Pattern Recognition 47 (2014) 3920–3930

Contents lists available at ScienceDirect

Pattern Recognition

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

Improvements to the relational fuzzy *c*-means clustering algorithm

Mohammed A. Khalilia [a,*], James Bezdek [b], Mihail Popescu [c], James M. Keller [b]

[a] Computer Science Department, University of Missouri, Columbia, MO 65211, USA
[b] Electrical and Computer Engineering Department, University of Missouri, Columbia, MO 65211, USA
[c] Health Management and Informatics Department, University of Missouri, Columbia, MO 65212, USA

ARTICLE INFO

Article history:
Received 30 December 2013
Received in revised form
20 June 2014
Accepted 23 June 2014
Available online 8 July 2014

Keywords:
Fuzzy clustering
Relational *c*-means
Euclidean distance matrices

ABSTRACT

Relational fuzzy *c*-means (RFCM) is an algorithm for clustering objects represented in a pairwise dissimilarity values in a dissimilarity data matrix D. RFCM is dual to the fuzzy *c*-means (FCM) object data algorithm when D is a Euclidean matrix. When D is not Euclidean, RFCM can fail to execute if it encounters negative relational distances. To overcome this problem we can Euclideanize the relation D prior to clustering. There are different ways to Euclideanize D such as the β-spread transformation. In this article we compare five methods for Euclideanizing D to $\tilde{D}$. The quality of $\tilde{D}$ for our purpose is judged by the ability of RFCM to discover the apparent cluster structure of the objects underlying the data matrix D. The subdominant ultrametric transformation is a clear winner, producing much better partitions of $\tilde{D}$ than the other four methods. This leads to a new algorithm which we call the improved RFCM (iRFCM).

1. Introduction

Consider a set of objects $O=\{o_1,\cdots,o_n\}$, where the goal is to group them into c natural groups. Objects can be described by feature vectors $X=\{x_1,\ldots,x_n\}\in\mathbb{R}^p$ such that x_i is an attribute vector of dimension p representing object o_i. Alternatively, objects can be represented using a pairwise relationship. The relationships are stored in a relational matrix R, where $R=[r_{ij}]$ measures the relationship between o_i and o_j. If R is a dissimilarity relation denoted by $D=[d_{ij}]$, then it must satisfy the following three conditions:

$$d_{ii}=0 \quad \text{for } i=1,\cdots,n; \tag{1a}$$

$$d_{ij}\geq 0 \quad \text{for } i=1,\cdots,n \text{ and } j=1,\cdots,n; \text{ and} \tag{1b}$$

$$d_{ij}=d_{ji} \quad \text{for } i=1,\cdots,n \text{ and } j=1,\cdots,n, \tag{1c}$$

where condition (1a) is self-dissimilarity, (1b) is non-negativity and (1c) is symmetry. A well-known relational clustering algorithm that is suitable for clustering objects described by D is the relational fuzzy *c*-means (RFCM) proposed in [1] (Algorithm 1). RFCM, the relational dual of the FCM algorithm, takes an input dissimilarity matrix D and outputs a fuzzy partition matrix $U\in M_{fcn}$, where

$$M_{fcn}=\left\{U\in\mathbb{R}^{c\times n}|u_{ik}\in[0,1],\ \sum_{k=1}^{n}u_{ik}>0,\ \sum_{i=1}^{c}u_{ik}=1,\ \forall\ 1\leq i\leq c \text{ and } 1\leq k\leq n\right\} \tag{2}$$

Algorithm 1. Relational fuzzy *c*-means (RFCM) [1]

1 **Input**: D, c, fuzzifier $m>1$ (default $m=2$), t_{max} (default $t_{max}=100$), ε (default $\varepsilon=0.0001$)
2 **Output**: U, V_R
3 **Initialize**: step$=\varepsilon$, $t=1$
4 Relational cluster centers $V_R^0=(v_{R,1}^0,v_{R,2}^0,\cdots,v_{R,c}^0)$, $v_{R,i}^0\in\mathbb{R}^n$
Note: we use c randomly chosen rows of D as initial centers.
5 **while** $t\leq t_{\max}$ and step $\geq\varepsilon$
6 $d_{R,ik}=(Dv_{R,i}^{t-1})_k-\frac{1}{2}(v_{R,i}^{t-1})^TDv_{R,i}^{t-1}$ for $1\leq i\leq c$ and $1\leq k\leq n$ (3)
7 **for** $k=1$ *to* n
8 **if** $d_{R,ik}\neq 0$ for all i
9 $u_{ik}=1/\left(\frac{d_{R,ik}}{\sum_{j=1}^{c}d_{R,ik}}\right)^{1/m-1}$; $\forall i$ (4)
10 **else**
11 Set $u_{ik}>0$ for $d_{R,ik}=0$, $u_{ik}\in[0,\ 1]$ and $\sum_{j=1}^{c}u_{jk}=1$
12 **endif**

* Corresponding author.
E-mail address: mohammed.khalilia@gmail.com (M.A. Khalilia).

http://dx.doi.org/10.1016/j.patcog.2014.06.021
0031-3203/

13 **endfor**

14 $v_{R,i}^{t} = (u_{i1}^{m}, \cdots, u_{in}^{m}) / \sum_{k=1}^{n} u_{ik}^{m} \quad \text{for } 1 \le i \le c$ (5)

15 $\text{step} \leftarrow \max_{\substack{1 \le i \le c \\ 1 \le j \le n}} \{|V_R^{(t)} - V_R^{(t-1)}|\}$

16 $t \leftarrow t+1$

17 **endwhile**

The duality relationship between RFCM and FCM is based on the squared Euclidean distance or 2-norm that defines the dissimilarity d_{ij} between two feature vectors x_i and x_j describing o_i and o_j and the dissimilarity between the cluster center v_i and o_j. In other words, RFCM assumes that

$$D = [d_{ij}] = [||x_i - x_j||_2^2] \tag{6}$$

The relation $D = [d_{ij}]$ is Euclidean if there exists feature vectors $X = \{x_1, \ldots, x_n\} \in \mathbb{R}^p$ with an embedding dimension $p < n$, such that for all i,j $d_{ij} = ||x_i - x_j||_2^2$. When D is Euclidean, it has a realization in some Euclidean space. In this case, RFCM and FCM will produce the same partition of relational and feature vector representation of the data. If D is not Euclidean, RFCM will still find clusters in any D whose entries satisfy (1) as long as it can execute, but in this case it is possible for RFCM to experience an execution failure. This happens when the relational distances between prototypes and objects $d_{R,ik}$ in Eq. (3) become negative for some i and k (Algorithm 1, line 6). Another important observation about RFCM is that it expects *squared* dissimilarities D. If the dissimilarities are not squared, meaning that we have $\sqrt{D}$ instead of D such that $\sqrt{D} = D^{1/2} = [\sqrt{d_{ij}}]$, then the dissimilarities must be squared before clustering using RFCM so that D is the Hadamard product $D = (\sqrt{D})^2$. Throughout this paper D is assumed to contain *squared* dissimilarities.

Non-Euclidean Relational Fuzzy c-Means (NERFCM), repairs RFCM "on the fly" with a self-healing property that automatically adjusts the values of $d_{R,ik}$ and the dissimilarities in D in case of failure [2]. The self-healing property is based on the β-spread, which works by adding a positive constant β to the off-diagonal elements of D. In fact, there exists β_0 such that the β-spread transformed matrix D_β is Euclidean for all $\beta \ge \beta_0$. The parameter β controls the amount spreading and must be as small as possible to minimize unnecessary dilation that distorts the original D, which in turn may result in the loss of cluster information. The exact value of β_0 is the largest positive eigenvalue of the matrix PDP, where $P = I - (1/n)(11^T)$ and I is $n \times n$ identity matrix. Eigenvalue computation is avoided by the self-healing module, which is invoked during execution only when needed. When activated, this module adjusts the current D by adding a minimal β-spread to its all off-diagonal elements.

An alternative to using NERFCM is to transform the matrix D by a mapping that converts it to Euclidean form (we call this operation "Euclideanizing D"), and then running RFCM on the Euclideanized matrix $\tilde{D}$. This approach guarantees that RFCM will not fail since $\tilde{D}$ is already Euclidean. There are at least five ways to Euclideanize D, including the β-spread transformation. In addition to the β-spread transformation, this paper will study the other four Euclideanization approaches indicated under option 1 in Fig. 1. As a result of this study, we will append an "i" (short for the word "improved") to RFCM, but not to NERFCM, which is NOT altered by these results. We hope to write a companion paper to this one that discusses improvements to NERFCM which would then become iNERFCM, but attempts to find an alternative to the current "self-healing" method described in [2] which is NERFCM have so far met stiff resistance.

