Fuzzy relational clustering around medoids: A unified view

Jian-Ping Mei, Lihui Chen*

Division of Information Engineering, School of EEE, Nanyang Technological University, Singapore 639798, Republic of Singapore

Received 8 July 2010; received in revised form 9 June 2011; accepted 11 June 2011
Available online 29 June 2011

Abstract

Medoid-based fuzzy clustering generates clusters of objects based on relational data, which records pairwise similarities or dissimilarities among objects. Compared with single-medoid based approaches, our recently proposed fuzzy clustering with multiple-weighted medoids has shown superior performance in clustering via experimental study. In this paper, we present a new version of fuzzy relational clustering in this family called fuzzy clustering with multi-medoids (FMMdd). Based on the new objective function of FMMdd, update equations can be derived more conveniently. Moreover, a unified view of FMMdd and two existing fuzzy relational approaches fuzzy c-medoids (FCMMdd) and assignment-prototype (A-P) can be established, which allows us to conduct further analytical study to investigate the effectiveness and feasibility of the proposed approach as well as the limitations of existing ones. The robustness of FMMdd is also investigated. Our theoretical and numerical studies show that the proposed approach produces good quality of clusters with rich cluster-based information and it is less sensitive to noise.

Keywords: Fuzzy clustering; Relational data; Weighted medoids

1. Introduction

As one of the most popular fuzzy clustering approaches, fuzzy c-means (FCM) [1] has been extensively studied for many years. In FCM, objects are clustered based on the object data, where each object in the dataset is represented as a vector in a feature space. The success of FCM and its extensions encourages the development of fuzzy partitioning clustering based on relational data, which records the degrees to which pairs of objects in the data set are related [3–11]. Typically, relational data are represented as pairwise similarities or dissimilarities. Among these existing (dis)similarity-based fuzzy clustering approaches, most of them, including relational fuzzy c-means (RFCM) [3], fuzzy analysis (FANNY) [5] and fuzzy relational clustering (FRC) [6], only generate a fuzzy partition of the target dataset at the end of the clustering process. The RFCM approach is derived as a relational dual of fuzzy c-means for “Euclidean type” data, i.e., the values in the dissimilarity matrix match the pairwise Euclidean distances of objects in some p-dimensional space [3]. The non-Euclidean RFCM (NERFCM) [4] is an improved version of RFCM, and FRC [6] is another closely related approach of RFCM. When the relational data are “Euclidean type”, FRC and NERFCM are equivalent to RFCM; otherwise, FRC and NERFCM adopt different ways for handling the “non-Euclidean” situation. Recently, it has been

* Corresponding author. Tel.: +65 67904484; fax: +65 67933318.
E-mail address: elhchen@ntu.edu.sg (L. Chen).

0165-0114/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.fss.2011.06.009
shown in [12] that the framework of NERFCM can be further extended to handle non-linearly separable clusters with kernelized relation matrix. In these approaches, fuzzy membership is the only type of variable to be updated and evaluated, which means these approaches only produce the partition of the data through finding suitable fuzzy memberships for each of the objects to all the $K$ clusters. Similar to fuzzy c-means, these fuzzy relational clustering approaches are sensitive to noise. In [6], Davé and Sen aim to enhance the robustness of this type of fuzzy relational clustering by integrating an additional “noise cluster”. Similar to the derivation of RFCM from fuzzy c-means, relational duals of possibilistic clustering are reported in [13,14]. Instead of designing clustering algorithms for relational data, some researchers consider clustering relational data by explicitly or implicitly projecting relational data onto object data, so that object data clustering can then be performed [15,16]. It has also been proposed that relational data may be interpreted as object data, where FCM can be directly applied [17,18].

Another group of fuzzy relational clustering approaches are formulated with a connection to the k-medoids or c-medoids approach. As a fuzzy extension of k-medoids, the fuzzy c-medoids (FCMdd) [8] approach produces fuzzy clusters where each of them is represented by a representative object or medoid. However, in some cases, it may not be sufficient enough to use only one object to represent the whole cluster. In a recent work [11], a generalized medoid-based fuzzy clustering called prototype-weights based fuzzy clustering (PFC) has been proposed. In PFC, each cluster is represented by more than one objects associated with different representative weights. The representative weight, also called “prototype weight” was first introduced in an algorithm called assignment-prototype (A-P) [7]. However, the benefit of using prototype weight has not been clearly discussed nor fully investigated in [7], and an even critical issue of A-P, as reported in [19], is that it only performs well on some datasets but fails to provide any reasonable results on many others. The effectiveness of using multiple representative objects to characterize a cluster has been further explored in PFC [11]. However, we are unable to establish a direct connection between PFC and existing medoid-based approaches.

Compared with those partition-focused approaches, medoid-based ones can handle any form of relational data without additional computations, and they provide additional information on the relative importance or representativeness of objects in each cluster, which is very useful for a better understanding and a description of each of the clusters produced. The study in [11] has already shown that multiple-weighted medoids are more powerful than a single medoid for representing a cluster. In this paper, we investigate the relationships among different medoid-based approaches. To do that, we first present a new version of fuzzy clustering with multiple-weighted medoids called FMMdd for relational data. Based on the new objective function of FMMdd, the update equations can be derived conveniently, and a unified view of FMMdd, FCMdd and A-P can be established. This unified view helps us to see the relationships among different medoid-based approaches as well as the improvement made in the formulation of the new approach compared with these existing ones. The effectiveness and robustness of FMMdd are discussed and also illustrated through experiments.

