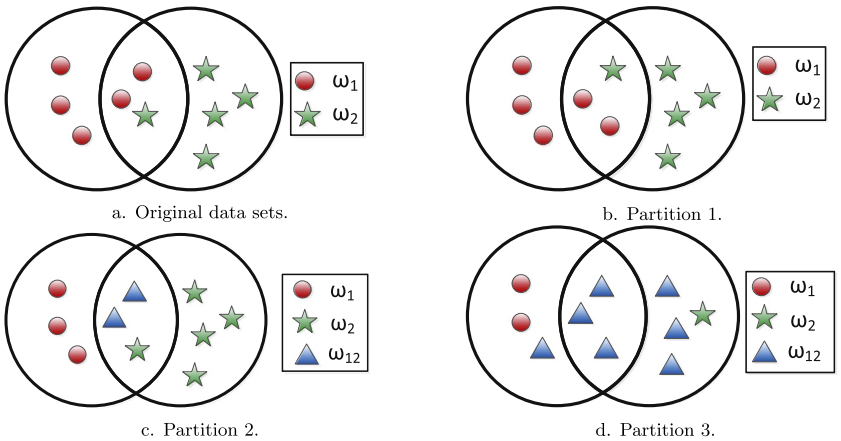
In order to show the influence of parameters in MECM and ECM, different values of γ, α, η and δ have been tested for this data set. The figure Fig. 3a displays the three evidential indices varying with γ (α is fixed to be 2) by MECM, while Fig. 3b depicts the results of MECM with different α but a fixed $\gamma = 0.4$ (η and δ are set 0.7 and 50, respectively, in the tests). For fixed α and γ, the results with different η and δ are shown in Fig. 3c. The effect of α and δ on the clusterings of ECM is illustrated in Fig. 3e. As we can see, for both MECM and ECM, if we want to make more imprecise decisions to improve ER, parameter α can be decreased. In MECM, we can also reduce the value of parameter γ to accomplish the same purpose. Although both α and γ have effect on imprecise clusters in MECM, the mechanisms they work are different. Parameter α tries to adjust the penalty degree to control the imprecise rates of the results. However, for γ, the same aim could be got by regulating the uncertainty degree of imprecise classes. It can be seen from the figures, the effect of γ is more conspicuous than α. Moreover, although α may be set too high to obtain good clusterings, "good" partitions can also be got by adjusting γ in this case. For both MECM and ECM, the stable limiting values of evidential measures are around 0.7 and 0.8. Such values suggest the equivalence of the two methods to a certain extent. Parameter η is used for discounting the distance between uncertain objects and specific clusters. As pointed out in Fig. 3c, if γ and α are well set, it has little effect on the final clusterings. The same is true in the case of δ which is applied to detect outliers. The effect of the different values of parameter β is illustrated in Fig. 3d. We can see that it has little influence on the final results as long as it is larger than 1. As in FCM and ECM, for which it is a usual choice, we use $\beta = 2$ in all the following experiments.

Table 1
Evaluation indices of the partitions.

	EP	ER	ERI	P	R	RI
Partition 1	0.6190	0.6190	0.6444	0.6190	0.6190	0.6444
Partition 2	1.0000	0.6190	0.8222	–	–	–
Partition 3	1.0000	0.0476	0.5556	–	–	–

The improvement of precision will bring about the decline of recall, as more data could not be clustered into specific classes. What we should do is to set parameters based on our own requirement to make a tradeoff between precision and recall. For instance, if we want to make a cautious decision in which EP is relatively high, we can reduce γ and α. Values of these parameters can be also learned from historical data if such data are available.

For the objects in the overlapped area, it is difficult to make a hard decision *i.e.* to decide about their specific groups. Thanks to the imprecise clusters, we can make a soft decision. As analyzed before, the soft decision will improve the precision of total results and reduce the risk of misclassifications caused by simply partitioning the overlapped objects into specific class. However, too many imprecise decisions will decrease the recall value. Therefore, the ideal partition should make a compromise between the two measures. Set $\alpha = 1.8, \gamma = 0.2, \eta = 0.7$ and $\delta = 50$, the "best" (with relatively high values on both precision and recall) clustering result by MECM is shown in Fig. 2b. As we can see, most of the data in the overlapped area are partitioned into imprecise cluster $\omega_{12} \triangleq \{\omega_1, \omega_2\}$ by the application of MECM. We adjust the coordinates of the center of the second circle to get overlapped data with different proportions (overlap rates), and the validity indices of the clustering results by different methods are illustrated in Fig. 4. For the application of MECM, MCM and MFCM, each algorithm is evoked 20 times with randomly selected initial prototypes for the same data set and the mean values of the evaluating indices are reported. The figure Fig. 4d shows the average values of the indices by MECM (plus and minus one standard deviation) for 20 repeated experiments as a function of the overlap rates. As we can see the initial prototypes indeed have effects on the final results, especially when the overlap rates are high. Certainly, we



a. Original data sets. b. Partition 1. c. Partition 2. d. Partition 3.

Fig. 1. An small data set with imprecise classes.

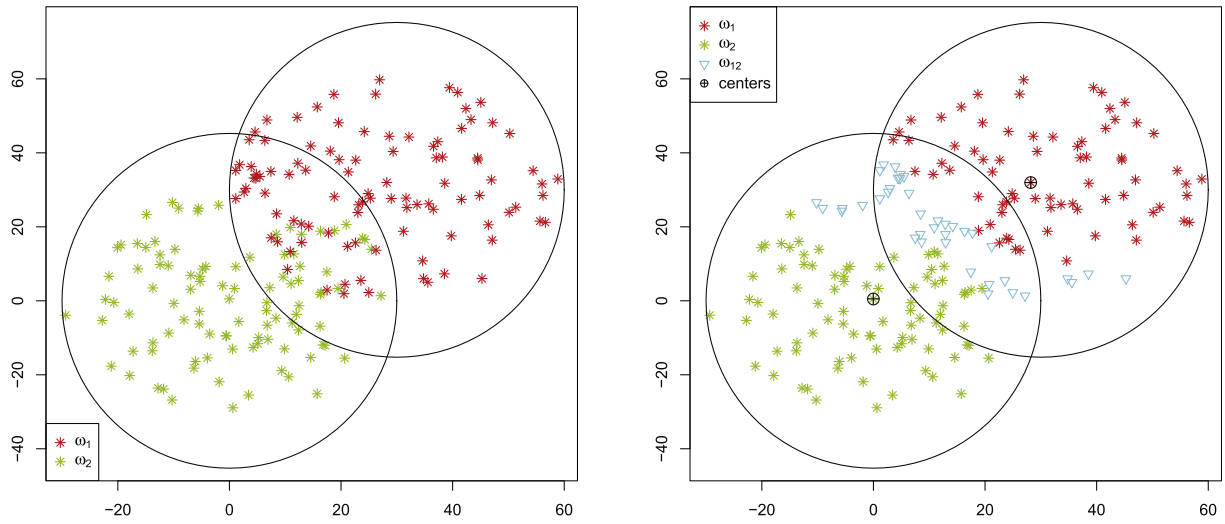


Fig. 2. Clustering of overlapped data set.

can avoid the influence by repeating the algorithm many times. But this is too expensive for MECM. Therefore, we suggest to use the prototypes obtained in MFCM or MCM as the initial. In the following experiments, we will set the initial prototypes to be the ones got by MFCM.

As it can be seen, for different overlap rates, the classical measures such as precision, recall, and RI are almost the same for all the methods. This reflects that pignistic probabilities play a similar role as fuzzy membership. But we can see that for MECM, EP is significantly high, and the increasing of overlap rates has least effects on it compared with the other methods. Such effect can be attributed to the introduced imprecise clusters which enable us to make a compromise decision between hard ones. But as many points are clustered into imprecise classes, the evidential recall value is low.

Overall, this example reflects one of the superiority of MECM that it can detect overlapped clusters. The objects in the overlapped area could be clustered into imprecise classes by this approach. Other possible available information or special techniques could be utilized for these imprecise data when we have to make hard decisions. Moreover, partitions with different degree of imprecision can be got by adjusting the parameters of the algorithm based on our own requirement.

5.2. Classical data sets from Gaussian mixture model

In the second experiment, we test on a data set consisting of $3\times 50+2\times 5$ points generated from different Gaussian distributions. The first 3×50 points are from Gaussian distributions $G(\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)(k=1,2,3)$ with

$$\boldsymbol{\mu}_1=(0,0)^{\mathrm{T}},\boldsymbol{\mu}_2=(40,40)^{\mathrm{T}},\boldsymbol{\mu}_3=(80,80)^{\mathrm{T}}$$
$$\boldsymbol{\Sigma}_1=\boldsymbol{\Sigma}_2=\boldsymbol{\Sigma}_3=\begin{pmatrix}120 & 0\\ 0 & 120\end{pmatrix}, \tag{47}$$

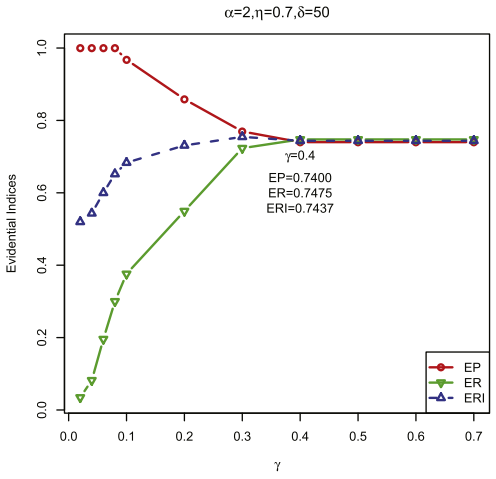
and the last 2×10 data are noisy points follow $G(\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)(k=4,5)$ with

$$\boldsymbol{\mu}_4=(-50,90)^{\mathrm{T}},\boldsymbol{\mu}_5=(-10,130)^{\mathrm{T}}$$
$$\boldsymbol{\Sigma}_4=\boldsymbol{\Sigma}_5=\begin{pmatrix}80 & 0\\ 0 & 80\end{pmatrix}. \tag{48}$$