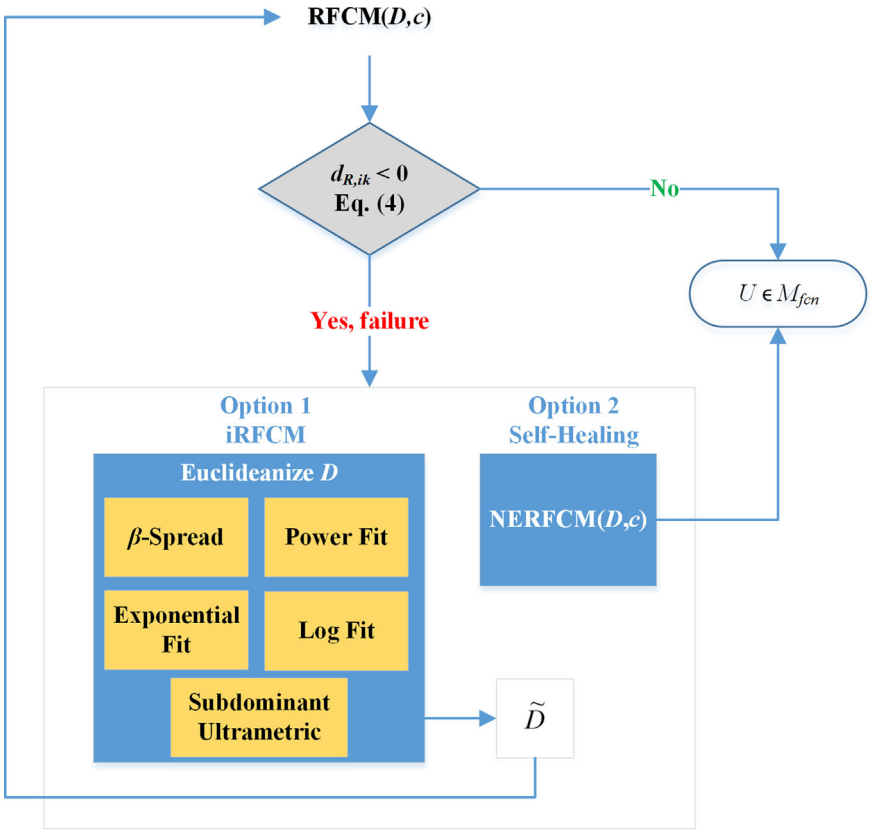

Fig. 1. Possible solutions RFCM can utilize when input D is non-Euclidean.

2. Euclidean distance matrices (EDM) and the iRFCM algorithm

Given a dissimilarity matrix D it is known that

D is a Euclidean distance matrix(EDM) $\Leftrightarrow$ $W(D_{0.5})$ is positive semi – definite ($p.s.d$) (7)

where

$$W(D_{0.5}) = PD_{0.5}P, \tag{8}$$

P is the centering matrix defined as

$$P = I - \frac{1}{n}(11^T), \tag{9}$$

I is the identity matrix and $D_{0.5}$ is defined as

$$D_{0.5} = \begin{cases} -1/2D & \text{for squared dissimilarities} \\ -1/2(\sqrt{D})^2 & \text{Otherwise} \end{cases}. \tag{10}$$

In (10) and below, $(\sqrt{D})^2$ is the Hadamard square of $\sqrt{D}$. The trick in using (8) is knowing if the dissimilarities are squared as in D or not squared as in $\sqrt{D}$, which determines which case of (10) to use. This is a question we cannot answer; rather the answer depends on one's knowledge of how the dissimilarities were computed.

The number of strictly positive eigenvalues of $W(D_{0.5})$ gives the maximum number of the embedding dimensions $p < n$ for the realization of D [3,4]. If $W(D_{0.5})$ is not $p.s.d$ then D can be Euclideanized to $\tilde{D}$ by making $W(D_{0.5})$ $p.s.d$ using the following general transformation

$$W(\tilde{D}) = W(D_{0.5}) + \gamma W(\Delta_{0.5}), \tag{11}$$

where γ is some positive constant,

$$\Delta = \{[\delta_{ij}] | \delta_{ij} = 0 \ \forall i = j, \delta_{ij} \geq 0 \ \forall i \neq j \text{ and } 1 \leq i, j \leq n\} \tag{12}$$

and $\Delta_{0.5}$ is computed the same way as $D_{0.5}$ using (10). Eq. (11) implies that the Euclideanized $\tilde{D}$ is given by

$$\tilde{D} = D + \gamma\Delta\,. \tag{13}$$

Table 1 lists the five transformations that carry non-Euclidean D's into Euclidean $\tilde{D}$'s that we will be considered in this paper.

Given that D is not Euclidean and Δ has an Euclidean representation of dimension $n-1$, the goal is to find a positive constant γ to Euclideanize D. In the β-spread case (14a) we can find the exact γ, which is $\gamma = -2\lambda$, where λ is the smallest eigenvalue of $W(D_{0.5})$ at (8). Bénasséni [5] generalized this concept by incorporating additional choices of Δ. To understand Benasseni's generalization we rewrite $W(\Delta_{0.5})$ in terms of its eigendecomposition $W(\Delta_{0.5}) = V(W(\Delta_{0.5})) \cdot \Lambda(W(\Delta_{0.5})) \cdot V(W(\Delta_{0.5}))^T$ where $\Lambda(W(\Delta_{0.5}))$ is the $(n-1) \times (n-1)$ diagonal matrix of the non-zero eigenvalues of $W(\Delta_{0.5})$ and $V(W(\Delta_{0.5}))$ is the corresponding $n \times (n-1)$ matrix of the normalized eigenvectors. Since $\Lambda(W(\Delta_{0.5}))$ is positive definite, the minimum constant γ that makes (11) $p.s.d$ is $\gamma = -\lambda(D_{0.5}, \Delta_{0.5})$ where $\lambda(D_{0.5}, \Delta_{0.5})$ is given by

$$\lambda(D_{0.5}, \Delta_{0.5}) = \lambda_{min}(\Lambda(W(\Delta_{0.5}))^{-1/2} \cdot V(W(\Delta_{0.5}))^T \cdot W(D_{0.5}) \cdot V(W(\Delta_{0.5})) \cdot \Lambda(W(\Delta_{0.5}))^{-1/2}). \tag{15}$$

For a more detailed proof the reader is referred to [5].

2.1. β-Spread

In (14a) the same constant is added to all the off-diagonal dissimilarity values in D. In some cases adding the same constant to all of the off-diagonal elements can cause a lot of distortion and a large correspondingly discrepancy between D and $\tilde{D}$, causing $\tilde{D}$ to lose the original structure of the data. This distortion can propagate into the RFCM clustering algorithm, causing a loss in the original cluster information. This is a very serious concern when $d_{ij} \in [0, 1]$ and the additive constant γ is large (an example of this will be shown in the results section). To alleviate this problem, we can use one of the other choices of Δ listed in Table 1, such as the subdominant ultrametric (SU).

2.2. Subdominant ultrametric

The SU of D, denoted as Δ^{SU}, is derived from the minimum spanning tree of D, $MST(D)$. Recall that D represents an undirected graph whose vertices are the objects described by D. A length d_{ij} is assigned to each edge (i, j). To determine Δ^{SU}, construct $MST(D)$ denoted as T, such as the one shown in Fig. 2. T may not be uniquely determined if some edges have identical weights, but Δ^{SU} is unique and does not depend on any particular choice of T [6]. We use Prim's algorithm to determine the MST [7].

Eq. (14b) states that the SU distance between i and j, δ_{ij}^{SU}, is the maximum weight along the path $P_k = (i = v_0, v_1, \cdots, v_k, v_{k+1} = j)$ connecting objects i and j. In Fig. 2 there are six edges between i and j (bolded color). The first edge (i, p) has weight d_{ip}, second edge (p, q) has weight d_{pq}, etc. The SU distance between i and j for the particular MST in Fig. 2 is then given by the edge with the highest weight

$$\delta_{ij}^{SU} = max\{d_{ip},\ d_{pq}\ d_{qr},\ d_{rw},\ d_{wy},\ d_{yj}\} = d_{ip}$$

Unlike the other transformations we will discuss later, the SU distance is a function of the original dissimilarities and its objective is to maximize the distance between any two objects. Holman in [8] proved that Δ^{SU} on n objects is Euclidean with $n-1$ dimensions. Once Δ^{SU} is computed we can find γ, where $\gamma = -\lambda(D_{0.5}, \Delta_{0.5}^{SU})$.