In the next section, we first present the formulation and solution of the new approach FMMdd. After that, we give discussions and analysis of the favorable properties of the proposed FCMdd approach and the relationships between FCMdd and some existing approaches in Section 3. To illustrate and verify our theoretical analysis and conclusions, numerical studies are reported and discussed in Section 4. Finally, we conclude this study in Section 5.

2. Proposed approach

In this section, a new objective function is presented for fuzzy relational clustering with weighted medoids. The dissimilarity-based clustering is formulated as a constrained minimization problem. An iterative algorithm is derived for finding local solutions of the proposed objective.

For a dataset $\mathcal{X} = \{x_i\}_{i=1,2,...,N}$, given the dissimilarity matrix $R$, where $r_{ij} \geq 0 \in R$ records the dissimilarity between each two objects $x_i$ and $x_j$, and the number of clusters $K$, the objective of the proposed fuzzy clustering with multi-medoids (FMMdd) is to minimize the following criterion:

$$J_{FMMdd}(U, V) = \sum_{c=1}^{K} \sum_{i=1}^{N} \sum_{j=1}^{N} u_{ci}^{m} v_{cj}^{n} r_{ij}$$ (1)
subject to

\[ \sum_{c=1}^{K} u_{ci} = 1, \quad \forall i, \quad (2) \]

\[ u_{ci} \geq 0, \quad \forall c \text{ and } i, \quad (3) \]

\[ \sum_{j=1}^{N} v_{cj} = 1, \quad \forall c, \quad (4) \]

\[ v_{cj} \geq 0, \quad \forall c \text{ and } j, \quad (5) \]

where \( u_{ci} \) denotes the fuzzy membership of \( x_i \) in cluster \( c \), and \( v_{cj} \) denotes the prototype weight of \( x_j \) in cluster \( c \). The fuzzy membership allows each object to belong to several clusters with various degrees, and the representative weight or prototype weight allows each cluster to be represented by multiple objects, which are weighted based on their importance in a cluster. The larger \( u_{ci} \) is, the more possible that \( x_i \) belongs to cluster \( c \). The larger \( v_{cj} \) is, the better that \( x_j \) represents cluster \( c \). We use matrices \( \mathbf{U}_{K \times N} \) and \( \mathbf{V}_{K \times N} \) to record the memberships and prototype weights of \( N \) objects with respect to all the \( K \) clusters. From the definition, \( \mathbf{U} \) corresponds to a fuzzy partition of the whole dataset and \( \mathbf{V} \) is used to describe and represent each cluster. From \( \mathbf{V} \), a more detailed description on the internal structure of each of the clusters can be found, which means cluster-based information, such as the ranks of objects in each cluster, is available. Parameter \( m \) controls the fuzziness of memberships, and \( n \) controls the level of smoothness of the distribution of prototype weights among all the objects in each of the clusters. In order to let \( \mathbf{U} \) and \( \mathbf{V} \) play different roles in clustering, i.e., one for partition and the other for cluster representation, we pose constraints on \( \mathbf{U} \) and \( \mathbf{V} \) in two different ways as given in (2) and (4), respectively. The constraints on \( \mathbf{U} \) is for each object and on \( \mathbf{V} \) is for each cluster. More analytical study on how \( \mathbf{U} \) and \( \mathbf{V} \) characterize different types of object-to-cluster relationships during the clustering process will be given in Section 3.1.

Now we need to solve the above minimization problem by finding the optimal \( \mathbf{U} \) and \( \mathbf{V} \) subject to constraints. Since \( u_{ci} \) and \( v_{cj} \) are continuous variables, we use the method of Lagrange multipliers to derive the solutions. The Lagrangian function is first constructed as

\[ L_{\text{FMMdd}} = \sum_{c=1}^{K} \sum_{i=1}^{N} \sum_{j=1}^{N} u_{ci} v_{cj} r_{ij} - \sum_{i=1}^{N} \lambda_i \left( \sum_{c=1}^{K} u_{ci} - 1 \right) - \sum_{c=1}^{K} \beta_c \left( \sum_{j=1}^{N} v_{cj} - 1 \right), \quad (6) \]

where \( \lambda_i \) and \( \beta_c \) are the Lagrange multipliers. By calculating \( \partial L_{\text{FMMdd}} / \partial u_{ci} = 0 \), \( \partial L_{\text{FMMdd}} / \partial \lambda_i = 0 \), \( \partial L_{\text{FMMdd}} / \partial v_{cj} = 0 \), and \( \partial L_{\text{FMMdd}} / \partial \beta_c = 0 \), the update equations of \( u_{ci} \) and \( v_{cj} \) are derived as below:

\[ u_{ci} = \frac{\left( \sum_{j=1}^{N} v_{cj}^m r_{ij} \right)^{1/(m-1)}}{\sum_{f=1}^{K} \left( \sum_{j=1}^{N} v_{cj}^m r_{ij} \right)^{1/(m-1)}}, \quad (7) \]

\[ v_{cj} = \frac{\left( \sum_{i=1}^{N} u_{ci}^m r_{ij} \right)^{1/(n-1)}}{\sum_{h=1}^{K} \left( \sum_{i=1}^{N} u_{ci}^m r_{ij} \right)^{1/(n-1)}}. \quad (8) \]