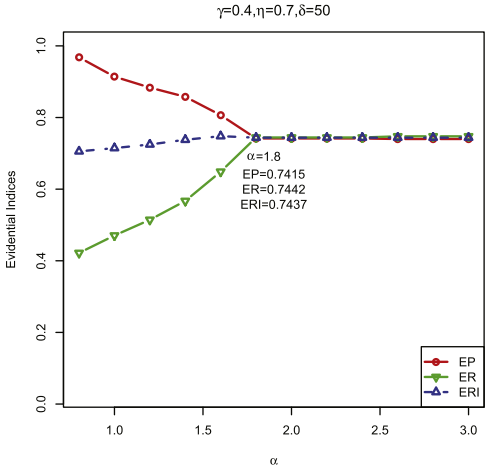
MECM is applied with the following settings: $\alpha=1,\delta=100$, $\eta=0.7$, while ECM has been tested using $\alpha=1.7,\delta=100$ (The appropriate parameters can be determined similarly as in the first example). One can see from Fig. 5b and c, MCM and MFCM can partition most of the regular data in ω_1,ω_2 and ω_3 into their correct clusters, but they could not detect the noisy points correctly. These noisy data are simply grouped into a specific cluster by both approaches. As can be seen from Fig. 5d, for the points located in the middle part of ω_2, ECM could not find their exact group and misclassify them into imprecise cluster ω_{13}. In the figures $\omega_{ij}\triangleq\{\omega_i,\omega_j\}$ denotes imprecise clusters.

As mentioned before, imprecise classes in MECM can measure ignorance and uncertainty at the same time, and the degree of ignorance in meta clusters can be adjusted by γ. We can see that MECM does not detect many points in the overlapped area between two groups if γ is set to 0.6. In such a case the test objects are partitioned into imprecise clusters mainly because of our ignorance about their specific classes. These objects attributed to meta classes mainly belong to noisy data in ω_4 and ω_5. The distance of these points to the prototypes of specific clusters is large (but not too large or they could be regarded to be in the emptyset, see Fig. 5e). Thus the distance between the prototype vectors is relatively small so that these specific clusters are indistinguishable. Decreasing γ to be 0.2 would make imprecise class denoting more uncertainty, as it can be seen from Fig. 5f, where many points located in the margin of each group are clustered into imprecise classes. In such a case, meta classes rather reflect our uncertainty on the data objects' specific cluster.

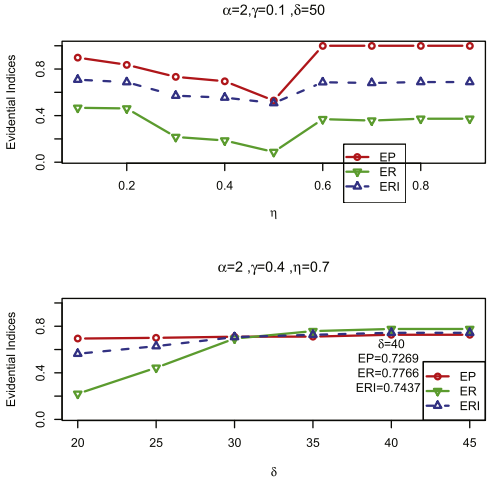
The table Table 2 lists the indices for evaluating the different methods. Bold entries in each column of this table (and also other tables in the following) indicate that the results are significant as the top performing algorithm(s) in terms of the corresponding evaluation index. We can see that the precision, recall and RI values for all approaches are similar except from those obtained for ECM which are significantly lower. As these classical measures are based on the associated pignistic probabilities for evidential clusterings, it seems that credal partitions can provide the same information as crisp and fuzzy ones. But from the same table, we can also see that the evidential measures EP and ERI obtained for MECM are higher (for hard partitions, the values of evidential measures equal to the corresponding classical ones) than the ones obtained for other methods. This fact confirms the accuracy of the specific decisions *i.e.* decisions clustering the objects into specific classes. The advantage can be attributed to the introduction of



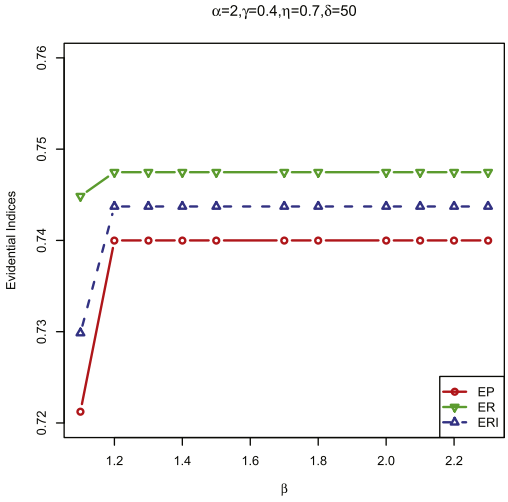
a. MECM (with respect to γ)



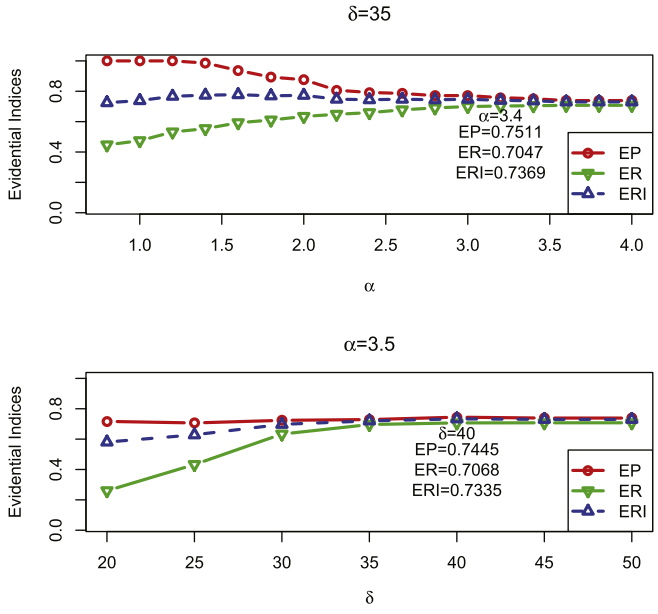
b. MECM (with respect to α)



c. MECM (with respect to η or δ)



d. MECM (with respect to β)



e. ECM (with respect to α or δ)

Fig. 3. Clustering of overlapped data set with different parameters.

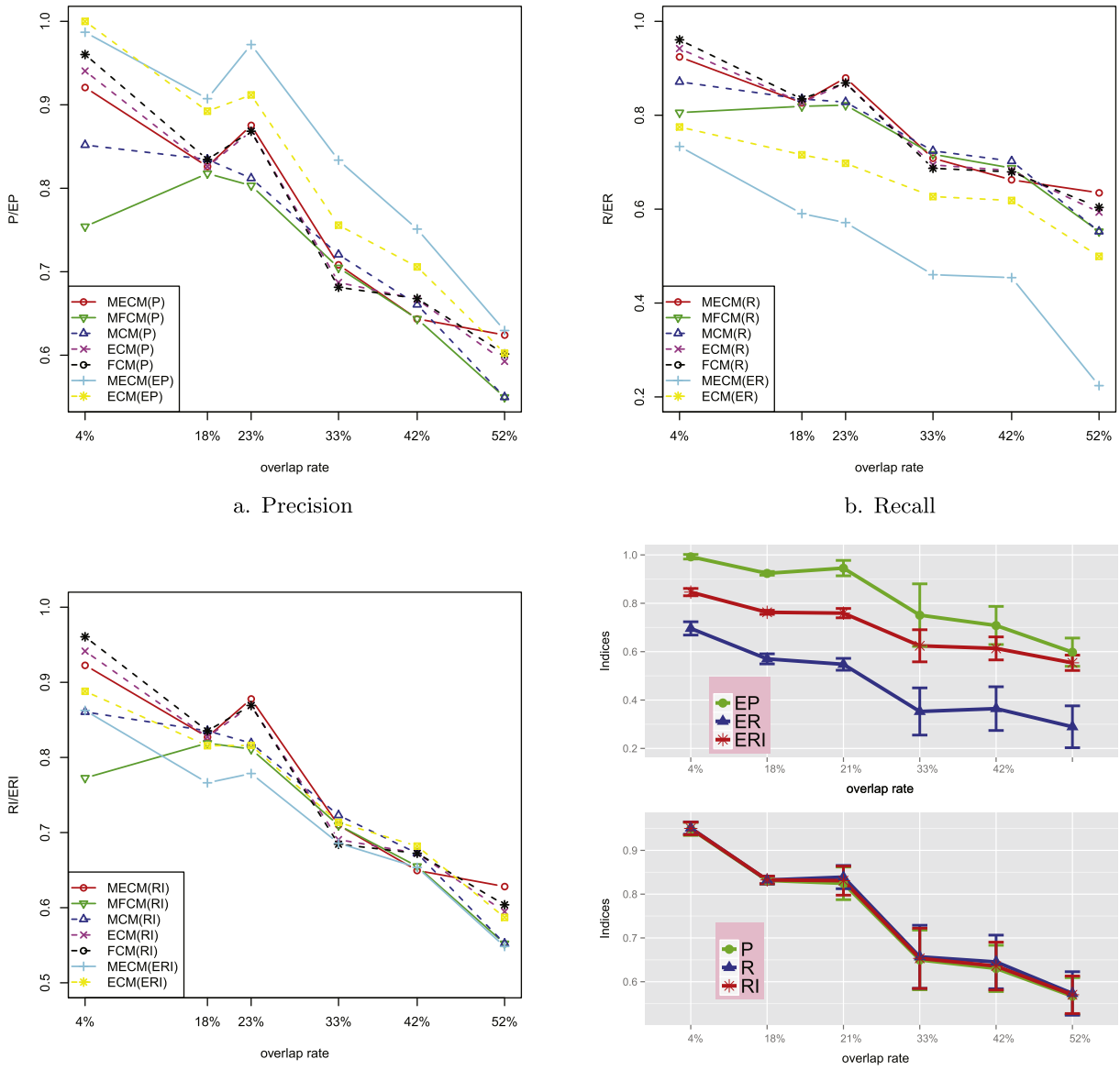


Fig. 4. Clustering of overlapped data set with different overlap rates. (d) Shows the average values of the indices (plus and minus one standard deviation) for 20 repeated experiments, as a function of the overlap rates. For MECM $\alpha = 1.8, \gamma = 0.2, \eta = 0.7, \delta = 50$.

