Fuzzy Relational Fixed Point Clustering

ROELOF K. BROUWER Senior Member IEEE
Department of Computing Science
Thompson Rivers University
Kamloops
British Columbia
V2C 5N3
Canada

Received: 21-06-2008
Revised: 27-01-2009

Abstract

The proposed relational fuzzy clustering method, called FRFP (fuzzy relational fixed point), is based on determining a fixed point of a function of the desired membership matrix. The ethod is compared to other relational clustering methods. Simulations show the method to be very effective and less computationally expensive than other fuzzy relational data clustering methods. The membership matrices that are produced by the proposed method are less crisp than those produced by NERFCM and more representative of the proximity matrix that is used as input to the clustering process.

*Keywords :*Fuzzy clustering, cluster quality, optimization, multidimensional scaling, visual representation, relational fuzzu clustering, prototype-less fuzzy clustering, clustering quality

1 Introduction

The first stage of knowledge acquisition concerning a group of objects is to partition or divide the objects into groups based either on individual object (IO) data such as feature vectors or on inter-object (object relational) (OR) data incorporated in proximity matrices.

Partitioning (clustering) algorithms may be divided according to whether the intended grouping is a crisp partition, a fuzzy partition or a hierarchy which is a set of nested partitions. In case of fuzzy clustering an object is assigned to all clusters to varying degrees that sum to one for each object in case of probabilistic fuzzy clustering. This degree may be close to zero for some objects and clusters. Fuzzy clustering is a generalization of crisp or hard clustering whereby objects are assigned completely to a single cluster.

Non-hierarchical clustering algorithms may also be divided according to the data that is provided as input. The two types of data are individual object (IO) data and object relational (OR) data. The former includes data about an individual object while the latter includes data about relationships between objects. A typical example of IO data is a set of feature vectors with a feature vector for each object. An example of OR data is a proximity (similarity or dissimilarity) matrix specifying the relationship between all pairs of objects. In areas like the social sciences, management and industrial engineering, OR data is most often available. An example of proximity measurements in the social sciences are the results of experiments in psychometric applications[1] in which a subject has been asked to rate the similarity between pairs of stimuli.

An algorithm that uses the proximity matrix of inter-object proximities directly as input is called an object relational data clustering (ORDC) [2-5] algorithm or relational clustering algorithm for short and an algorithm that uses object data is called individual object data clustering (IODC) algorithm or object data clustering algorithm for short.

Fuzzy ORDC algorithms may be based on optimizing a cost function, on creating hierarchies, on decomposition of fuzzy equations or, as in the case of the proposed method, on finding the fixed point of a function.

A proximity matrix accumulates the pair wise indices of proximity and may be either a similarity index or a dissimilarity index. Similarity measurements, $\sigma_{i,j}$, may

have to be transformed to the corresponding dissimilarity scores $\boldsymbol{\delta}_{i,j}$, if visualization of the inter-object proximities is desired as in this paper, by using (1).

$$\boldsymbol{\delta}_{i,j} = \sqrt{1-\boldsymbol{\sigma}_{i,j}} \tag{1}$$

Proximity data can be readily obtained from feature vectors if the components of the feature vectors are numeric but not so readily if they are ordinal or nominal.
In this paper it is assumed that either pattern vectors are not available or that meaningful ways of aggregating patterns are not available but that a proximity matrix is available or can be calculated. The method of relational clustering proposed in this paper is based on finding fixed points of functions of the membership matrix rather than on objective functions to be optimized
In addition to producing vectors for automatic clusterering, two-dimensional vectors are also produced to allow for visual clustering as a means for evaluating the result of automatic clustering. The visual representation is compared to the membership matrices that are obtained by automatic clustering and also to the values of a clustering quality index.
The remainder of the paper is organized as follows. First is a description of previous work. This is followed by descriptions of other relational clustering methods. Next is a section on clustering quality indices followed by a section on experiments trying out the various methods. The final section is a summary with conclusions.
To allow variable names of more than one letter and thereby permit variable names to be mnemonic, all operations are denoted by explicit operators. Multiplication, for example, is defined explicitly using the operator ×. Implicit multiplication does not exist in this paper and *na* for example is just a variable name. Names of arrays are in bold font. Rank-1 array variables are in lower case and names of arrays of rank greater than 1 are in capital. Logic expressions evaluate to 0 and 1 for false and true respectively so that they can be used in arithmetic expressions.