Now the FMMdd algorithm can be stated as follows: starting with a non-negative initialization, \( \mathbf{U} \) and \( \mathbf{V} \) are iteratively updated with (7) and (8), respectively, in an alternating manner until the successive estimates of \( \mathbf{U} \) are close enough or the maximum number of iterations is reached. During the iteration process, the quality of the partition in terms of the criterion defined in (1) is successively improved through reassignment of objects to clusters based on the current cluster representation and reselection of cluster representatives based on the current partition. Our experiments show that this FMMdd algorithm usually converges rapidly.

In [11], final update rules of \( \mathbf{U} \) and \( \mathbf{V} \) need to be derived with the Karush–Kuhn–Tucker conditions to consider the non-negative constraints when constructing the Lagrangian. However here, it is straightforward to see that the above two formulas (7) and (8) are able to preserve non-negativity, which means that during the iteration process, the values of \( u_{ci} \) and \( v_{cj} \) are always non-negative as long as a non-negative initialization is used. Therefore, the non-negative
constraints given in (3) and (5) are automatically satisfied. We can view FMMdd as an alternative approach of fuzzy clustering with weighted medoids, which offers a unified view for the theoretical discussions of other existing fuzzy relational clustering approaches. Given an $N \times N$ relation matrix, the time complexity of FMMdd is $O(N^2)$, which is the same as that of other fuzzy relational clustering approaches.

3. Discussion

Based on the new formulation presented in the previous section, now we provide further discussions on some unique properties of FMMdd. We also show that FMMdd is a generalization of two existing fuzzy approaches.

3.1. A close look of update rules in FMMdd

In (1) we want the two output matrices $U$ and $V$ to capture the partition and the cluster-based representativeness, respectively. Now we take a close look at the update rules of membership and prototype weight to explore the different roles that they play in the clustering process. To make it easier to understand, we denote

$$v_{nc} = (v^n_{c1}, v^n_{c2}, ..., v^n_{cN})^T,$$

$$r_l = (r_{l1}, r_{l2}, ..., r_{lN})^T,$$

and define

$$z_c = v^n_c$$  \hspace{1cm} (9)

and

$$a(x_i, z_c) = \sum_{j=1}^{N} z_{cj} r_{lj} = z^T_c r_l.$$  \hspace{1cm} (10)

With these definitions, the update formula of $u_{ci}$ in (7) can be expressed as

$$u_{ci} = \frac{a(x_i, z_c)^{-1/(m-1)}}{\sum_{f=1}^{K} a(x_i, z_f)^{-1/(m-1)}}.$$  \hspace{1cm} (11)

Since $z_c$ in (9) is a vector carrying the representativeness of each object in cluster $c$, it could be regarded as a “center-vector” of that cluster, of which each dimension is the weight of a representative object. Compared to conventional one-medoid representation of a cluster, such kind of a “center-vector” enlarges the space of the representative of a cluster from the discrete sample space to a continuous one. Since cluster $c$ is characterized by $z_c$, we may consider $a(x_i, z_c)$ in (10) as the dissimilarity between $x_i$ and cluster $c$. From (11), it is shown that the fuzzy membership $u_{ci}$ of $x_i$ in a cluster $c$ is a relative object-to-cluster dissimilarity, which is dependent on $a(x_i, z_f)$, where $f \neq c$, i.e., the dissimilarities between this object $x_i$ and clusters other than $c$. This is the essential property of fuzzy membership for interpreting the structure of a dataset, which indicates that $u_{ci}$ is a relative measure of how $x_i$ belongs to cluster $c$ compared to the degrees of belonging of this object to other clusters. When the “center-vectors” of all clusters are found, the labeling of one object is simply decided by the distances between this object and all the clusters.

Similarly, we denote

$$y_c = u^m_c = (u^m_{c1}, u^m_{c2}, ..., u^m_{cN})^T,$$

and

$$b(x_j, y_c) = \sum_{i=1}^{N} y_{ci} r_{ij} = y^T_c r_j,$$

thus, the update equation of $v_{cj}$ in (8) can be rewritten as

$$v_{cj} = \frac{b(x_j, y_c)^{-1/(n-1)}}{\sum_{h=1}^{N} b(x_h, y_c)^{-1/(n-1)}}.$$  \hspace{1cm} (14)

It is straightforward to see that $y_c$ in (12) is the soft assignment vector in cluster $c$, which records the degrees of belonging of all objects in cluster $c$, and $b(x_j, y_c)$ is the sum of dissimilarities of $x_j$ to other objects in cluster $c$ weighted by their
memberships in that cluster. Thus, the value of $b(x_j, y_c)$ reflects how central or typical an object $x_j$ is in cluster $c$. The final representative weight $v_{cj}$ in (14) is just a normalized $b(x_j, y_c)$ over all objects in the same cluster. From Eq. (14), it is shown that $v_{cj}$ is independent on other clusters. Therefore, $V$ is a measurement of intra-cluster representativeness.