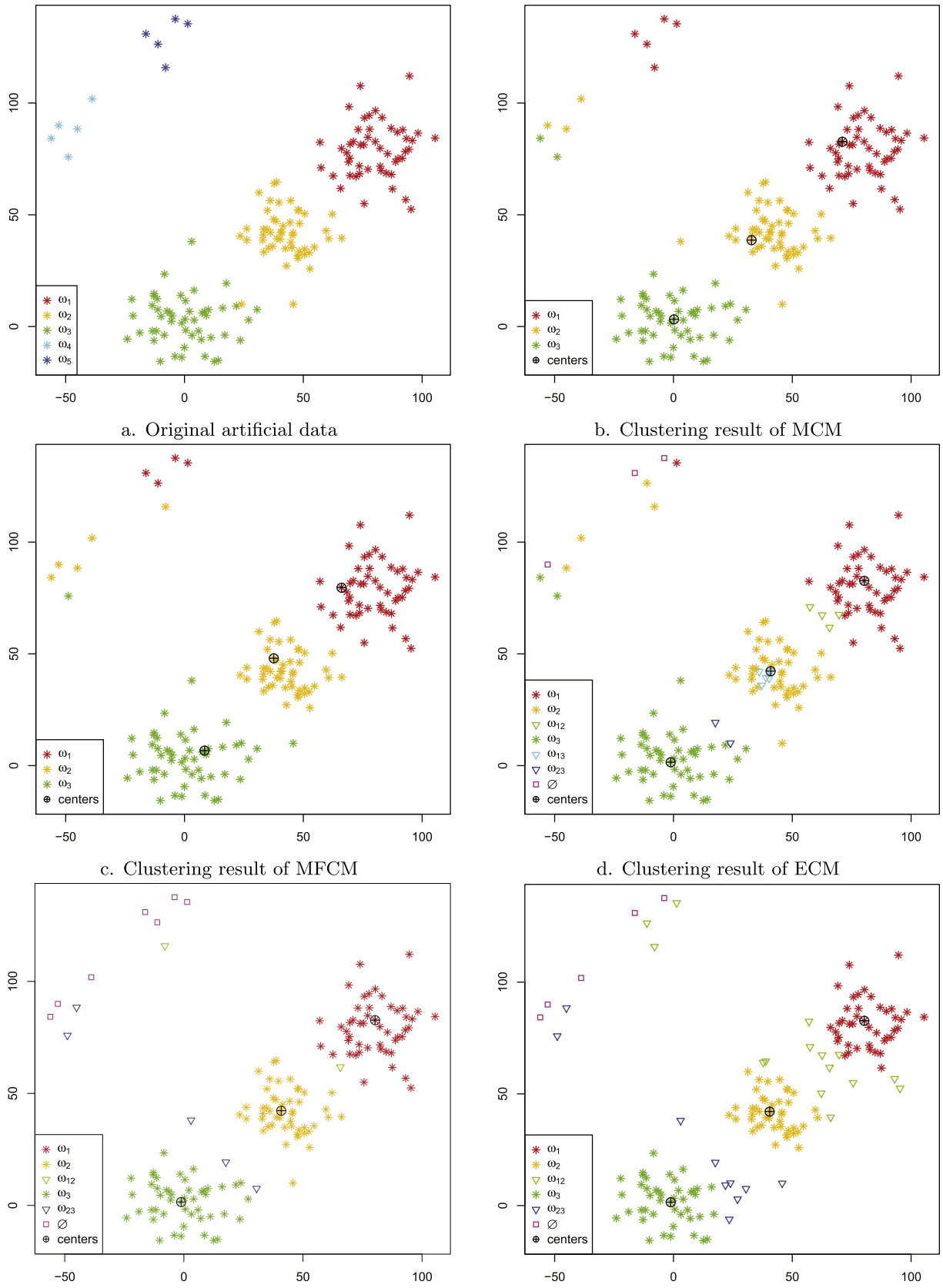
imprecise clusters, with which we do not have to partition the uncertain or unknown objects into a specific cluster. Consequently, it could reduce the risk of misclassification. However, although ECM also deals with imprecise clusters, the accuracy is not improved as much as in the case of applying MECM. As illustrated before in the case of ECM application, many objects of a specific cluster are partitioned into an irrelevant imprecise class and, as a result, the evidential precision value and ERI decrease as well.

We also test on "Iris flower", "cat cortex" and "protein" data sets [12,17,22]. The first is object data while the other two are relational data sets. Thus we compare our method with FCM and ECM for the Iris data set, and with RECM and NRFCM (Non-Euclidean Relational Fuzzy Clustering Method [19]) for the last two data sets. The results are displayed in Fig. 6.

Presented results allow us to sum up the characteristics of MECM. Firstly, one can see that the behavior of MECM is similar to ECM for traditional data. Besides, credal partitions provided by MECM allow to recover the information of crisp and fuzzy partitions. Moreover, we are able to balance influence of our uncertainty and ignorance according to the actual needs. The examples utilized before deal with classical data sets. But the superiority of MECM makes it applicable in the case of data sets for which only dissimilarity measures are known *e.g.* social networks. Thus in the following experiments, we will use some graph data to illustrate the behavior of the proposed method on the community detection problem in social networks. The dissimilarity index used here is the one brought forward by Zhou [49]. To have a fair comparison, in the following experiments, we also compare with three classical algorithms for community detection *i.e.* BGLL [4], LPA [36] and ZFCM (a fuzzy *c*-means based approach proposed by Zhang et al. [46]). The obtained community structures are compared with known performance measures, *i.e.,* NMI (Normalized Mutual Information), VI (Variation of Information) and Modularity.

5.3. Artificial graphs and generated benchmarks

To show the performance of the algorithm in detecting communities in networks, we first apply the method to a sample network



a. Original artificial data

b. Clustering result of MCM

c. Clustering result of MFCM

d. Clustering result of ECM

e. Clustering result of MECM ($\gamma = 0.6$)

f. Clustering result of MECM ($\gamma = 0.2$)

Fig. 5. Clustering of the artificial data set by different methods.

Table 2
The clustering results for gaussian points by different methods. For each method, we generate 20 data sets with the same parameters and report the mean values of the evaluation indices for all the data sets.

	Precision	Recall	RI	EP	ER	ERI
MCM	0.7802	0.9570	0.9002	0.7802	0.9570	0.9002
MFCM	0.8616	0.9797	0.9484	0.8616	0.9797	0.9484
FCM	0.8644	0.9820	0.9500	0.8644	**0.9820**	0.9500
ECM	0.8215	0.9353	0.9222	0.9069	0.8436	0.9294
MECM ($\gamma = 0.2$)	**0.8674**	**0.9855**	**0.9520**	**0.9993**	0.7721	0.9336
MECM ($\gamma = 0.6$)	0.8662	0.9851	0.9515	0.9958	0.9586	**0.9868**

generated from Gaussian mixture model. This model has been used for testing community detection approaches by Liu and Liu [29].

The artificial graph is composed of 3×50 nodes, $n_1, n_2, \ldots, n_{150}$, which are represented by 150 sample points, $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{150}$, in two-dimensional Euclidean space. There are 3×50 points generated from Gaussian distributions $G(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)(k = 1, 2, 3)$ with

$$\boldsymbol{\mu}_1 = (1.0, 4.0)^{\mathrm{T}}, \quad \boldsymbol{\mu}_2 = (2.5, 5.5)^{\mathrm{T}}, \quad \boldsymbol{\mu}_3 = (0.5, 6.0)^{\mathrm{T}}$$
$$\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}_3 = \begin{pmatrix} 0.25 & 0 \\ 0 & 0.25 \end{pmatrix}. \tag{49}$$

Then, the edges of the graph are generated by the following thresholding strategy: if $|x_i - x_j| \leqslant dist$, we set an edge between node i and node j; Otherwise the two nodes are not directly connected. The graph is shown in Fig. 7a (with $dist = 0.8$) and the dissimilarity matrix of the nodes is displayed in Fig. 7b. From the figures we can see that there are three significant communities in the graph, and some nodes in the bordering of their groups seem to be in overlapped classes as they contact with members in different communities simultaneously.