The following squared dissimilarity matrix

$$D = \begin{bmatrix} 0 & 9 & 36 & 81 \\ 9 & 0 & 49 & 36 \\ 36 & 49 & 0 & 4 \\ 81 & 36 & 4 & 0 \end{bmatrix}$$

is not $p.s.d$ since the eigenvalues of $W(D_{0.5}) = PD_{0.5}P$ at (8) are $\{-11.31,\ 0,\ 15.77,\ 49.28\}$.

Table 1
Transformations of $D \to \tilde{D}$.

Name	Formula		Reference	Eqn.
β-Spread	$\Delta^{\beta} = 11^T - I;\ \beta > 0$	(14a)	[2,3,5]	(14a)
Subdominant Ultrametric (SU)	$\Delta^{SU} = [\delta_{ij}^{SU}]$	(14b)	[6,8]	(14b)
	$\delta_{ij}^{SU} = \max\{d_{v_k v_{k+1}} \in P_k | P_k = (i = v_0, v_1, \cdots, v_k, v_{k+1} = j) \in \text{MST}(D)\}$			
	$MST(D)$ is the minimum spanning tree of D (Fig. 2)			
Power Fit (PF)	$\Delta^{PF} = D^{\alpha};\ 0 < \alpha \leq 1$	(14c)	[5,11]	(14c)
Exponential Fit (EF)	$\Delta^{EP} = (11^T - e^{-\alpha\sqrt{D}})^2;\ \alpha > 0$	(14d)	[5,11]	(14d)
Log Fit (LF)	$\Delta^{LF} = (\log_2(11^T + (\sqrt{D})^{\alpha}))^2;\ 0 < \alpha \leq 1$	(14e)	[11]	(14e)

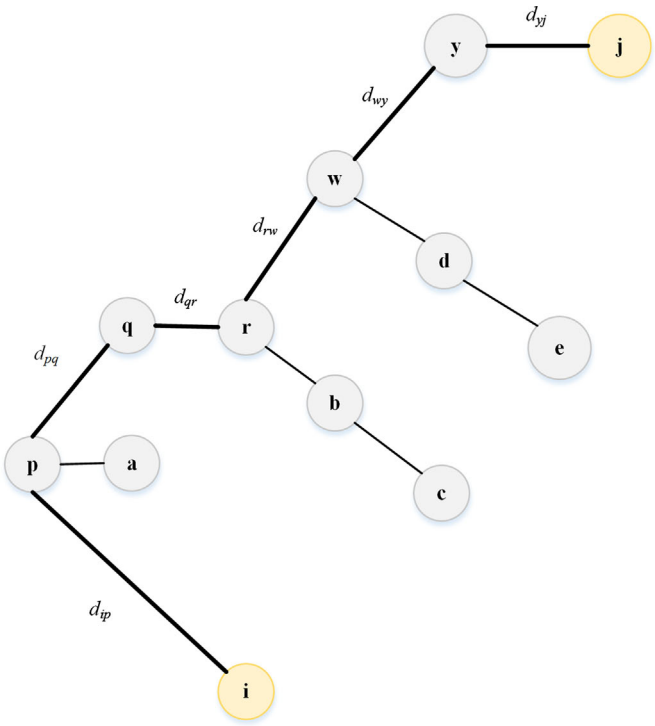

Fig. 2. Example of a minimum spanning tree.

Using the SU we can Euclideanize D, which first involves computing the MST of D that will be used to compute Δ^{SU}.

$$D=\begin{bmatrix}0&9&36&81\\9&0&49&36\\36&49&0&4\\81&36&4&0\end{bmatrix}\xrightarrow{MST(D)}\quad\begin{matrix}36\\1\\9\end{matrix}\ \begin{matrix}3\\2\end{matrix}\ \begin{matrix}4\\4\end{matrix}\quad\xrightarrow{\Delta^{SU}}\quad\begin{bmatrix}0&9&36&36\\9&0&36&36\\36&36&0&4\\36&36&4&0\end{bmatrix}$$

To compute the smallest eigenvalue $\lambda(D_{0.5},\Delta_{0.5}^{SU})$ in (15) we first compute $W(\Delta_{0.5}^{SU})=P\Delta_{0.5}^{SU}P$, where $\Delta_{0.5}^{SU}=-1/2\Delta^{SU}$.

$$W(\Delta_{0.5}^{SU})=\begin{bmatrix}3/4&-1/4&-1/4&-1/4\\-1/4&3/4&-1/4&-1/4\\-1/4&-1/4&3/4&-1/4\\-1/4&-1/4&-1/4&3/4\end{bmatrix}\times\begin{bmatrix}0&-4.5&-18&-18\\-4.5&0&-18&-18\\-18&-18&0&-2\\-18&-18&-2&0\end{bmatrix}\times\begin{bmatrix}3/4&-1/4&-1/4&-1/4\\-1/4&3/4&-1/4&-1/4\\-1/4&-1/4&3/4&-1/4\\-1/4&-1/4&-1/4&3/4\end{bmatrix}$$

$W(D_{0.5})$ is computed the same way. To save space we do not show the eigenvalues and eigenvectors of $W(\Delta_{0.5}^{SU})$, but once computed, we will have $\Lambda(W(\Delta_{0.5}))$, 3×3 diagonal matrix of non-zero eigenvalues and $V(W(\Delta_{0.5}))$, 3×4 normalized eigenvectors. Inserting $V(W(\Delta_{0.5}))$, $\Lambda(W(\Delta_{0.5}))$ and $W(D_{0.5})$ in (15) gives $\lambda(D_{0.5},\Delta_{0.5}^{SU})=-3.84$. Then with D, Δ^{SU} and γ, where $\gamma=-\lambda(D_{0.5},\Delta_{0.5}^{SU})=3.84$, in (13) results in Euclidean form of D realized by the SU transformation,

$$\tilde{D}=\begin{bmatrix}0&43.55&174.19&219.19\\43.55&0&187.19&174.19\\174.19&187.19&0&19.35\\219.19&174.19&19.35&0\end{bmatrix}.$$

We can verify that $\tilde{D}$ is Euclidean by computing the eigenvalues of $W(\tilde{D}_{0.5})$, {0, 0, 31, 173.41}, which indicates that $W(\tilde{D}_{0.5})$ is *p.s.d.*

Euclideanization using SU incurs the highest run time complexity compared to the other four methods. It requires computing the $MST(D)$, which takes $O(n*\log n)$ and finding all-pairs shortest path using Dijkstra's algorithm takes $O(n^3*\log n)$ [9]. Computing the eigenvalues and eigenvectors of $W(\Delta_{0.5})$ is performed in all five transformations and has a running time complexity of $O(n^3)$ [10].

2.3. Power fit

The third choice of Δ (14c) belongs to the family of power functions parameterized by α. Using a transformation based on the power fit (PF) involves a smaller distortion to the original dissimilarities D compared to the β-spread transformation. According to Bénasséni [5] there exists some real constant α_0 such that D^{α} is Euclidean for $\alpha\le\alpha_0$. Notice that for any $d_{ij}^{\alpha}>0$, as $\alpha\to0$, then d_{ij}^{α} tends monotonically to 1. In other words, if $d_{ij}>0$ for all i, j and $i\neq j$ the $\lim_{\alpha\to0}D^{\alpha}=\Delta^{\beta}$, where Δ^{β} is given in (14a).

2.4. Exponential fit

The exponential fit (EF) Δ^{EF} (14d) was first mentioned in Dattoro [11] to show that some nonlinear compositions of EDMs are also EDMs. Bénasséni [5] used this transformation to Euclideanize D. Similar to Δ^{PF}, Δ^{EF} is a function of α and the limit property of (14d) states that as the $\lim_{\alpha\to\infty}(11^T-e^{-\alpha\sqrt{D}})^2=\Delta^{\beta}$ if $d_{ij}>0$ for all i,j and $i\neq j$. Bénasséni [5] shows that there exists α_0 such that for $\alpha\ge\alpha_0$, Δ^{EF} is Euclidean.

2.5. Log fit

Preliminary findings by Dattoro raise the question on whether any concave non-decreasing composition of the entries in D will produce an EDM. From the empirical evidence in [11] it is suggested that for fixed $0\le\alpha\le1$ the composition $(\log_2(11^T+(\sqrt{D})^{\alpha}))^2$ is an EDM. This nonlinear composition of EDM denoted by the log fit (LF) given in (14e) can be used in the Euclideanization of D.