To sum up, we can see that in a cluster, each object $x_j$ is weighted or ranked by comparing the centrality of this object to that of other objects in the same cluster, and the representative weights of all objects with respect to a cluster $c$ are used to represent or describe that cluster. Given the representation of each cluster, each of the objects is then assigned memberships to the $K$ clusters based on the relative distances between this object to each of the clusters. In this way, the update rules derived in (7) and (8) make $U$ a cluster assignment measure for partitioning of a dataset, and $V$ a cluster-based representative measure for selection of representative objects in each cluster by weighting them in a reasonable way.

3.2. Robustness

Clustering algorithms which are sensitive to the presence of noise or outliers are unable to perform well in real applications where noise is unavoidable. An outlier or noisy object should not be classified into any of the clusters, yet technically partitioning implies that each object is placed in some cluster. Generally, it can be understood that an outlier is very dissimilar from all objects and is relatively distant from all clusters. It is well known that fuzzy-based clustering approaches are not robust to noise. A possible solution to this problem is to relax the summation constraint on the fuzzy membership to give the so-called possibilistic membership in [20]. Recent studies in [21,22] show that combinations of fuzzy and possibilistic approaches may achieve better results. In our approach, we would like to investigate whether the prototype weight could be used to reduce the overall impact of noise. Since it is defined to describe the relative degree of representativeness of a cluster by an object, an outlier is always associated with some small values of prototype weights in all clusters, although it may have large fuzzy membership values in some clusters. This indicates that prototype weight is able to capture noise or is less affected by noise than fuzzy membership. As fuzzy membership $U$ in FMMdd is updated based on prototype weight $V$ in each iteration, the robustness of $V$ in turn helps to reduce or alleviate the negative impact of noise on $U$. Thus, without any modification, FMMdd is less sensitive to noise than other membership-alone clustering approaches. We will illustrate this with numerical examples later in Section 4.3.

3.3. Relation to FCMdd and A-P

Now we are going to show that FCMdd and A-P can be regarded as two special cases of FMMdd with $n \rightarrow 1^+$ and $m = n = 2$, respectively.

From (8), for $c \in \{1, \ldots, K\}$,

$$\lim_{n \rightarrow 1^+} v_{cj} = \begin{cases} 1 & \text{for } j = q, \\ 0 & \text{otherwise,} \end{cases} \quad (15)$$

where

$$q = \arg \min_{1 \leq i \leq N} \sum_{i=1}^{N} u_{ci}^{m} r_{it}. \quad (16)$$

This means that only one object $x_q$ is selected to represent cluster $c$. If we use $\delta_c$ to denote the representative object of cluster $c$, i.e., $\delta_c = x_q$, then (7) becomes

$$u_{ci} = \frac{((\text{Dis}(x_i, \delta_c))^{-1/(m-1)} \sum_{j=1}^{K} \text{Dis}(x_i, \delta_j))^{-1/(m-1)}}, \quad (17)$$

where $\text{Dis}(x_i, \delta_c)$ denotes the dissimilarity between $x_i$ and $\delta_c$, and $\text{Dis}(x_i, \delta_c) = r_{iq}$ when $x_q$ is the representative object of cluster $c$. Eqs. (16) and (17) are identical to update rules of medoid and fuzzy membership in FCMdd [8], respectively. This means that FCMdd is an extreme case of FMMdd with parameter $n \rightarrow 1^+$. 
Moreover, \( \forall c, i \),
\[
\lim_{m \to \infty} u_{ci} = \frac{1}{K}, \quad \lim_{m \to \infty} v_{cj} = \frac{\left( \sum_{i=1}^{N} r_{ij} \right)^{-1/(n-1)}}{\sum_{q=1}^{N} \left( \sum_{i=1}^{N} r_{iq} \right)^{-1/(n-1)}}
\]
and
\[
\lim_{n \to \infty} u_{ci} = \frac{1}{K}, \quad \lim_{n \to \infty} v_{cj} = \frac{1}{N}.
\]

The results in (18) and (19) indicate that when \( m \) is very large, all objects will be assigned with equal memberships to all the clusters, and all clusters have the same set of representative objects. When \( n \) is very large, the same result of \( U \) is obtained as when \( m \) is very large, but all objects are equally weighted as cluster representatives in all the clusters. Obviously, both cases fail to provide useful information for the partitioning of the data and the selection of representative objects of each cluster. The A-P algorithm [7] is another special case of FMMdd with a fixed \( m = n = 2 \) in (1) and accordingly in (7) and (8). Experimental results reported in [19] show that A-P failed to provide any useful results on some datasets. According to the analysis given above, it is clear now that the failure was caused by the wrong setting of \( m \) and \( n \). The values of \( m \) and \( n \) should be chosen within a proper range in order to avoid the membership being either too crisp or too fuzzy. Since the amount of fuzziness needed in memberships varies from dataset to dataset, it becomes less possible to make A-P work well on various datasets without any mechanism to fine-tune the fuzziness. In fact, the optimal values of \( m \) and \( n \) do not have to be the same. According to our experimental experience, setting \( m = n = 2 \) always makes the final memberships and prototype weights too fuzzy to be useful as in A-P.