The table Table 3 lists the indices for evaluating the results. It shows that MECM performs well as the evidential precision resulting from its application is high. MECM utilization also results in decreasing the probabilities of clustering failure thanks to the

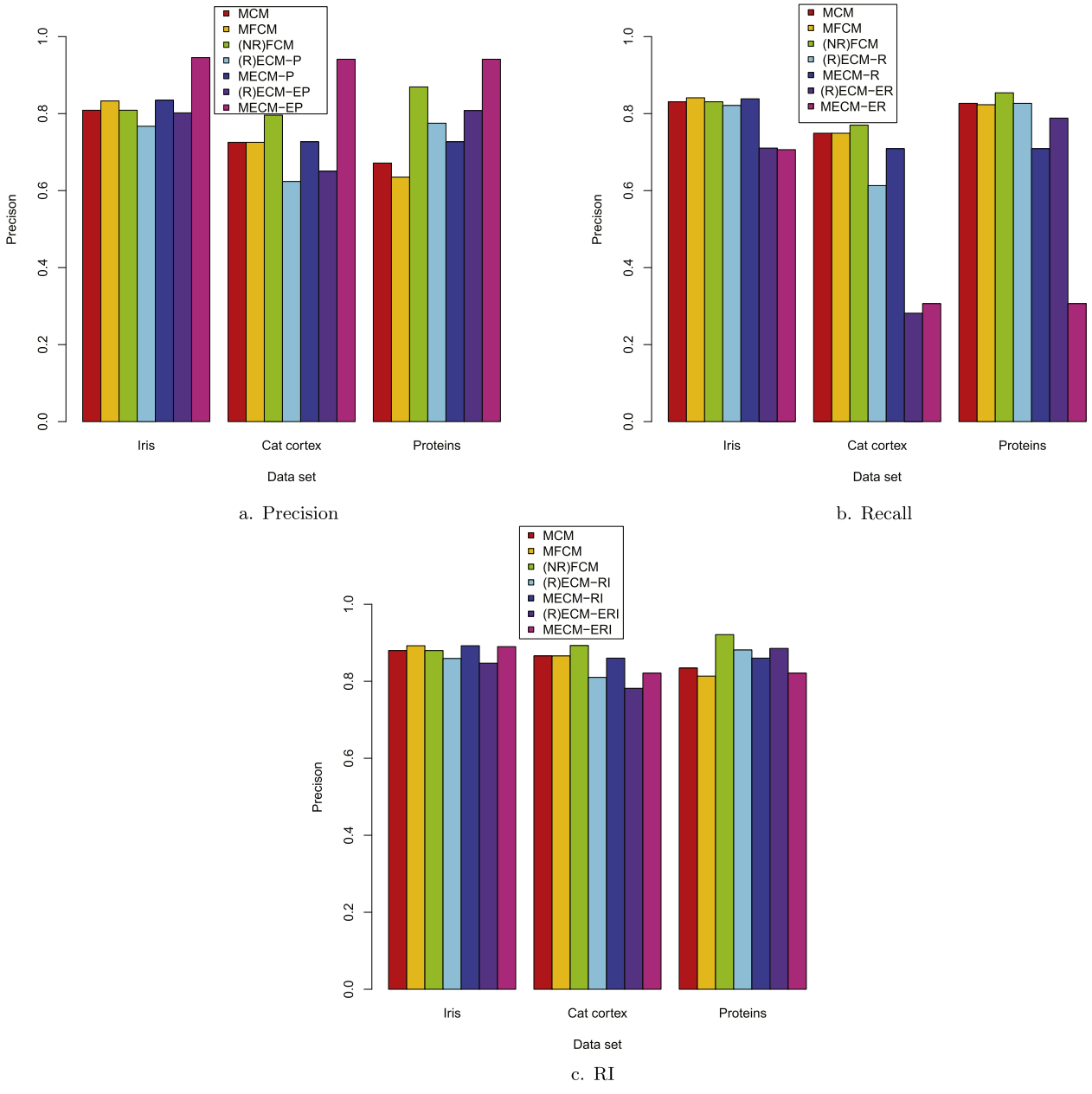


Fig. 6. Clustering results of different data sets.

introduction of imprecise clusters. This makes the decision-making process more cautious and reasonable.

The algorithms are also compared by means of Lancichinetti et al. [28] benchmark (LFR) networks. The results of different methods in two kinds of LFR networks with 500 and 1000 nodes are displayed in Figs. 8 and 9 respectively. The parameter μ showed in the x-axis in the figures identifies whether the network has clear communities. When μ is small, the graph has well community structure. In such a case, almost all the methods perform well. But we can see that when μ is large, the results by MECM have the largest values of precision. It means that the decisions which partition the nodes into a specific cluster are of great confidence. In terms of NMI, the results are similar to those by BGLL and LPA, but better than those of MCM and MFCM. This fact well explains that the hard or fuzzy partitions could be recovered when necessary.

5.4. Some real-world networks

5.4.1. A. Zachary's karate club

The Zachary's Karate Club data [45] is an The original graph and the dissimilarity of the nodes are shown in Fig. 10a and b respectively.

Let the parameters of MECM be $\alpha = 1.5, \delta = 100$, $\eta = 0.9, \gamma = 0.6$. The modularity functions by MECM, MCM, MFCM and ZFCM (Fig. 12a) peak around $c = 2$ and $c = 3$. Let $c = 2$, all the methods can detect the two communities exactly. If we set $c = 3$, a small community, which can also be found in the dissimilarity matrix (Fig. 10b), is separated from ω_1 by all the approaches (see Fig. 11). But ZFCM assigns the maximum membership to ω_1 for node 9, which is actually in ω_2. It seems that the loss of accuracy in the mapping process may cause such results.

MECM does not find imprecise groups when $\gamma = 0.6$ as the network has apparent community structure, and this reflects the fact that the communities are distinguishable for all the nodes. But there may be some overlap between two communities. The nodes in the overlapped cluster can be detected by decreasing γ (increasing the uncertainty for imprecise communities). As is displayed in Fig. 11c and d, by declining γ to 0.1 and 0.05 respectively (the other parameters remain unchanged), nodes 3 and 9 are clustered into both ω_1 and ω_2 (ω_{12}) one after another.

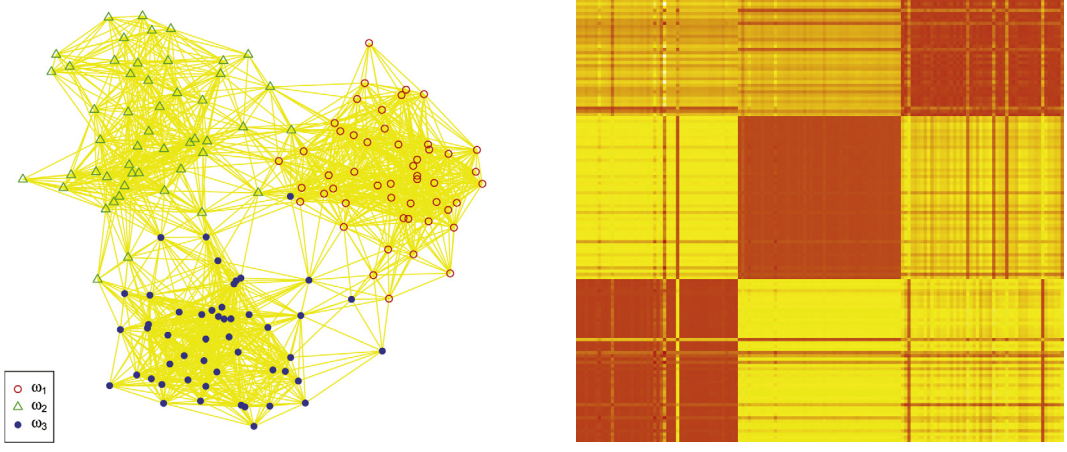
From the results we can see that MECM takes both the ignorance and the uncertainty into consideration while introducing imprecise communities. The degree of ignorance and uncertainty could be balanced through adjusting γ. The analysis shows that there appears only uncertainty without ignorance in the original club network. In order to show the performance of MECM when there are noisy conditions such that some communities are indistinguishable, two noisy nodes are added to the original graph in the next experiment.

5.4.2. B. Karate club network with some added noisy nodes

In this test, two noisy nodes are added to the original karate club network (see Fig. 13a). The first one is node 35, which is directly connected with nodes 18 and 27. The other one is 36, which is connected to nodes 1 and 33. It can be seen from the dissimilarity matrix that node 36 has stronger relationships with both communities than node 35. This is due to the fact that the nodes connected to node 36 play leader roles in their own group, but node 35 contacts with two marginal nodes with "small" or insignificant roles in their own group only.

The results obtained by the application of different methods are shown in Fig. 14. The MECM parameters are set as follows: $\alpha = 1.5, \delta = 100, \eta = 0.9$ and γ is tuned according to the extent that the imprecise communities reflect our ignorance. As we can see, MCM, MFCM and ZFCM simply group the two noisy nodes into ω_1. With $\gamma = 0.4$ MECM regards node 36 as a member of ω_1 while node 35 is grouped into imprecise community ω_{12}. And ω_{12} mainly reflects our ignorance rather than uncertainty on the actual community of node 36. This is why node 36 is not clustered into ω_{12} since ω_1 and ω_2 are distinguishable for him but we are just not sure for the final decision. The increase in the extent of uncertainty in imprecise communities results from the decrease of γ value. We can see that more nodes (including nodes 36,9,1,12,27, see Fig. 14e and f) are clustered into ω_{12} or ω_{13} due to uncertainty. The imprecise communities consider both ignorance (node 35) and uncertainty (other nodes).