The last three transformations in Table 1 are parametric and hence require finding a value α that makes D Euclidean. In this paper an intense search for α was performed for these three transformations. There may be a more efficient approach for finding α, but this is beyond the scope of this paper and will be addressed in future work. What follows is the iRFCM algorithm that incorporates the transformations mentioned above.

Algorithm 2. Improved relational fuzzy c-means (iRFCM)

1 **Input**: D, c, Δ, $m>1$ (default $m=2$), t_{max} (default $t_{max}=100$), ε (default $\varepsilon=0.0001$)
2 **Output**: U,V_R
3 **Initialize**: $D_{0.5}=-1/2D$
4 $W(D_{0.5})=PD_{0.5}P$
5 **if** $W(D_{0.5})$ is not *p.s.d*
6 $\Delta_{0.5}=-1/2\Delta$, where $\Delta\in\{\Delta^{\beta},\Delta^{SU},\Delta^{PF},\Delta^{EF},\Delta^{LF}\}$
7 $\gamma=-\lambda(D_{0.5},\Delta_{0.5})$
8 $\tilde{D}=D+\gamma\Delta$
9 $D\leftarrow\tilde{D}$
10 **endif**
11 U,V_R=RFCM(D, c, m, t_{max}, ε)

We have listed Algorithms 1 and 2 for the fuzzy case ($m>1$). There are also hard and possibilistic versions of RFCM, the relational hard c-means (RHCM) [1] and the relational possibilistic c-means (RPCM) [12]. Algorithm 2 will also generalize them by replacing line 11 in Algorithm 2 with your choice of RHCM or RPCM assuming the appropriate changes are made to Algorithm 1.

3. Experimental results

We have three types of ground truth partitions U_{GT} for the examples in this section. U_{GT} per Table 2 is assumed for Example 1, adopted from another study for Example 2, and supplied as physical labels for the three Iris subsets in Example 3. Since the ground truth itself is on somewhat shaky ground, we will use just one external cluster validity index, viz., the Hubert and Arabie Adjusted Rand Index (ARI), which is ably discussed in [13]. We denote this index as $ARI(U|U_{GT})$. This ARI (there are several other ARIs not discussed here) ranges from 0 to 1, with larger values indicating better matches between U and U_{GT} in the sense described next. Real clustering problems involve unlabeled data, hence we never have U_{GT}. But $ARI(U|U_{GT})$ can give us some confidence in the ability of a clustering algorithm to recover "meaningful clusters". The ARI is computed after first obtaining the following four values: a (number of objects paired in U and U_{GT}), d (number of objects paired in neither U nor U_{GT}), b (number of objects paired in U_{GT}, but not in U) and c (number of objects paired in U, but not in U_{GT}). With these values in hand, the adjusted Rand Index [13] is formally given by

$$ARI(U|U_{GT})=\frac{(a-((a+c)(a+b)/a+b+c+d))}{(((a+c)+(a+b)/2)-((a+c)(a+b)/d))} \tag{16}$$

Example 1. Mutation data

The Mutation dataset [14] contains 20 objects each representing a cytochrome (the names of the animal organisms are listed in Fig. 4). Fitch and Margoliash [15] recount the history of the work in [9] in a delightful one page essay that is historically charming. The distance between two cytochromes is defined as the minimum number of nucleotides that must be altered in order for the gene of one of the cytochromes to code for the other [14]. Squaring the distances in $\sqrt{D_{mut}}$ given in the lower half of Table 3 in [14] leads to the conclusion that D_{mut} is not an EDM. However, when D_{mut} is submitted to RFCM, we find experimentally that it does not fail, even though the theory predicts possible execution failure. So, this

Table 3
$ARI(U|U_{GT})$ measured from clustering Mutation, GDP194 and Iris data using the five choices of Δ.

Δ	Mutation	GDP194	Iris
D	0.53	–	–
SU	0.67	0.98	0.81
β-Spread	0.41	0.41	0.33
PF	0.33	0.74	0.84
EF	0.33	0.42	0.7
LF	0.43	0.74	0.84

Table 2
Hardened 4-partitions found by RFCM(D_{mut}) and iRFCM($\tilde{D}_{mut}$).

*Cluster 1 +Cluster 2 •Cluster 3 ■Cluster 4

			$m=1.05$							$m=2$					
o_i	Organism	U_{GT}	$U_{D_{mut}}$	$U_{\Delta^{SU}}$	$U_{\Delta^{\beta}}$	$U_{\Delta^{PF}}$	$U_{\Delta^{EF}}$	$U_{\Delta^{LF}}$	$U_{D_{mut}}$	$U_{\Delta^{SU}}$	$U_{\Delta^{\beta}}$	$U_{\Delta^{PF}}$	$U_{\Delta^{EF}}$	$U_{\Delta^{LF}}$
1	Man	*	*	*	*	*	*	*	*	*	*	+	+	*
2	Monkey	*	*	*	*	*	*	*	*	*	*	+	+	*
3	Dog	*	*	*	*	*	*	*	*	*	*	*	*	*
4	Horse	*	*	*	*	*	*	*	*	*	*	*	*	*
5	Donkey	*	*	*	*	*	*	*	*	*	*	*	*	*
6	Pig	*	*	*	*	*	*	*	*	*	*	*	*	*
7	Rabbit	*	*	*	*	*	*	*	*	*	*	*	*	*
8	Kangaroo	*	*	*	*	*	*	*	*	*	*	*	*	*
9	Pigeon	*	*	*	*	*	*	*	•	*	•	•	•	•
10	Duck	*	*	*	*	*	*	*	•	*	•	•	•	•
11	Chicken	*	*	*	*	*	*	*	•	*	•	•	•	•
12	Penguin	*	*	*	*	*	*	*	•	*	•	•	•	•
13	Turtle	*	*	*	*	*	*	*	•	*	•	•	•	•
14	Tuna	*	*	*	*	*	*	*	•	•	•	+	+	•
15	Snake	*	*	*	*	*	*	*	•	•	•	+	+	*
16	Fly	*	•	*	*	•	•	•	*	•	+	+	+	+
17	Moth	*	•	*	*	•	•	•	•	•	+	+	+	+
18	Mold	+	+	+	+	+	+	+	+	+	■	■	■	■
19	Yeast	•	■	•	•	■	■	■	■	■	■	■	■	■
20	Fungus	■	■	■	■	■	■	■	■	■	■	■	■	■

dataset is ideal for studying how much distortion each of the five transformations in Table 1 will introduce into the clusters detected by RFCM on D_{mut}. Fitch et. al. [14] visualize the data using the phylogenetic tree shown in Fig. 3.

We can also visualize the structure of the data using the Improved Visual Assessment of Tendency (iVAT) algorithm [16] as in Fig. 4. The four darkest diagonal sub-blocks in the image of Fig. 4 correspond to the three singletons Mold, Yeast and Fungus and a larger block containing the other 17 objects. Thus, the four clusters most strongly suggested by Fig. 4 are {1−17}, {18}, {19}, {20}. This agrees exactly with the clusters that would be obtained by cutting the tree in Fig. 3 at $c=4$. We will (arbitrarily) call this partition the ground truth U_{GT} 4-partition of D_{mut}.

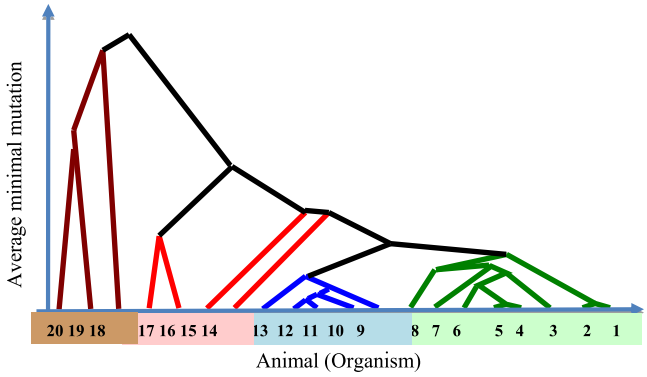

Fig. 3. Mutation phylogeny tree (Fig. 2 in [14]).