4. Experimental results

In this section, we give some numerical studies on FMMdd and several existing fuzzy relational clustering approaches including FCMdd [8], A-P [7], NERFCM [4] and FRC [6]. The comparisons are made in terms of effectiveness as well as robustness. The results of PFC [11] are not included here since both FMMdd and PFC are in the family of fuzzy clustering with weighted medoids and a similar performance can be expected for them. More experimental results of PFC including those on real-world datasets can be found in [11].

4.1. Weighted medoids vs. single medoid

First, we compare the results of FMMdd with FCMdd to show that multiple-weighted objects are able to represent the clusters better than a single object. The test dataset Data1 is originally used in [11]. The distribution of Data1 is shown in Fig. 1, and the corresponding coordinates are included in Table 1. It can be seen from the figure that object \( x_1 \) to \( x_4 \) form one cluster \( C_1 \), and \( x_5 \) to \( x_{10} \) form the other cluster \( C_2 \). Object \( x_{11} \) is located near the middle of two clusters but slightly biased towards \( C_2 \). The dissimilarity matrix is calculated using Euclidean distance and is scaled as \( r_{ij} = d_{ij} / \max_{i,j} \{ d_{ij} \} \). We set \( K = 2, \, e = 1.0 \times 10^{-5} \) for the two approaches, and \( m = 1.9 \) for FCMdd. Two groups of different parameter settings \( m = 1.9, n = 1.2 \) and \( m = 1.9, n = 1.5 \) are used for FMMdd. The results of FCMdd and FMMdd are given in Table 1, where the medoids found by FCMdd are labeled with “*”.

![Fig. 1. The scatter plot of Data1. The circle “◦” represents the ideal center of each cluster. Object \( x_{11} \) (0.05, 1) is slightly biased to \( C_2 \). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)](image-url)
Table 1 shows that, generally, the partitions produced by the two methods are similar, which indicates that $x_1$ to $x_4$ are in $C_1$ where $x_4$ is the most representative object, and object $x_5$ to $x_{10}$ are in $C_2$ where $x_7$ is the most representative object. However, the two approaches give inconsistent memberships for $x_{11}$ to the two clusters. In FCMdd, $x_{11}$ is assigned a larger membership in $C_1$ than in $C_2$ while the situation is opposite in FMMdd. Since $x_{11}$ is closer to the ideal center of $C_2$, the result given by FMMdd is more reasonable. The misleading result on the cluster assignment of $x_{11}$ produced by FCMdd is due to its single-medoid representation. Because the distance from $x_{11}$ to the medoid of $C_1$, which is $x_4$, is smaller than that to the medoid of $C_2$ which is $x_7$, the membership of $x_{11}$ in $C_1$ turns out to be larger than that in the other cluster. For FMMdd, it shows that when the value of $n$ is increased from 1.2 to 1.5, more objects are engaged in representing each cluster and more obviously that $x_{11}$ is biased to $C_2$. This example illustrates the limitation of the single-object cluster representation, especially in cases where for some cluster there is no candidate object locating near the ideal center of the cluster. At the same time, it is confirmed that the weighted-multiple-objects cluster representation in FMMdd is more powerful for capturing more complicated cluster structures.

4.2. Comparison: FMMdd, FRC, NERFCM, and A-P

Fig. 2 depicts the distribution of Data2 with its coordinates included in Table 3. This dataset has been used in [19] for numerical comparison between RFCM [3] and A-P [7]. Six different distance metrics listed in Table 2 including squared Euclidean $R_E$, squared Mahalanobis $R_M$ and four distorted ones namely $R_E + 10$, $R_E + 20$, $R_E + 30$, and $R_E + 40$ are used to generate six different dissimilarity matrices for Data2. The corresponding relational data generated tends to be less separated when the dissimilarity metric used is ranged from $R_E$ to $R_E + 40$ as listed in Table 2. It has been observed in [19] that A-P fails to produce useful information based on dissimilarity matrices generated by those metrics except the one computed with $R_E$. Here we use the same data to retest and compare FMMdd with A-P [7], NERFCM [4] and FRC [6]. We follow the same settings and initialization as in [19]. The resulting fuzzy memberships
of four approaches are given in Table 3, where FRC and NERFCM produce the same results. Due to the symmetric property of Data2, only fuzzy memberships in one cluster are listed.

Table 3 shows that the results of NERFCM and FRC we obtained are identical to the results of RFCM reported in [19], which means that all the dissimilarity data are “Euclidean type”. Like these three approaches, FMMdd produces reasonable results on dissimilarity data generated by all the six metrics. According to the memberships produced by NERFCM or FMMdd, \( x_1 \) to \( x_5 \) are in the same cluster, \( x_1 \) to \( x_{11} \) are in the other cluster, and object \( x_6 \) is located in the middle of two clusters. However, A-P can only give similar results to those of other approaches when the relational data are calculated by \( R_E \). Using other dissimilarity metrics, A-P fails to detect any cluster structures since all values in \( U \) are nearly the same. The same results of A-P were reported in [19], but the reason for this problem was not further investigated in that study.