These results reflect the difference between ignorance and uncertainty. As node 35 is only related to one outward node of each community, thus we are node 36 connects with the key members (playing an important role in the community), and in this case the dissimilarity between the prototypes of ω_1 and ω_2 is relatively large so they are distinguishable. Thus there is uncertainty rather than ignorance about which community node 36 is in. In this network, node 36 is a "good" member for both communities, whereas node 35 is a "poor" member. It can be seen from Fig. 15a that the fuzzy partition by MFCM also gives large similar membership values to ω_1 and ω_2 for node 35, just like in the case of such good

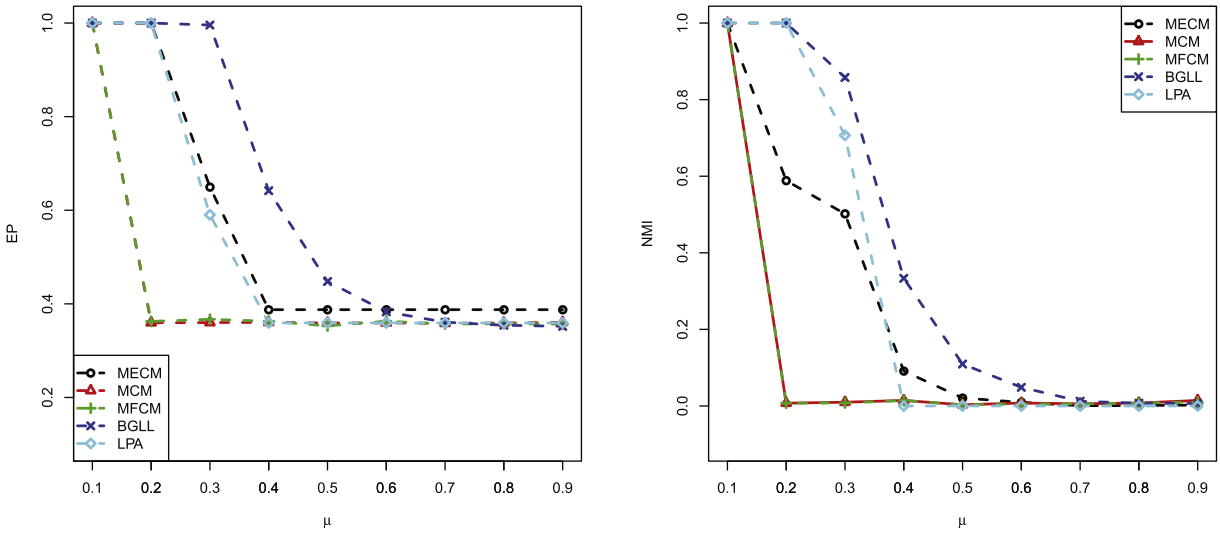


a. Gaussian network b. Dissimilarity matrix

Fig. 7. Artificial network from Gaussian mixture model.

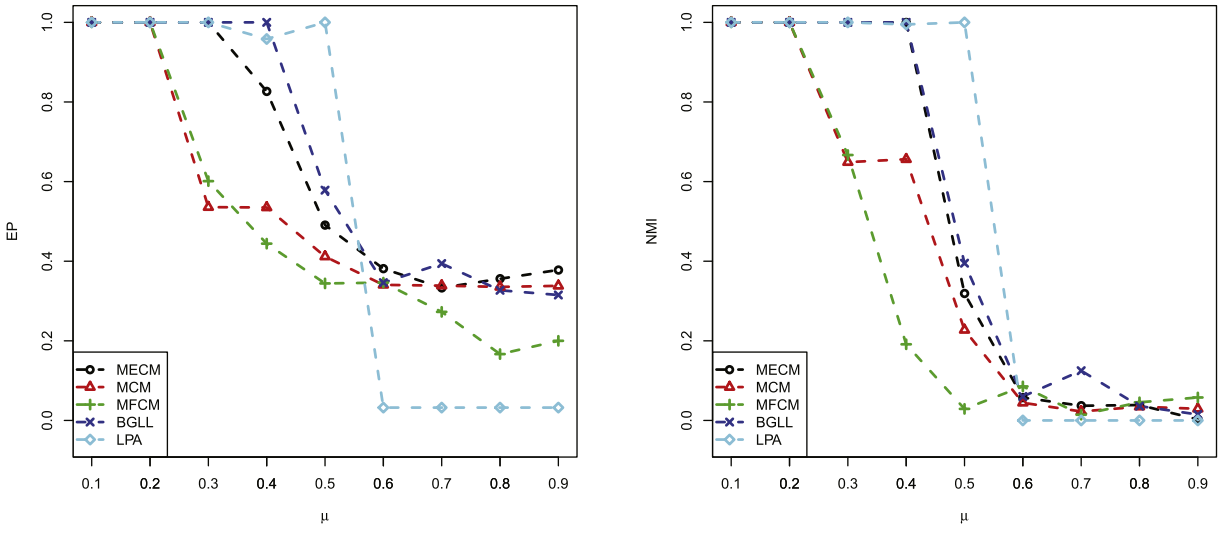
Table 3
The results for Gaussian graph by different methods.

	Precision	Recall	RI	EP	ER	ERI	NMI	VI	Modularity
MCM	0.9049	0.9110	0.9392	0.9049	0.9110	0.9392	0.8282	0.3769	0.6100
MFCM	0.9067	0.9099	0.9396	0.9067	0.9099	0.9396	0.8172	0.4013	0.6115
ZFCM	0.9202	0.9224	0.9482	0.9202	0.9224	0.9482	0.8386	0.3545	0.6118
MECM	**0.9470**	**0.9472**	**0.9652**	**0.9789**	0.6060	0.8661	**0.8895**	**0.2428**	0.6072
BGLL	0.9329	0.9347	0.9564	0.9329	0.9347	**0.9564**	0.8597	0.3081	**0.6119**
LPA	0.3289	1.0000	0.3289	0.3289	1.0000	0.3289	0.0000	1.0986	0.0000



a. Precision b. NMI

Fig. 8. Comparison of MECM and other algorithms in LFR networks. The number of nodes is $N = 500$. The average degree is $|k| = 15$, and the pair for the exponents is $(\gamma, \beta) = (2, 1)$.



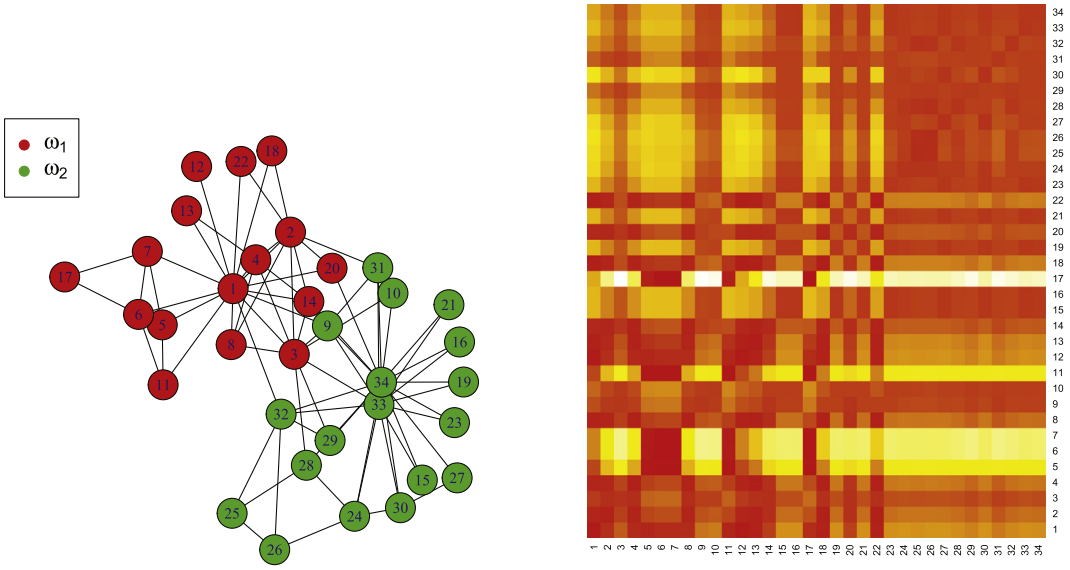
a. Precision b. NMI

Fig. 9. Comparison of MECM and other algorithms in LFR networks. The number of nodes is $N = 1000$. The average degree is $|k| = 20$, and the pair for the exponents is $(\gamma, \beta) = (2, 1)$.

members as node 36 and 9. The obtained results show the problem of distinguishing between ignorance and the "equal evidence" (uncertainty) for fuzzy partitions. But Fig. 15b shows that the credal partition by MECM assigns small mass belief to ω_1 and ω_2 for node 35, indicating our ignorance on its situation.

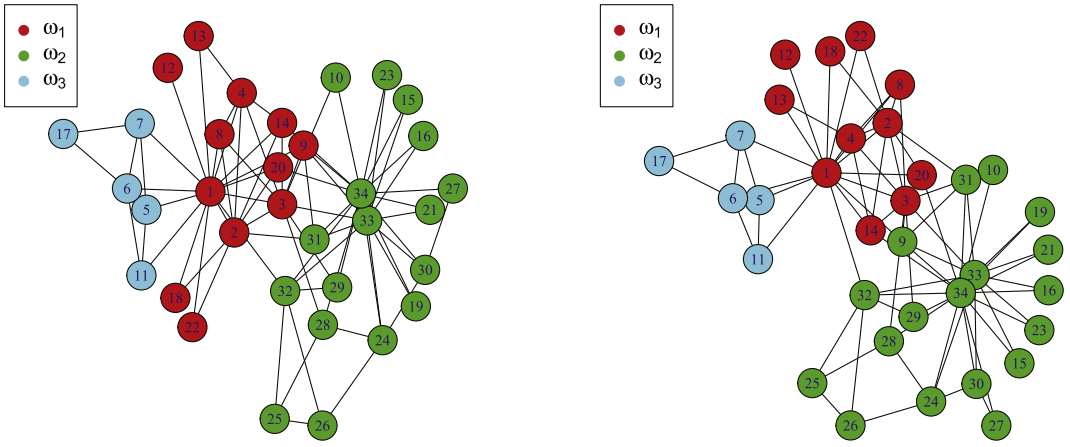
We also test our method on four other real-world graphs: American football network, Dolphins network, Lesmis network and Political books network.[1] The measures applied to evaluate the performance of different methods are listed in Table 4–7. It can been seen from the tables, for all the graphs MECM application results in a community structure with high evidential precision

[1] These data sets can be found in http://networkdata.ics.uci.edu/index.php.