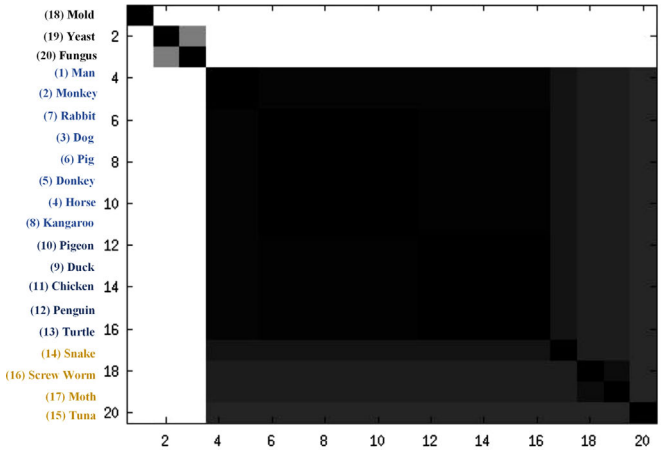

Fig. 4. iVAT image of the *squared* Mutation data D_{mut}.

It is important to point out that clusters in these objects vary by choosing different properties of the organisms. For example, we can ask: what organisms cluster together based on the number of legs, etc. So, using many different criteria results in many "correct" solutions, but the goal of this paper is not to discover the number of clusters in the Mutation data or the other datasets that we will discuss later, rather it is about the effect of Euclideanizing D_{mut} on the clustering.

Since RFCM and iRFCM produce fuzzy partitions, we need a way to convert them to crisp partitions in order to compare them with our U_{GT} partition at $c=4$. Here we use the standard *hardening* scheme, i.e., the maximum membership in each column of fuzzy partition U is replaced by 1, and the remaining $c-1$ values become 0's. RFCM was applied to D_{mut} for the fuzzifier values $m=1.05$ and $m=2$. At $m=1.05$, we are (almost) seeing the results of running RHCM on this data.

Table 2 lists the U_{GT}, the (arbitrarily chosen) ground truth partition of $\sqrt{D_{mut}}$, and also the hardened 4-partitions of D_{mut} at these two values ($m=1.05$ and $m=2$). The five transformation methods are identified by the inducing matrix $D_{mut,0.5}$, which is used in Eq. (10) to realize the EDM built with D. There are 3 mismatched labels between U_{GT} and $U_{D_{mut}}$ at $m=1.05$, and 9 label differences at $m=2$. As expected, this confirms that at $m=2$, memberships of the 20 organisms are much more widely distributed across the 4 clusters than at $m=1.05$.

Both RFCM runs group {1−8} together, and a quick look at the tree in Fig. 3 confirms this as a primary structure in the data. Visual acuity makes it hard to see this in Fig. 4, but the pixels corresponding to these 8 organisms are identified and grouped together along the vertical axis in Fig. 4 too. So, this inference is consistent with both visual representations of D_{mut}, and with our intuition about what clusters "should be" in the data, based on our everyday notions about classes of animals.

How much are RFCM partitions of D_{mut} distorted when iRFCM is applied to $\tilde{D}_{mut}$? We can get a somewhat surprising picture of what this type of feature extraction does by comparing the 5 Euclideanized data results to the results for D_{mut}. For example, compare the columns for D_{mut} and $\tilde{D}_{mut}$ using (13) with Δ^{SU} and Δ^{β}. At $m=1.05$, there are 3 disagreements between the hardened labels of $U_{D_{mut}}$ and the 4-partitions $U_{\Delta^{SU}}$ and $U_{\Delta^{\beta}}$-BUT-$U_{\Delta^{SU}}$ and $U_{\Delta^{\beta}}$ both match U_{GT} perfectly! So, this is an instance where feature extraction does its job for clustering, by improving the results obtained by the same algorithm on the transformed data. The other three partitions obtained at $m=1.05$ are identical to $U_{D_{mut}}$. An interesting conundrum: the transforms that preserve the cluster structure in D_{mut} do not yield the best matches to the ground truth.

At $m=2$, fuzziness increases, memberships are more distributed, and there are five different hardened partitions available. The max-optimal cluster validity index (ARI) achieves its maximum

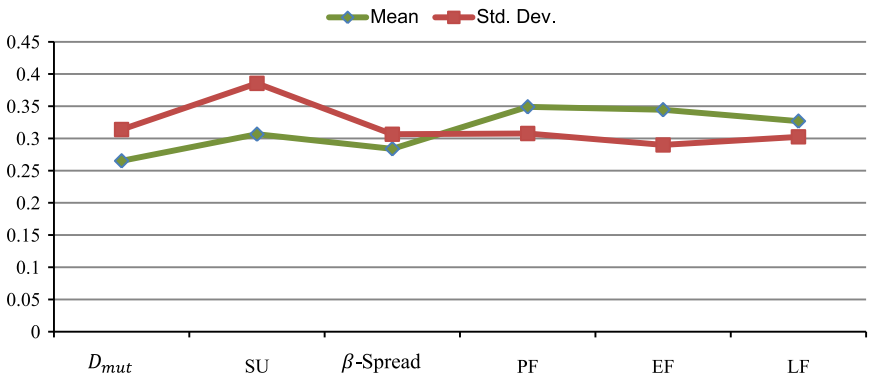

Fig. 5. The mean and standard deviation of the normalized elements in $\tilde{D}_{mut}$ for the 5 input matrices compared to D_{mut}.

value of 0.67 for the SU scheme as shown in Table 3. The minimum distortion of $U_{D_{mut}}$ by Euclideanization is realized by U_{Δ^β} (3 disagreements). Another thing we can notice from Table 2 and the $ARI(U|U_{GT})$ in Table 3 is that the PF and EF methods offer identical interpretations of this data for both values of m.

Now that we have presented the various clustering results, let's take a closer look at how every choice Δ affected the original dissimilarities. Fig. 5 shows the mean and standard deviation of the normalized dissimilarities $\tilde{D}_{mut}$, where $\tilde{D}_{mut}=\{\hat{d}_{ij}|\hat{d}_{ij}=d_{ij}/\max_{1\le i,j\le n}\{d_{ij}\}\}$ computed for every Δ. The β-spread and SU have mean μ that are very close to the original dissimilarities. Similarly, the β-spread, PF, EF and LF have standard deviation σ that are close to the original dissimilarities. The SU on the other hand, resulted in the highest standard deviation $\sigma=0.39$. This is expected from the SU as it amplifies the dissimilarities using the minimum spanning tree. However, despite the large spreading caused by the SU, it has shown to provide better results as we will see in later experiments. Please be careful to distinguish the two effects of Euclideanizing D_{mut} we have studied in this example: Fig. 5 is about the distortion of D_{mut}, whereas Table 2 is concerned with the distortion of RFCM partitions $U_{D_{mut}}$. Evidently the β-spread transform causes the least distortion from the input data (and this seems confirmed by the resultant partition information in Table 2). But the SU provides the best U_{GT} matches. Later examples will corroborate our early belief that Δ^{SU} yields the most reliable Euclideanization of D from the clustering point of view.

Example 2. GDP194 Data

The GDP194 dataset contains 194 sequences of human gene products and was obtained from ENSEMBL 2009 [17]. The relational data of the gene products was computed using a fuzzy measure similarity, which is based on Sugeno's λ measure [18]. The GDP194 characteristics are shown in Table 4, where we see that the data contains three classes and hence we will use iRFCM with $c=3$. Based on Table 4 we will assume the ground truth for GDP194 data to be $U_{GT}=[1:21 \quad 22:108 \quad 109:194]$, which indicates that the first 21 objects belong to cluster 1, the next 87 objects belong to cluster 2 and the last 86 objects belong to cluster 3.

GPD194 as used here is represented by a matrix $\sqrt{D_{194}}$ of (unsquared) dissimilarity data, which was built from the similarity data such that $\sqrt{d_{ij}}=1-s_{ij}$, where s_{ij} is the fuzzy similarity between gene products i and j. We then squared the values, obtaining D_{194}, and computed the eigenvalues of the matrix $W(D_{194,0.5})$ defined by (8). This matrix is not *p.s.d.* (there are 12 negative eigenvalues), so it is possible that RFCM will fail to execute. But, unlike D_{mut} in Example 1, which was also not *p.s.d.* but for which RFCM ran anyway, here RFCM experiences execution failure after encountering 27 negative relational distances appearing during the first iteration. At this point, we have the two options shown in Fig. 1: Euclideanize D using the 5 transformations in Table 1, or alteration of RFCM with self-healing. Since this paper is about option 1, we clustered the data using iRFCM with $m=2$, $c=3$ and the five choices of Δ (Table 1). Let U denote the fuzzy 3-partition produced by iRFCM on $\tilde{D}_{194}$ made with the SU, Fig. 6b is a visual representation of U made with an induced dissimilarity image $D(U)$.