---

Table 2

Dissimilarity measures used for Data2.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>((i,j)) the off-diagonal entry</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( R_E )</td>
<td>((x_i - x_j)^2 )</td>
<td>Squared Euclidean</td>
</tr>
<tr>
<td>( R_M )</td>
<td>((x_i - x_j)^2 M^{-1}(x_i - x_j))</td>
<td>Squared Mahalanobis</td>
</tr>
<tr>
<td>( R_E + 10 )</td>
<td>((x_i - x_j)^2 (x_i - x_j) + 10)</td>
<td>Distorted ( R_E )</td>
</tr>
<tr>
<td>( R_E + 20 )</td>
<td>((x_i - x_j)^2 (x_i - x_j) + 20)</td>
<td>Distorted ( R_E )</td>
</tr>
<tr>
<td>( R_E + 30 )</td>
<td>((x_i - x_j)^2 (x_i - x_j) + 30)</td>
<td>Distorted ( R_E )</td>
</tr>
<tr>
<td>( R_E + 40 )</td>
<td>((x_i - x_j)^2 (x_i - x_j) + 40)</td>
<td>Distorted ( R_E )</td>
</tr>
</tbody>
</table>

---

Table 3

Resulting \( u_1 \) with NERFCM/FRC, A-P and FMMdd on Data2.

<table>
<thead>
<tr>
<th>( i ) (ID)</th>
<th>( u_1 ) (ID)</th>
<th>( u_1 ) (ID)</th>
<th>( u_1 ) (ID)</th>
<th>( u_1 ) (ID)</th>
<th>( u_1 ) (ID)</th>
<th>( u_1 ) (ID)</th>
<th>( u_1 ) (ID)</th>
<th>( u_1 ) (ID)</th>
<th>( u_1 ) (ID)</th>
<th>( u_1 ) (ID)</th>
<th>( u_1 ) (ID)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coordinates</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( y )</td>
<td>–5</td>
<td>–3</td>
<td>–3</td>
<td>–3</td>
<td>–1</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>( z )</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>–2</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>–2</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>NERFCM/FRC (m=2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( u_1 ) (R_E)</td>
<td>0.9311</td>
<td>0.9051</td>
<td>0.9994</td>
<td>0.9051</td>
<td>0.8114</td>
<td>0.4999</td>
<td>0.1885</td>
<td>0.0949</td>
<td>0.0006</td>
<td>0.0949</td>
<td>0.0689</td>
</tr>
<tr>
<td>( u_1 ) (R_M)</td>
<td>0.8998</td>
<td>0.6795</td>
<td>0.9917</td>
<td>0.6795</td>
<td>0.8450</td>
<td>0.4999</td>
<td>0.1549</td>
<td>0.3205</td>
<td>0.0083</td>
<td>0.3205</td>
<td>0.1002</td>
</tr>
<tr>
<td>A-P</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( u_1 ) (R_E)</td>
<td>0.8539</td>
<td>0.8264</td>
<td>0.8976</td>
<td>0.8264</td>
<td>0.7350</td>
<td>0.5000</td>
<td>0.2650</td>
<td>0.1736</td>
<td>0.1024</td>
<td>0.1736</td>
<td>0.1461</td>
</tr>
<tr>
<td>( u_1 ) (R_M)</td>
<td>0.5002</td>
<td>0.5001</td>
<td>0.5003</td>
<td>0.5001</td>
<td>0.5002</td>
<td>0.5000</td>
<td>0.4998</td>
<td>0.4999</td>
<td>0.4997</td>
<td>0.4999</td>
<td>0.4998</td>
</tr>
<tr>
<td>FMMdd (m=1.3, n=1.5)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( u_1 ) (R_E)</td>
<td>0.9998</td>
<td>0.9992</td>
<td>1.000</td>
<td>0.9992</td>
<td>0.9861</td>
<td>0.4706</td>
<td>0.0111</td>
<td>0.0008</td>
<td>0.0000</td>
<td>0.0008</td>
<td>0.0002</td>
</tr>
<tr>
<td>( u_1 ) (R_M)</td>
<td>0.9980</td>
<td>0.9071</td>
<td>0.9997</td>
<td>0.9071</td>
<td>0.9765</td>
<td>0.4670</td>
<td>0.0184</td>
<td>0.0886</td>
<td>0.0003</td>
<td>0.0886</td>
<td>0.0021</td>
</tr>
</tbody>
</table>
Taking a close look at Table 3, it can be seen that even the value of the fuzzifier(s) is fixed, the memberships produced by NERFCM/FRC or FMMdd tend to become fuzzier when the distortion on the data increases. This indicates that other than the fuzzifier which decides the fuzziness in the resulting memberships, the data also have impact on the level of fuzziness in the memberships, i.e., the less well separated the input data is, the fuzzier the memberships turn out to be. The results of A-P with five non-standard Euclidean metrics in Table 3 are too fuzzy to be useful. For a given dataset, a direct way of reducing the fuzziness in memberships is to decrease the value of the fuzzifier. However, there is no fuzzifier that can be adjusted in A-P. This means that the fuzziness of \( U \) generated by A-P may just be proper for some datasets that are well separated, but may be too much for other datasets, i.e., those with overlaps. In other words, for a given dataset, it is unknown whether A-P can produce useful results due to such a defect in the design. The fact that FMMdd performs as well as NERFCM on all these dissimilarity data demonstrates that with a proper parameter setting, FMMdd is able to produce useful fuzzy clusters on a variety of datasets.