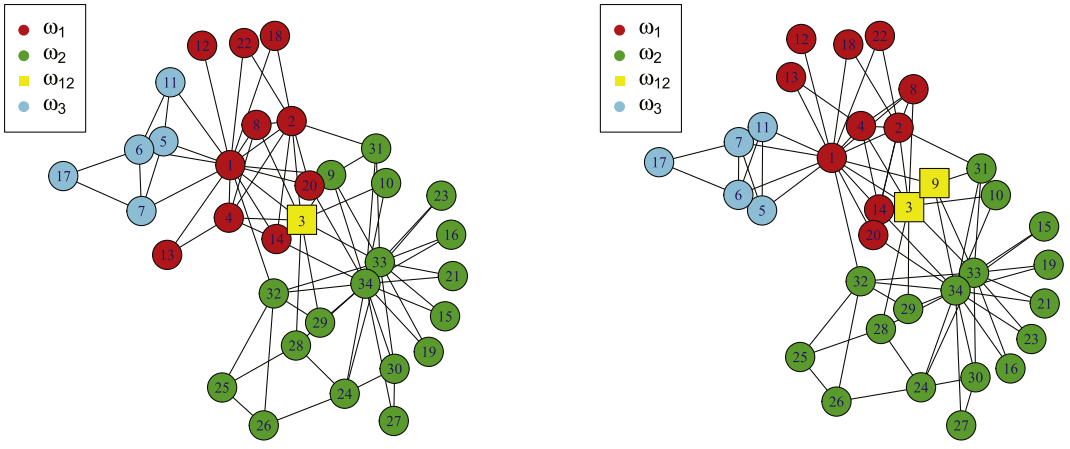
a. Original karate club network b. Dissimilarity matrix

Fig. 10. Original karate club network.



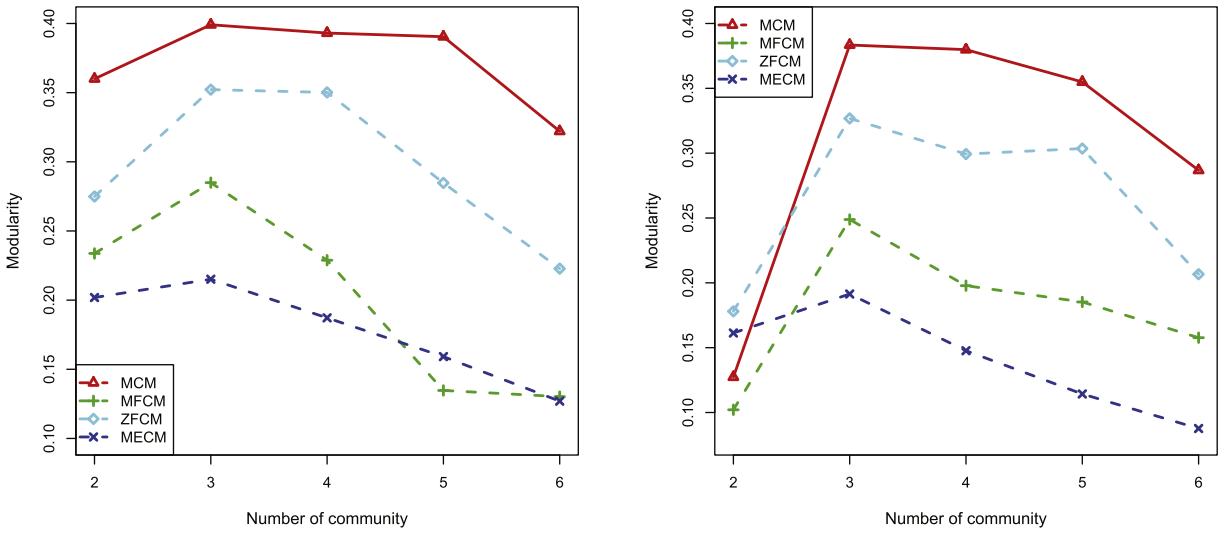
a. Clustering result of ZFCM

b. Clustering result of MCM, MFCM, and MECM ($\gamma = 0.1$)



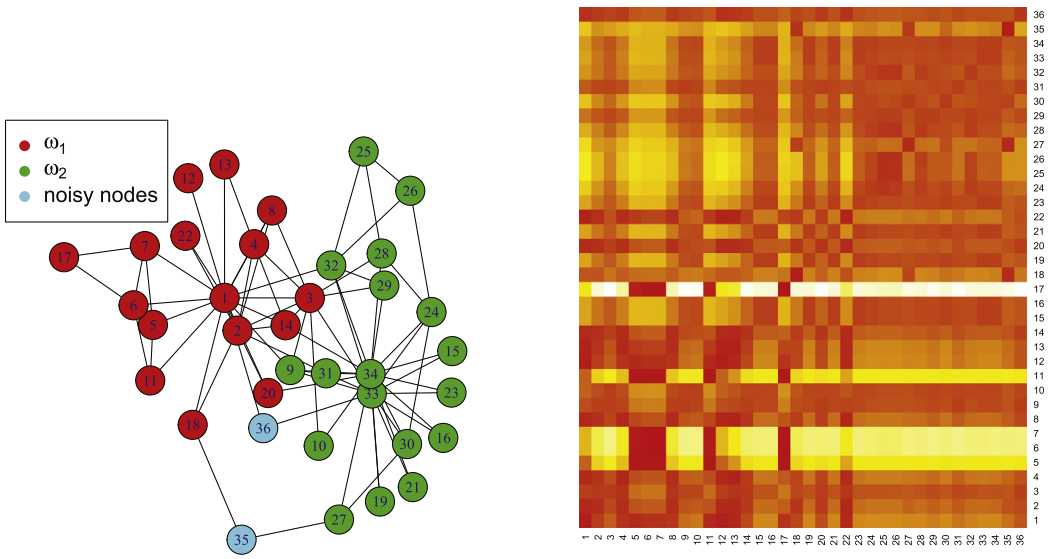
c. Clustering result of MECM ($\gamma = 0.1$) d. Clustering result of MECM ($\gamma = 0.05$)

Fig. 11. Detected communities of karate club network by different methods.



a. Original karate club b. Karate club with noisy nodes

Fig. 12. Modularity functions of karate club network by different methods.



a. Karate club network with added nodes b. Dissimilarity matrix

Fig. 13. Karate club network with two noisy nodes.

level. The precision results from a cautious decision making process which clusters the noisy nodes into imprecise communities. In terms of classical performance measures like NMI, VI and modularity, MECM slightly outperforms the other algorithms. Note that these classical measures for hard partitions are calculated by the pignistic probabilities associated with the credal partitions provided by MECM. Therefore, we can also see the possibility to recover the hard decisions here when using the proposed evidential detection approach.

5.5. Discussion

We will discuss for which application MECM is designed here. As analyzed before, for MECM only dissimilarities between objects are required and only the intuitive assumptions need to be satisfied for the dissimilarity measure. Therefore, the algorithm could be appropriate for many clustering tasks for non-metric data objects. This type of data is very common in social sciences, psychology, etc, where any metric assumptions about the similarities/dissimilarities could not be assured. The freedom for the data set leads to the restriction that the prototypes should be the objects themselves. Nevertheless, this constraint seems reasonable for social networks as the center of a community is usually the person (node) frequently contacting with others. Thus the approach can be applied to community detection problems. Thanks to the introduction of imprecise classes, it could reduce the risk of partitioning the objects which we are uncertain or ignorant into an incorrect cluster. For this reason the algorithm can help us make soft decisions when clustering the data set without distinct cluster/community structures or with overlap.

Due to the computational complexity, the proposed algorithm is not well directly adapted to handle very large data sets. However, here we discuss the possibility to apply the evidential community detection approach to large-scale networks. Firstly, the number of parameters to be optimized is exponential and depends on the number of clusters [32]. For the number of classes larger than 10,