$D(U)$ is given by

$$D(U)=1-\frac{U^TU}{\max_{\forall i,j}\{U^TU\}}, \tag{17}$$

where $(U^TU)_{ij}=\sum_{k=1}^{c}u_{ki}u_{kj}$ is the coupling of objects i and j overall c clusters. The theory underlying (17) and several other examples of the use of this induced dissimilarity measure appear in [19]. The SU view in Fig. 6b is clearly superior to the partitions produced by the other four methods and has the highest ARI, 0.98 (Table 3). Part of the SU performance is attributed to its attempt to maximize the distance between any two objects by taking the longest edge along the shortest path connecting them, thus causing a larger separation among the objects.

It was reported in [18,20] that the third family, the collagen alpha chain, is divided into three subgroups: fibril forming collagens, type IV collagens, and fibril associated collagens with interrupted triple helices. Those groups are visible in Fig. 6b, in the lower right corner.

The β-spread transformation result - the all black image in view 6c - is very interesting. The β-spread is widely cited approach in the literature for Euclideanizing D, but for clustering it is not clear that this is the best choice. The induced partition dissimilarity in Fig. 6c shows no clusters and its ARI reported in Table 3 is the lowest at 0.41. The dissimilarities in the GDP194 are bounded such that $0\le d_{ij}\le 1$, so adding a large constant to all of the off-diagonal dissimilarities can distort the structure of the data. Subsequently, this causes a large difference between D_{194} and $\tilde{D}_{194}$. In this case adding the constant, $\beta=17.28$, to the dissimilarities makes it harder for iRFCM to distinguish the objects and hence iRFCM assigns a membership $u_{ik}\approx 1/c$, where $c=3$ to every object. This is one interpretation for the black image in Fig. 6c–the image we call the "black image of death."

The PF image in Fig. 6d of the dissimilarity induced by the iRFCM partition of D_{194} using (17) suggests that the data contain two compact clusters. The first cluster which is the first block along the diagonal corresponds to the 87 sequences in the receptor precursor family. The second cluster corresponds to the first sub-cluster identified in the SU case, which is the fibril forming collagens family.

The EF in Fig. 6e suggests a different interpretation of this data. The most notable cluster is the black block in Fig. 6e corresponding to a subgroup of the receptor precursor family, which are the sequences having the gene FGFR2. The lighter color block contains the sequences in fibril forming collagens subgroup. The LF in Fig. 6f has some resemblance to the PF in Fig. 6d and both have the same ARI as shown in Table 3.

There are two take-away messages from this example: (i) the SU is clearly the *best* way to convert D_{194} to $\tilde{D}_{194}$ using (13) to preclude execution failure of RFCM; and (ii) the β-spread is clearly the *worst* of the five methods considered here.

Returning to Table 3, the SU transformation achieves an ARI value of 0.98, indicating that the 3-partition based on SU Euclideanization is again superior to the other four methods.

Table 4
Characteristics of the GDP194 dataset.

ENSEMBL Family ID	F_i=Protein Family	Gene Symbols	No. of Genes	No. of Sequences
339	Myotubularin	MTMR1÷4, MTMR1÷4	7	21
73	Receptor Precursor	FGFR1÷4, RET, TEK, TIE1	7	87
42	Collagen Alpha Chain	COL1A2, COL21A2, COL24A2, COL27A2, COL2A1, COL3A1, COL4A1, COL4A2, COL4A3, COL4A6, COL5A3, COL9A1, COL9A2	13	86

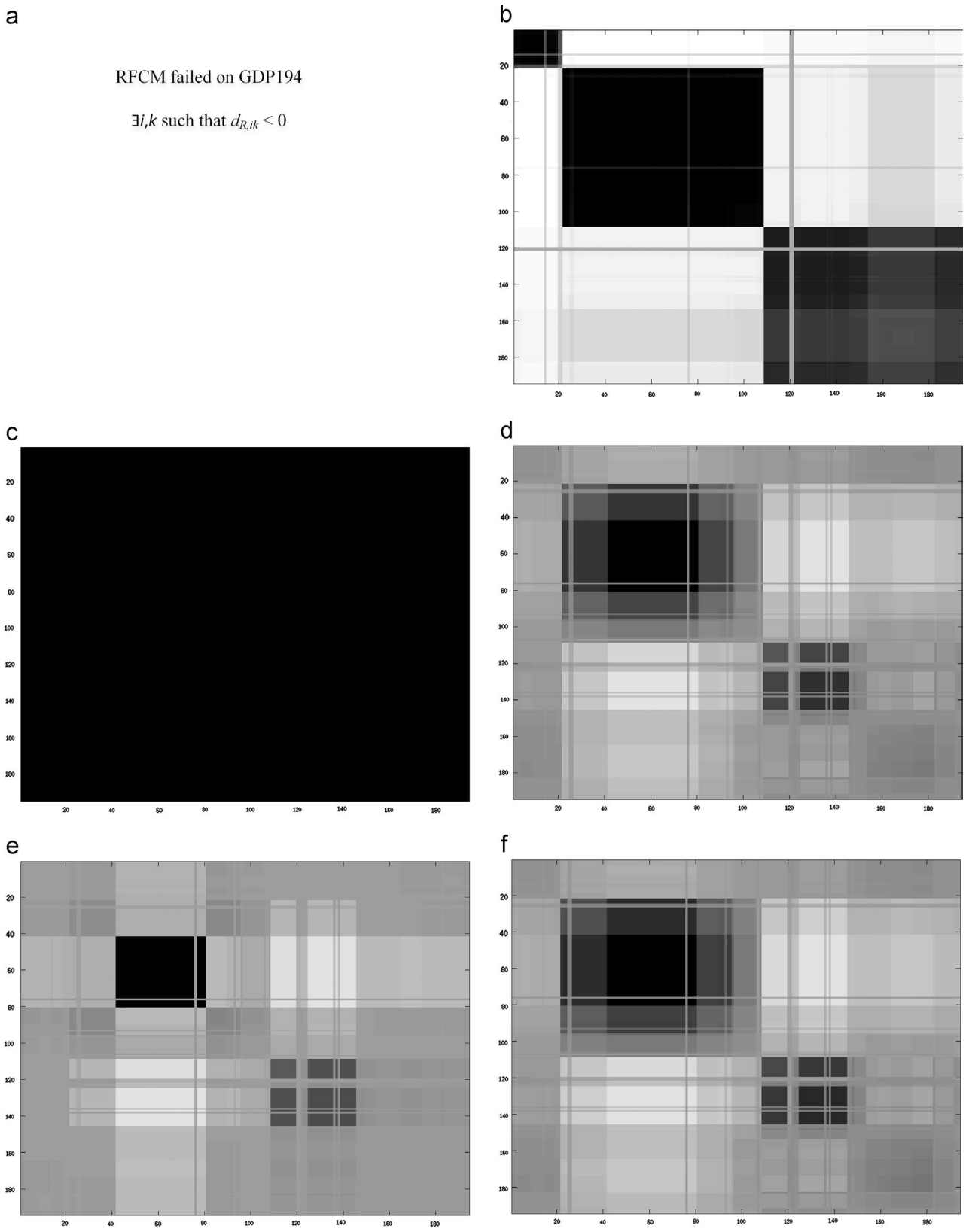

Fig. 6. The induced dissimilarity images $D(U)$ produced from clusterings of the GDP194 dataset using iRFCM with different choices of Δ and $c=3$. (a) RFCM(D_{194}, c), (b) iRFCM(D_{194}, c, Δ^{SU}), (c) iRFCM($D_{194}, c, \Delta^{\beta}$), (d) iRFCM($D_{194}, c, \Delta^{\mathrm{PF}}$), (e) iRFCM($D_{194}, c, \Delta^{\mathrm{EF}}$) and (f) iRFCM($D_{194}, c, \Delta^{\mathrm{LF}}$).