4.3. Robustness: FMMdd vs. NERFCM/FRC

As NERFCM/FRC and FMMdd produce similar clustering results on Data2 without outliers, we now check their performance in the presence of one outlier. Data3 shown in Fig. 3 is used for this purpose. This dataset is an extended version of Data2 with one additional object \( x_{12} \) located at \((0,10)\). The same as object \( x_6 \), object \( x_{12} \) is equally close to both clusters, but it is far way from both clusters. Table 4 shows the memberships for objects in Data3 produced by NERFCM/FRC and FMMdd with Euclidean distance \( R_E \) based dissimilarity data.

From Table 4, it can be seen that due to the fuzzy membership constraints, \( x_6 \) and \( x_{12} \) are assigned the same value of memberships in both NERFCM and FMMdd, i.e., \( u_{1,6} = u_{2,6} = u_{1,12} = u_{2,12} \), although it is natural to expect that \( x_6 \) should have a larger degree of belonging to each of the two clusters than \( x_{12} \) since \( x_6 \) is much closer to both clusters than \( x_{12} \). Nevertheless, from the last two columns of Table 4, it shows that the values of prototype weights assigned to \( x_{12} \) in two clusters are significantly smaller than those of \( x_6 \), i.e., \( v_{1,12} = v_{2,12} \prec v_{1,6} = v_{2,6} \). This is reasonable given the actual distribution of the data. In other words, this confirms that prototype weight is less sensitive to outliers than fuzzy membership in this particular case.

Now we take a look at the impact on fuzzy memberships of other objects due to the presence of the outlier. The results on Data2 in Table 3 show that when Euclidean distance \( (R_E) \) is used, for both NERFCM/FRC and FMMdd, the \( u_1 \) values of the first five objects are ranked as \( u_{1,3} > u_{1,1} > u_{1,2} = u_{1,4} > u_{1,5} \). When the outlier object \( x_{12} \) is added in, it is shown from Table 4 that for NERFCM/FRC \( u_{1,3} > u_{1,2} > u_{1,1} > u_{1,4} > u_{1,5} \) and for FMMdd \( u_{1,3} > u_{1,1} > u_{1,2} > u_{1,4} > u_{1,5} \). Comparing the ranking of memberships in this cluster, it is observed that the overall fuzzy memberships in FMMdd is less affected by noise than that in the case of NERFCM/FRC. Similar results can be derived for the other cluster. This indicates that the robustness of representative weights in FMMdd in turn decreases the overall effect of outliers on FMMdd.
Table 4
Results of NERFCM/FRC and FMMdd on Data3.

<table>
<thead>
<tr>
<th>i</th>
<th>NERFCM/FRC (m=2)</th>
<th>FMMdd (m=1.5, n=2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.9111 0.0889</td>
<td>0.9219 0.0781 0.1014 0.0285</td>
</tr>
<tr>
<td>2</td>
<td>0.9430 0.0570</td>
<td>0.9129 0.0871 0.1410 0.0451</td>
</tr>
<tr>
<td>3</td>
<td>0.9858 0.0142</td>
<td>0.9470 0.0530 0.1545 0.0463</td>
</tr>
<tr>
<td>4</td>
<td>0.8499 0.1501</td>
<td>0.8936 0.1064 0.0993 0.0397</td>
</tr>
<tr>
<td>5</td>
<td>0.8142 0.1858</td>
<td>0.8223 0.1777 0.1371 0.0811</td>
</tr>
<tr>
<td>6</td>
<td>0.5 0.5</td>
<td>0.5 0.5 0.1077 0.1077</td>
</tr>
<tr>
<td>7</td>
<td>0.1857 0.8143</td>
<td>0.1777 0.8223 0.0811 0.1370</td>
</tr>
<tr>
<td>8</td>
<td>0.0570 0.9430</td>
<td>0.0871 0.9129 0.0451 0.1410</td>
</tr>
<tr>
<td>9</td>
<td>0.0142 0.9858</td>
<td>0.0530 0.9470 0.0463 0.1545</td>
</tr>
<tr>
<td>10</td>
<td>0.1501 0.8499</td>
<td>0.1064 0.8936 0.0397 0.0993</td>
</tr>
<tr>
<td>11</td>
<td>0.0889 0.9111</td>
<td>0.0781 0.9219 0.0285 0.1014</td>
</tr>
<tr>
<td>12*</td>
<td>0.5 0.5</td>
<td>0.5 0.5 0.0184 0.0184</td>
</tr>
</tbody>
</table>

Fig. 4. Profiles of the seed objects for generating GENE.