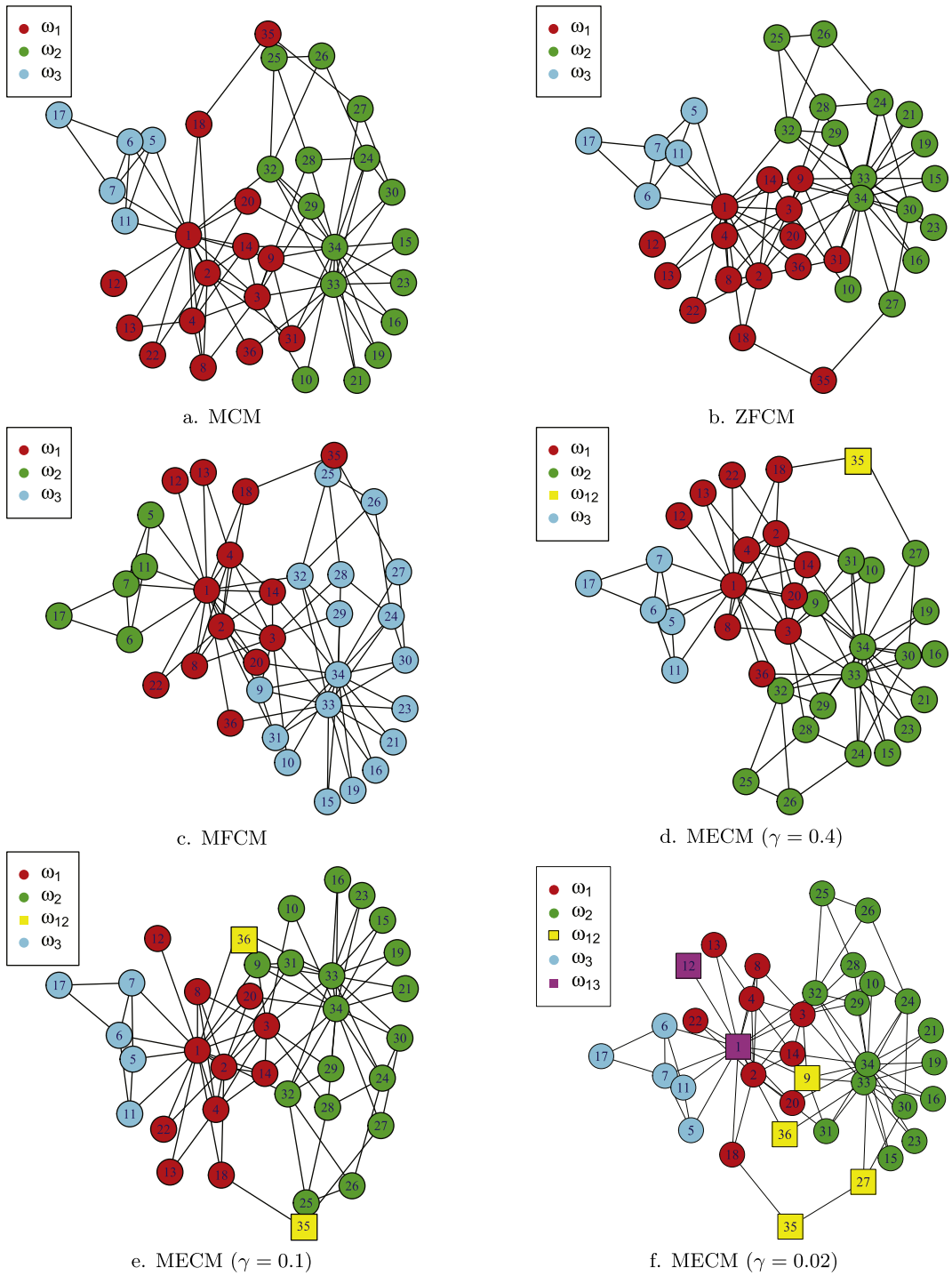
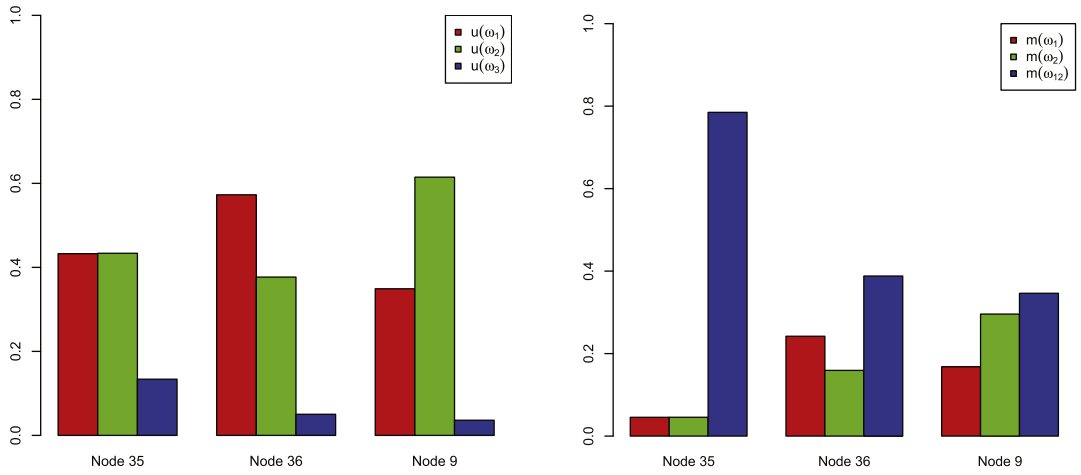


Fig. 14. Detected communities in karate club network with noisy nodes.

calculations are not tractable. But we can consider only a subclass with a limited number of focal sets [32]. For instance, we could constrain the focal sets to be composed of at most two classes (except Ω). Secondly, for the network with millions of nodes, MCM or MFCM could be evoked as a first step to merge some nodes into small clusters. After that we can apply MECM to the "coarsened" network. But how to define the edges or connections of the new graph should be studied. Lastly we emphasize that the evidential community detection algorithm could be utilized for gaining a better insight into the network structure and detecting the imprecise classes. For the large-scale network, it is difficult to make specific decisions for all of nodes due to the limitation of time, money or techniques. In this case we can use the proposed approach to make some "soft" decisions first and then use some techniques special for the imprecise parts of the graph.

6. Conclusion

We introduced a Median variant of Evidential *C*-means (MECM) as a new prototype-based clustering algorithm in the present contribution. The proposed approach is an extension of median *c*-means and median fuzzy *c*-means. It is based on the framework of belief function theory. The applied median-based clustering requires the definition of the dissimilarity between the objects



a. Fuzzy membership by MFCM b. Mass belief by MECM

Fig. 15. Fuzzy membership and mass belief of the nodes in karate club network with noisy nodes.

Table 4
The results for American football network by different methods.

	Precision	Recall	RI	EP	ER	ERI	NMI	VI	Modularity
MCM	0.7416	0.8834	0.9661	0.7416	0.8834	0.9661	0.8637	0.6467	0.5862
MFCM	0.7583	0.8757	0.9678	0.7583	0.8757	0.9678	0.8715	0.6160	0.5745
ZFCM	0.8176	0.9082	0.9765	0.8176	0.9082	0.9765	0.9035	0.4653	0.6022
MECM	**0.8232**	0.9082	**0.9771**	**0.9303**	0.8681	**0.9843**	**0.9042**	**0.4625**	0.5995
BGLL	0.7512	**0.9120**	0.9689	0.7512	**0.9120**	0.9689	0.8903	0.5195	**0.6046**
LPA	0.6698	0.8298	0.9538	0.6698	0.8298	0.9538	0.8623	0.6580	0.5757

Table 5
The results for Dolphins network by different methods.

	Precision	Recall	RI	EP	ER	ERI	NMI	VI	Modularity
MCM	1	1	1	1	1	1	1	0	0.3787
MFCM	1	1	1	1	1	1	1	0	0.3787
ZFCM	1	1	1	1	1	1	1	0	0.3787
MECM	1	1	1	1	1	1	1	0	0.3787
BGLL	0.9271	0.3583	0.6351	0.9271	0.3583	0.6351	0.4617	1.1784	0.5185
LPA	0.9250	0.5029	0.7070	0.9250	0.5029	0.7070	0.5595	0.8354	0.5070

Table 6
The results for Lesmis network by different methods.

	Precision	Recall	RI	EP	ER	ERI	NMI	VI	Modularity
MCM	0.6109	0.5522	0.9005	0.6109	0.5522	0.9005	0.7381	1.1295	0.4732
MFCM	0.5774	0.6456	0.8971	0.5774	0.6456	0.8971	0.7743	0.9555	0.4705
ZFCM	**0.7368**	0.5769	0.9217	0.7368	0.5769	0.9217	0.7805	0.9666	0.4983
MECM	0.7065	0.7473	**0.9299**	**0.9298**	0.4368	**0.9258**	**0.7977**	**0.8531**	0.4884
BGLL	0.5796	0.8104	0.9033	0.5796	0.8104	0.9033	0.7551	0.9435	**0.5556**
LPA	0.4594	**0.9643**	0.8544	0.4594	**0.9643**	0.8544	0.7500	0.8637	0.5428

Table 7
The results for Political books network by different methods.

	Precision	Recall	RI	EP	ER	ERI	NMI	VI	Modularity
MCM	0.8109	0.8030	0.8482	0.8109	0.8030	0.8482	0.5721	0.8426	0.4979
MFCM	0.8020	0.8187	**0.8485**	0.8020	0.8187	**0.8485**	**0.5755**	0.8256	0.4962
ZFCM	0.7928	0.7487	0.8234	0.7928	0.7487	0.8234	0.5301	0.9407	0.5048
MECM	0.7880	0.8081	0.8383	**0.8458**	0.6435	0.8128	**0.5755**	0.8247	0.4725
BGLL	**0.8244**	0.6203	0.7978	0.8244	0.6203	0.7978	0.5121	1.0987	**0.5205**
LPA	0.7331	**0.8558**	0.8200	0.7331	**0.8558**	0.8200	0.5612	**0.7925**	0.4604

only. Therefore, it is not restricted to a metric space application. The prototypes of the clusters are constrained to the data objects themselves. MECM provides us with not only credal partitions but also hard and fuzzy partitions as by-products through computing pignistic probabilities. Moreover, it could distinguish ignorance from uncertainty while the fuzzy or crisp partitions could not. By the introduced imprecise clusters, we could find some overlapped and indistinguishable clusters for related nodes. Thanks to the advantages of belief function theory and median clustering, MECM could be applied to community detection problems in social networks. As other median clustering approaches, MECM tends to get stuck in local minima such that several runs have to be performed to obtain good performance. However, we propose an initial prototype-selection scheme using the evidential semi-centrality for the application of MECM in community detection to solve the problems brought by the initial prototypes. Results of presented experiments on artificial and real-world networks show that the credal partitions on graphs provided by MECM application are more refined than crisp and fuzzy ones. Therefore, they could enable us to gain a better understanding of analyzed community structure. Some examples on the classical metric space are also given to illustrate the interest of MECM and to show its difference with respect to the existing methods.