Example 3. Iris Data

Anderson's Iris data X_{Iris}, collected by Anderson in 1935 comprises $n=150$ feature vectors in $p=4$ dimensions [21,22]. Each vector in Iris has one of three (crisp) physical labels corresponding to the Iris subspecies it belongs to: Setosa, Versicolor, or Virginica. This famous data set has probably appeared in more clustering papers than any

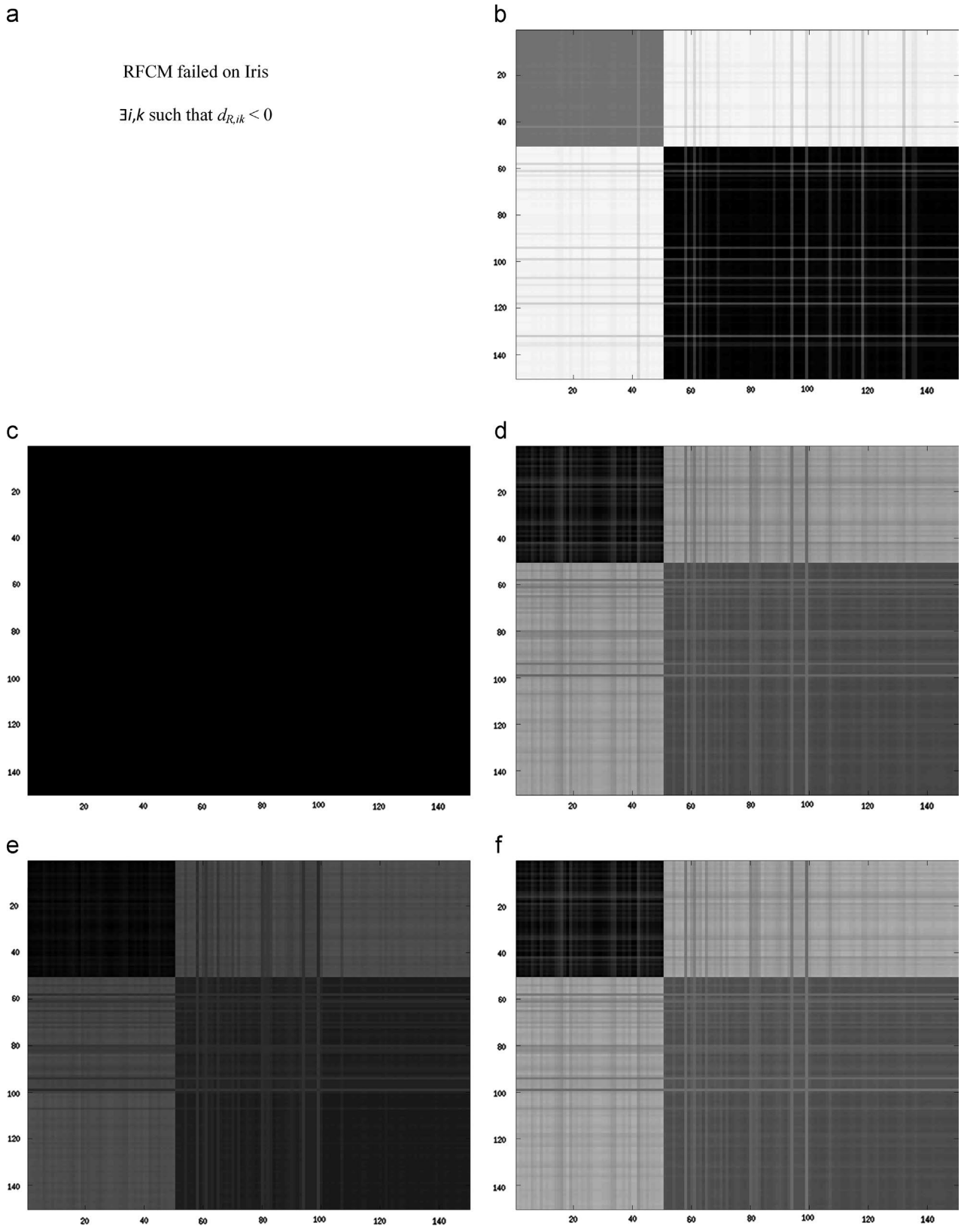

Fig. 7. The induced dissimilarity images $D(U)$ produced from clusterings of the Iris dataset using iRFCM with different choices of Δ and $c=3$. (a) RFCM(D_{Iris}, c), (b) iRFCM(D_{Iris}, c, Δ^{SU}), (c) iRFCM($D_{Iris}, c, \Delta^{\beta}$), (d) iRFCM($D_{Iris}, c, \Delta^{PF}$), (e) iRFCM($D_{Iris}, c, \Delta^{EF}$) and (f) iRFCM($D_{Iris}, c, \Delta^{LF}$).

other dataset on the planet. Perhaps the most interesting property of X_{Iris} is that this data has 3 classes (physically labeled) representing the ground truth ($U_{GT} = [1:50 \quad 51:100 \quad 101:150]$), but only 2 computer point of view clusters. Let's find out what iRFCM thinks.

We begin with X_{Iris}, and construct from it the matrix D_{Iris}. The ij-th entry of this matrix is the square of the sup norm between x_i and x_j, i.e., $d_{ij} = ||x_i - x_j||^2_{sup}$. The matrix $W(D_{Iris,0.5})$ is indefinite (it has 73 eigenvalues < 0 and 77 eigenvalues ≥ 0), so D_{Iris} is not

Euclidean. Similar to Example 2, we find that RFCM ($m=2$, $c=3$) fails to execute directly on D_{Iris}, so we compute $\tilde{D}_{Iris}$ via (13) with the five transformations at (14) which make it Euclidean. The largest negative eigenvalue of $W(D_{Iris,0.5})$ is 16.977, so adding this value to all of the off-diagonal elements of D_{Iris}, as the β-spread does at (14a), makes it Euclidean.

Fig. 7 displays visual representations of the five partitions obtained by iRFCM using the five Euclideanized versions of D_{Iris} and the induced dissimilarity matrix $D(U)$ at (17). First, note that three of the five results (views b, d, and f) strongly support the conclusion that iRFCM thinks there are only $c=2$ clusters in Iris, even though we ran the algorithm with $c=3$. If you look carefully at Fig. 7e, you will see that the EF also supports this, but with much less assurance. This is consistent with our view of Iris. The β-spread partition again produces the black image of death in Fig. 7c and ARI of 0.33. Another observation is that except for the β-spread, the first 50 objects received high membership values in the first cluster and low, but almost equal memberships in the second and third clusters, while the last 100 objects received high, but almost equal membership values in the second and third clusters. Overall, it certainly appears the SU again offers the best way to extend RFCM when an execution failure occurs due to a non-Euclidean input.

Finally, a third trip to Table 3 shows that the best ARI values for the Iris data are 0.84, achieved by the PF and LF methods. But SU comes in at 0.81, so it runs a very close third to the photo finish for first and second.

4. Conclusion and discussion

RFCM is a popular algorithm for (fuzzily) clustering objects described by a dissimilarity data matrix D. But since RFCM is the relational dual of FCM, execution of the algorithm is guaranteed only when the dissimilarities in D have a Euclidean representation with an embedding dimension $p<n$. If D is not Euclidean then the duality relation will be violated and most importantly the distances $d_{R,ik}$ can become negative. There are two options to circumvent this problem. Option 2 in Fig. 1 advocates the use of a self-healing RFCM such as NERFCM, which adjusts the dissimilarities and the distances "on the fly," and only when needed, if a negative distance is encountered. The second choice (Option 1 in Fig. 1) is to Euclideanize D prior to running RFCM. This second strategy is the one pursued here, leading to a new algorithm, iRFCM.

Five different choices of Δ were used to Euclideanize D prior to clustering. Computationally, the easiest transformation to use is the β-spread. In the β-spread approach, the same constant is added to all off-diagonal dissimilarities. If the dissimilarities are small and the constant is large, as in the GDP194 data, the original structure of the data gets distorted, and with that one can lose the cluster information. Our examples suggest that the β-spread mapping delivers good news, and bad news. The good news: it minimizes the distortion between D and $\tilde{D}$; the bad news is that it seems to maximize the distortion between the partitions U_D and $U_{\tilde{D}}$. On the other hand, the SU transformation seems have the best performance when visualized using the induced partition dissimilarity. The three parametric based transformations, viz., the PF, EF and LF, have varying performance, but the main limitation of the parametric functions is finding an optimal α_0 that can Euclideanize D and produce reasonable partitions of the data. In this paper we took a simple approach and directly searched for α_0. Determining an optimal value of α for iRFCM clustering using these three transformations is a challenging and important problem that we defer to a future investigation.

Every Δ produces a different dataset that somewhat resembles the original dissimilarities. We have witnessed in the results that different choices of Δ have resulted in different clusterings. The main difference that separates the five methods into three types is the effect that they have on the original object distances. The SU distance between two objects is the maximum dissimilarity (edge) along the shortest path connecting those objects in $MST(D)$. This is the only transformation among the five that uses original data values (as opposed to transformed ones); thus, the positions of the objects are not changed to achieve Euclideanization. The PF, EF and LF mappings are all parametric, and all replace the original dissimilarities with new ones. In terms of object locations, this amounts to rearranging the underlying realization of the objects to make it Euclidean. Thus, these three transformations distribute the spread. Finally, the β-spread is the most disruptive of the five. Adding the largest negative eigenvalue of $W(D_{0.5})$ to all of the off-diagonal entries of D amounts to spreading (literally) the objects by a fixed, maximal amount, so the original dissimilarities in this conversion are all gone.