4.4. Synthetic data

In this experiment, we test the performance of FMMdd in a noisy environment on a 15-dimensional artificial dataset called GENE, which is first discussed in [23]. This is a correlation-inclined dataset with around 20% noise. The dataset consists of three well separated clusters C1, C2 and C3 which contains 900, 700 and 500 objects, respectively. Following the steps given in [23], we first generate three well separated objects as seeds. The correlation coefficients between each of the seed objects and the other two are less than 0.1. After that, a required number of normal objects in each cluster are randomly generated. An object is kept only if the correlation coefficient between this object and the seed object of that cluster is larger than 0.8. Finally, 400 randomly generated objects are added into the dataset as outliers. The profiles of three seed objects are plotted in Fig. 4. As suggested in [23], Pearson’s correlation coefficient is used as the similarity measure and negative values are ignored. If \( S \) denotes Pearson’s similarity matrix, then the dissimilarity matrix is calculated as \( R = 1 - S \), where \( 1 \) is a matrix of all 1s.

Table 5 displays the clustering results of this dataset. In FMMdd, we label outliers based on \( V \). Since outliers always rank low in all of the clusters, we check the largest prototype weight of each object with respect to all the clusters and label it as an outlier if this value is smaller than a predefined threshold \( T \), i.e., if \( \forall c, v_{cj} < T, x_j \) is regarded to be an outlier. In this experiment, we set \( T=1/N \). To make comparison, we include the results of similarity-based PCM (SPCM)
Table 5
Contingency table for GENE data.

<table>
<thead>
<tr>
<th></th>
<th>FMMdd</th>
<th></th>
<th></th>
<th></th>
<th>SPCM [23]</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>C1</td>
<td>C2</td>
<td>C3</td>
<td>Noise</td>
<td>C1</td>
<td>C2</td>
<td>C3</td>
<td>Noise</td>
</tr>
<tr>
<td>Original clusters</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C1</td>
<td>900</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>896</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>C2</td>
<td>0</td>
<td>700</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>700</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>C3</td>
<td>0</td>
<td>0</td>
<td>500</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>500</td>
<td>0</td>
</tr>
<tr>
<td>Noise</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>398</td>
<td>0</td>
<td>5</td>
<td>9</td>
<td>386</td>
</tr>
</tbody>
</table>

Table 6
Summary of UCI datasets and parameter settings.

<table>
<thead>
<tr>
<th>Data summary</th>
<th>Parameter setting</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td># of objects</td>
<td># of features</td>
</tr>
<tr>
<td>wine</td>
<td>178</td>
</tr>
<tr>
<td>segment</td>
<td>2310</td>
</tr>
<tr>
<td>wdbc</td>
<td>569</td>
</tr>
<tr>
<td>digits</td>
<td>1797</td>
</tr>
<tr>
<td>glass</td>
<td>214</td>
</tr>
</tbody>
</table>

reported in [23] in Table 5. From this table, it is shown that FMMdd groups all normal objects into the right clusters and successfully filters out most of the outliers with only two of them being grouped into C2 and C3, respectively; while for SPCM, four normal objects are mis-grouped into the noise cluster and 14 outliers have not been identified. This example shows that FMMdd is able to produce good results on noisy non-spatial relational data.

4.5. FMMdd vs FCMdd and A-P on UCI data

Finally, we compare the performance of FMMdd with two closely related approaches FCMdd and A-P on five real-world UCI datasets[^1] summarized in Table 6. We use two external evaluation metrics Purity [24] and F-Measure [25] to measure the quality of the clustering results produced by each algorithm. The larger the values of Purity and F-Measure are, the better the clustering result is. The results of three approaches are plotted in Fig. 5 with parameter settings in Table 6 being used for FCMdd and FMMdd.

From Fig. 5, it can be seen that FMMdd outperforms both FCMdd and A-P on all the five datasets. A-P only gives comparable Purity score on wdbc and comparable F-Measure scores on wdbc and glass, but its performance in other cases degrades to a very low level. Benefitted from the multiple-weighted medoids scheme of cluster representation, FMMdd makes improvements in the final clustering results over those produced by FCMdd. The improvement is especially significant for the wine and digits datasets. This is possibly because compared with the other three datasets namely segment, wdbc and glass, the single-object cluster representation in FCMdd is much less sufficient for these two datasets. Thus, more benefit can be made of the multiple-weighted medoids representation in FMMdd.

5. Conclusions

In this paper, we study the medoid-based fuzzy relational clustering in a unified view based on the new formulation FMMdd. We have shown that FMMdd is a general form of FCMdd and A-P, but providing better results. We also provide discussions on the robustness of FMMdd to show that FMMdd is less sensitive to noise than partition-alone fuzzy relational clustering approaches. These properties of FMMdd are illustrated and confirmed by our experimental results.

results. In the future, more efforts may be made to solve some important but challenging issues, such as the selection of parameter $m$ and $n$, and the estimation of the number of clusters $K$.

**References**