As mentioned in this paper, there may be more than one center in each community network. Nevertheless, we ignore "multi-center" to avoid the troubles brought by the need for an initial seed using ESC and the definition of a threshold to control the distance between prototypes. We are aware that this is a drawback of the presented approach as not all the centers in each community are taken into consideration. Therefore, we intend to include the feature of multiple centers in our future research work.

Acknowledgements

The authors are grateful to the anonymous reviewers for all their remarks which helped us to clarify and improve the quality of this paper. This work was supported by the National Natural Science Foundation of China (Nos. 61135001, 61403310).

References

[1] B. Amiri, L. Hossain, J.W. Crawford, R.T. Wigand, Community detection in complex networks: multi-objective enhanced firefly algorithm, Knowl.-Based Syst. 46 (2013) 1–11.
[2] D. Arthur, S. Vassilvitskii, *k*-means++: the advantages of careful seeding, in: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.
[3] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Kluwer Academic Publishers., 1981.
[4] V.D. Blondel, J.-L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, J. Stat. Mech.: Theory Exp. 2008 (10) (2008) P10008.
[5] G. Bordogna, G. Pasi, A quality driven hierarchical data divisive soft clustering for information retrieval, Knowl.-Based Syst. 26 (2012) 9–19.
[6] C. Borgelt, Prototype-based classification and clustering, Ph.D. thesis, Otto-von-Guericke-Universität Magdeburg, Universitätsbibliothek, 2006.
[7] A. Clauset, M.E. Newman, C. Moore, Finding community structure in very large networks, Phys. Rev. E 70 (6) (2004) 066111.
[8] M. Cottrell, B. Hammer, A. Hasenfuß, T. Villmann, Batch and median neural gas, Neural Netw. 19 (6) (2006) 762–771.
[9] T. Denœux, M.-H. Masson, EVCLUS: evidential clustering of proximity data, IEEE Trans. Syst. Man Cybernet. Part B: Cybernet. 34 (1) (2004) 95–109.
[10] J. Duch, A. Arenas, Community detection in complex networks using extremal optimization, Phys. Rev. E 72 (2) (2005) 027104.
[11] J.C. Dunn, A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters, J. Cybernet. 3 (1973) 32–57.
[12] R.A. Fisher, The use of multiple measurements in taxonomic problems, Ann. Eugen. 7 (2) (1936) 179–188.
[13] S. Fortunato, Community detection in graphs, Phys. Rep. 486 (3) (2010) 75–174.
[14] S. Fortunato, M. Barthelemy, Resolution limit in community detection, Proc. Natl. Acad. Sci. 104 (1) (2007) 36–41.
[15] C. Gao, D. Wei, Y. Hu, S. Mahadevan, Y. Deng, A modified evidential methodology of identifying influential nodes in weighted networks, Physica A: Stat. Mech. Its Appl. 392 (21) (2013) 5490–5500.
[16] T. Geweniger, D. Zülke, B. Hammer, T. Villmann, Median fuzzy *c*-means for clustering dissimilarity data, Neurocomputing 73 (7) (2010) 1109–1116.
[17] T. Graepel, R. Herbrich, P. Bollmann-Sdorra, K. Obermayer, Classification on pairwise proximity data, Adv. Neural Inform. Process. Syst. (1999) 438–444.
[18] B. Hammer, A. Hasenfuss, Relational neural gas, in: KI 2007: Advances in Artificial Intelligence, Springer, 2007, pp. 190–204.
[19] R.J. Hathaway, J.C. Bezdek, Nerf *c*-means: non-euclidean relational fuzzy clustering, Pattern Recogn. 27 (3) (1994) 429–437.
[20] R.J. Hathaway, J.W. Davenport, J.C. Bezdek, Relational duals of the *c*-means clustering algorithms, Pattern Recogn. 22 (2) (1989) 205–212.
[21] T. Havens, J. Bezdek, C. Leckie, K. Ramamohanarao, M. Palaniswami, A soft modularity function for detecting fuzzy communities in social networks, IEEE Trans. Fuzzy Syst. 21 (6) (2013) 1170–1175.
[22] T. Hofmann, J.M. Buhmann, Pairwise data clustering by deterministic annealing, IEEE Trans. Pattern Anal. Machine Intell. 19 (1) (1997) 1–14.
[23] Y. Hu, M. Li, P. Zhang, Y. Fan, Z. Di, Community detection by signaling on complex networks, Phys. Rev. E 78 (1) (2008) 016115.
[24] M.Z. Islam, L. Brankovic, Privacy preserving data mining: a noise addition framework using a novel clustering technique, Knowl.-Based Syst. 24 (8) (2011) 1214–1223.
[25] A.K. Jain, Data clustering: 50 years beyond *k*-means, Pattern Recogn. Lett. 31 (8) (2010) 651–666.
[26] Y. Jiang, C. Jia, J. Yu, An efficient community detection method based on rank centrality, Physica A: Stat. Mech. Its Appl. (2012).
[27] A. Lancichinetti, S. Fortunato, J. Kertész, Detecting the overlapping and hierarchical community structure in complex networks, New J. Phys. 11 (3) (2009) 033015.
[28] A. Lancichinetti, S. Fortunato, F. Radicchi, Benchmark graphs for testing community detection algorithms, Phys. Rev. E 78 (4) (2008) 046110.
[29] J. Liu, T. Liu, Detecting community structure in complex networks using simulated annealing with *k*-means algorithms, Physica A: Stat. Mech. Its Appl. 389 (11) (2010) 2300–2309.
[30] Z.-g. Liu, Q. Pan, J. Dezert, Evidential classifier for imprecise data based on belief functions, Knowl.-Based Syst. 52 (2013) 246–257.
[31] A. Martin, I. Quidu, Decision support with belief functions theory for seabed characterization, in: 2008 11th International Conference on Information Fusion, IEEE, 2008, pp. 1–8.
[32] M.-H. Masson, T. Denoeux, ECM: an evidential version of the fuzzy *c*-means algorithm, Pattern Recogn. 41 (4) (2008) 1384–1397.
[33] M.-H. Masson, T. Denœux, RECM: relational evidential *c*-means algorithm, Pattern Recogn. Lett. 30 (11) (2009) 1015–1026.
[34] M. Mendes, L. Sacks, Evaluating fuzzy clustering for relevance-based information access, The 12th IEEE International Conference on Fuzzy Systems, 2003 (FUZZ'03), vol. 1, IEEE, 2003, pp. 648–653.
[35] M.E. Newman, M. Girvan, Finding and evaluating community structure in networks, Phys. Rev. E 69 (2) (2004) 026113.
[36] U.N. Raghavan, R. Albert, S. Kumara, Near linear time algorithm to detect community structures in large-scale networks, Phys. Rev. E 76 (3) (2007) 036106.
[37] M. Sales-Pardo, R. Guimera, A.A. Moreira, L.A.N. Amaral, Extracting the hierarchical organization of complex systems, Proc. Natl. Acad. Sci. 104 (39) (2007) 15224–15229.
[38] J. Schubert, Clustering belief functions based on attracting and conflicting metalevel evidence using potts spin mean field theory, Inform. Fusion 5 (4) (2004) 309–318.
[39] P. Smets, Decision making in the TBM: the necessity of the pignistic transformation, Int. J. Approx. Reason. 38 (2) (2005) 133–147.
[40] S. Smyth, S. White, A spectral clustering approach to finding communities in graphs, in: Proceedings of the 5th SIAM International Conference on Data Mining, 2005, pp. 76–84.
[41] M. Tabassian, R. Ghaderi, R. Ebrahimpour, Combining complementary information sources in the Dempster–Shafer framework for solving classification problems with imperfect labels, Knowl.-Based Syst. 27 (2012) 92–102.
[42] M. Tasgin, A. Herdagdelen, H. Bingol, Community detection in complex networks using genetic algorithms, arXiv preprint arXiv:0711.0491, 2007.
[43] B. Yang, J. Di, J. Liu, D. Liu, Hierarchical community detection with applications to real-world network analysis, Data Knowl. Eng. 83 (2013) 20–38.
[44] X. Yang, G. Zhang, J. Lu, J. Ma, A kernel fuzzy *c*-means clustering-based fuzzy support vector machine algorithm for classification problems with outliers or noises, IEEE Trans. Fuzzy Syst. 19 (1) (2011) 105–115.
[45] W.W. Zachary, An information flow model for conflict and fission in small groups, J. Anthropol. Res. (1977) 452–473.
[46] S. Zhang, R.-S. Wang, X.-S. Zhang, Identification of overlapping community structure in complex networks using fuzzy *c*-means clustering, Physica A: Stat. Mech. Its Appl. 374 (1) (2007) 483–490.
[47] W. Zhang, T. Yoshida, X. Tang, Q. Wang, Text clustering using frequent itemsets, Knowl.-Based Syst. 23 (5) (2010) 379–388.
[48] X. Zhang, J. Zhu, Q. Wang, H. Zhao, Identifying influential nodes in complex networks with community structure, Knowl.-Based Syst. 42 (2013) 74–84.
[49] H. Zhou, Distance, dissimilarity index, and network community structure, Phys. Rev. E 67 (6) (2003) 061901.
[50] K. Zhou, A. Martin, Q. Pan, Evidential communities for complex networks, in: Information Processing and Management of Uncertainty in Knowledge-Based Systems, Springer, 2014, pp. 557–566.