We used the external Adjusted Rand Index of Hubert and Arabie to assess the amount by which apparent clusters in ground truth partitions U_{GT} are distorted in the partitions U produced by Euclideanization → RFCM clustering → Hardening to get U. The values in Table 3 indicate the SU transform has the least damaging effect on clusters in the input data, winning two of the three races by good margins, and finishing close to the top in the third. We are wary about making a strong general conclusion from this for three reasons: (i) ground truth labels may or may not identify clusters, as seen from the perch of any clustering algorithm; (ii) two of the three examples really have assumed ground truth labels – only Iris has actual physical labels; (iii) all cluster validity functionals are notoriously fickle. Our guess is that 40 different validity measures would produce 30 sets of conclusions. All these disclaimers notwithstanding, we believe that the SU method is probably the best of the five in terms of cluster preservation.

Some limitations emerge from Euclideanizing D prior to clustering. First, it is not very scalable. It definitely works for small datasets, but as n increases so does the time needed to Euclideanize D. It will require a large amount of time to compute the SU distance, which involves the construction of the minimum spanning tree. Second, as n increases, the time to compute the smallest eigenvalue of $W(D_{0.5})$ will also increase. Recall that $W(D_{0.5})$ always has a zero eigenvalue, and many of the non-zero eigenvalues are close to zero. Actually getting the eigenvalues becomes a numerically intractable problem due to instability and scalability as n increases. Overall, the running time complexity for the five transformations is dominated by the eigensolvers, which has a run time complexity of $O(n^3)$. Large-scale parallel eigensolvers based on Message Passing Interface (MPI) exist for large matrices, but they were tested on matrices with a maximum order of less than 1 million [23]. In the age of very large data we need tools that scale to matrices at the order of billions, such as the one based on MapReduce and Hadoop proposed in [23]. The situation becomes even more computationally expensive when we search for α that makes D Euclidean because for every α, we have to evaluate whether Δ is Euclidean or not. Third, the original dissimilarities get distorted and can lose their original structure when a constant is added, which was made abundantly clear in the β-spread case. A possible and a more scalable solution to this is to use a different approach such as self-healing RFCM (NERFCM), where the dissimilarities are transformed "on the fly", only if needed, and only a small constant is added to keep the discrepancy between D and $\tilde{D}$ to the minimum, a topic that will (hopefully) be discussed further in future work.

iRFCM MATLAB toolbox, including the source code and some documentation, is available online at: https://github.com/mohammedkhalilia/iRFCM.

References

[1] R.J. Hathaway, J.W. Davenport, J.C. Bezdek, Relational duals of the c-means clustering algorithms, Pattern Recognit. 22 (2) (1989) 205–212 (Jan.).
[2] R.J. Hathaway, J.C. Bezdek, Nerf c-means: non-Euclidean relational fuzzy clustering, Pattern Recognit. 27 (3) (1994) 429–437 (Mar.).
[3] T. Cox, M. Cox, Multidimensional Scaling, 2nd ed., Chapman and Hall, London, 2000.
[4] K.V. Mardia, J.T. Kent, J.M. Bibby, Multivariate Analysis, Probability and Mathematical Statistics, Academic Press, London, 1979.
[5] J. Benasseni, M.B. Dosse, S. Joly, On a general transformation making a Dissimilarity Matrix Euclidean, J. Classif. 24 (1) (2007) 33–51 (Jun.).
[6] S. Sattath, A. Tversky, Additive similarity trees, Psychometrika 42 (3) (1977) 319–345 (Sep.).
[7] R. Prim, Shortest connection networks and some generalizations, Bell Syst. Tech. J. (1957).
[8] E. Holman, The relation between hierarchical and Euclidean models for psychological distances, Psychometrika (1972).
[9] R.L. Graham, P. Hell, On the history of the minimum spanning tree problem, IEEE Ann. Hist. Comput. 7 (1) (1985) 43–57.
[10] I. Bar-On, M. Paprzycki, High performance solution of the complex symmetric eigenproblem, Numer. Algorithms 18 (2) (1998) 195–208 (Jun.).
[11] J. Dattorro, Convex Optimization & Euclidean Distance Geometry, Meboo Publishing, USA, 2005.
[12] I. Sledge, J. Bezdek, T. Havens, and J. Keller, A relational dual of the fuzzy possibilistic c-means algorithm, in: Proceedings of the International Conference on Fuzzy Systems, 2010, pp. 1–9.
[13] L. Hubert, P. Arabie, Comparing partitions, J. Classif. 2 (1) (1985) 193–218 (Dec.).
[14] W. Fitch, E. Margoliash, Construction of phylogenetic trees, Science (1967) 279–284.
[15] W. Fitch, E. Margoliash, This week's citation classic, Curr. Contents (27) .
[16] L. Wang, U. Nguyen, J. Bezdek, C. Leckie, K. Ramamohanarao, iVAT and aVAT: enhanced visual analysis for cluster tendency assessment, Adv. Knowl. Discov. Data Min. 6118 (2010) 16–27.
[17] T.J.P. Hubbard, B.L. Aken, S. Ayling, B. Ballester, K. Beal, E. Bragin, S. Brent, Y. Chen, P. Clapham, L. Clarke, G. Coates, S. Fairley, S. Fitzgerald, J. Fernandez-Banet, L. Gordon, S. Graf, S. Haider, M. Hammond, R. Holland, K. Howe, A. Jenkinson, N. Johnson, A. Kahari, D. Keefe, S. Keenan, R. Kinsella, F. Kokocinski, E. Kulesha, D. Lawson, I. Longden, K. Megy, P. Meidl, B. Overduin, A. Parker, B. Pritchard, D. Rios, M. Schuster, G. Slater, D. Smedley, W. Spooner, G. Spudich, S. Trevanion, A. Vilella, J. Vogel, S. White, S. Wilder, A. Zadissa, E. Birney, F. Cunningham, V. Curwen, R. Durbin, X.M. Fernandez-Suarez, J. Herrero, A. Kasprzyk, G. Proctor, J. Smith, S. Searle, P. Flicek, Ensembl 2009 (no. Database issue), Nucleic Acids Res. 37 (2009) D690–D697 (Jan.).
[18] M. Popescu, J.M. Keller, J.A. Mitchell, Fuzzy measures on the gene ontology for gene product similarity, {IEEE/ACM} Trans. Comput. Biol. Bioinforma (2006) 263–274.
[19] J. Huband, J. Bezdek, VCV2–Visual cluster validity, Comput. Intell. Res. Front (2008).
[20] T.C. Havens, J.M. Keller, M. Popescu, Computing with words with the ontological self-organizing map, Fuzzy Syst. IEEE Trans. 18 (3) (2010) 473–485.
[21] E. Anderson, The Irises of the Gaspe Peninsula, Bull. Am. Iris Soc. 59 (1935) 2–5.
[22] J.C. Bezdek, J.M. Keller, R. Krishnapuram, L.I. Kuncheva, N.R. Pal, Will the real iris data please stand up? IEEE Trans. Fuzzy Syst. 7 (3) (1999) 368–369 (Jun.).
[23] U. Kang, B. Meeder, C. Faloutsos, Spectral analysis for billion-scale graphs: discoveries and implementation, Adv. Knowl. Discov. Data Min. 6635 (2011) 13–25.

Mohammed A. Khalilia received a Ph.D. in computer science (2014) from the University of Missouri. His research interests include pattern recognition, computational intelligence and natural language processing.

James Bezdek has a Ph.D., Applied Mathematics, Cornell, 1973; past president - NAFIPS, IFSA and IEEE CIS; founding editor – Int'l. Jo. Approximate Reasoning, IEEE Transactions on Fuzzy Systems; Life fellow – IEEE and IFSA; recipient – IEEE 3rd Millennium, IEEE CIS Fuzzy Systems Pioneer, IEEE Frank Rosenblatt TFA, IPMU Kempe de Feret Medal.

Mihail Popescu is currently an Associate Professor with the Department of Health Management and Informatics, University of Missouri. His research interests include eldercare technologies, fuzzy logic, ontologies and pattern recognition.

James M. Keller holds the University of Missouri Curators Professorship in the Electrical and Computer Engineering and Computer Science Departments on the Columbia campus. He is also the R. L. Tatum Professor in the College of Engineering. His research interests include computational intelligence, computer vision, pattern recognition, and information fusion.