2 Previous Work

There are several different implementations of relational clustering [3-9]. Of these, two very successful ones [3, 7] are both based on fuzzy c-means (FCM) [10]. Relational fuzzy c-means (RFCM) [3] has the restriction that in theory the proximities should be Euclidian distances. This restriction in practice often is not satisfied and its requirement is removed in another algorithm that is a modification and is called non-euclidian relational fuzzy c-means (NERFCM) [7]. The modification requires an extra step called the β-spread transformation that prevents distances between objects and pseudo-prototypes from becoming negative. RFCM and therefore also NERFCM is somewhat non-intuitive in the creation of meaningless prototypes and the main objective of the two algorithms is to provide duals of FCM. Since feature vectors are not available, centroids, that would designate prototypes otherwise are not available.
Some work done by others on relational clustering is described next. Nasraoui etal. [11] introduce the Relational Fuzzy C—Maximal Density Estimator (RFC-MDE) algorithm that is robust and can deal with outliers as is typical in web mining. It can work on non-euclidian relational data. Krishnapuram etal. [12] present new algorithms called fuzzy c-moids (FCMdd) and robust fuzzy c-moids (RFCMdd) for clustering of objects based only on relational data. Objects, called medoids, one for each cluster, are selected in such a way that the total fuzzy dissimilarity within each cluster is minimized. The originators make the claim that the FCMdd method is more efficient than RFCM which is very important when it comes to web mining where it is proposed to be applied. Dave and Sen [9, 13] propose a new algorithm, called FRC, that is a generalization of FANNY [8] with one of the objectives being increased robustness.
Cimino etal. [14] propose a new approach to robust fuzzy relational clustering that puts no restriction on the proximity matrix. Hathaway etal. [15] propose a kernalized form of the non-euclidian relational fuzzy c-means algorithm. Denoeux and Masson[16] introduce a new relational clustering algorithm that is based on the Dempster-Shafer theory of belief functions.
Another paper that studies fuzzy clustering without prototypes is by Borgelt [17]. An iterative update rule is derived from an objective function that only involves the proximities between data points and the membership degrees of the data points in the various clusters.

3 Existing Relational Clustering Methods that are Compared to the Proposed Clustering Method

The methods used in the comparison to the proposed method are described next.

3.1 Roubens

Roubens [4], following the ideas of Dunn [18] and Ruspini [19] , proposes a non-metric and fuzzy method that is based on the objective function (2) that is to be minimized.

$$\begin{aligned} &\sum_{i,j,k} \boldsymbol{M}_{i,k}^2 \times \boldsymbol{M}_{j,k}^2 \times \boldsymbol{D}_{i,j} \\ &\boldsymbol{M}_{i,k} \geq 0 \quad \forall k,i \qquad \sum_k \boldsymbol{M}_{i,k} = 1 \quad \forall i \end{aligned} \tag{2}$$

with solution

$$\boldsymbol{M}_{i,k} = \frac{\dfrac{1}{\sum_j \boldsymbol{M}_{j,k}^2 \times \boldsymbol{D}_{i,j}}}{\sum_l \dfrac{1}{\sum_j \boldsymbol{M}_{j,l}^2 \times \boldsymbol{D}_{i,j}}} \tag{3}$$

$\boldsymbol{D}$ is the dissimilarity matrix. Each row of $\boldsymbol{M}$ corresponds to an object and each column corresponds to a cluster. The effect of the normalizing denominator ensures that the values in the rows of $\boldsymbol{M}$ sum to 1.

According to the author [4] "the global minimum of this non-linear program is computationally impractical for moderate or large data sets". We can get an approximation by using (3) iteratively until there is little change. The result depends on the initial value that is used for $\boldsymbol{M}$ however. The solution $\boldsymbol{M}$ is actually a fixed point of the function on the right of (3) obtained by optimizing an objective function however rather than directly.

3.2 Windham's AP

The relational clustering method by Windham [5], called the assignment-prototype(AP) algorithm, is very interesting in that although it does not make use of object-attribute data to determine prototypes, it provides prototype information. Each object is a prototype for a cluster to a certain degree. The constraint is that the sum of the degrees for a prototype is 1 just as the sum of the degrees to which an object belongs to clusters is 1. The objective function to be minimized is

$$\begin{aligned} &\sum_{k,i,j} \boldsymbol{M}_{i,k}^2 \times \boldsymbol{P}_{j,k}^2 \times \boldsymbol{D}_{i,j} \\ &\boldsymbol{M}_{i,k} \geq 0 \quad \forall k,i \qquad \sum_k \boldsymbol{M}_{i,k} = 1 \quad \forall i \\ &\boldsymbol{P}_{i,k} \geq 0 \quad \forall k,i \qquad \sum_i \boldsymbol{P}_{i,k} = 1 \quad \forall k \end{aligned} \tag{4}$$

That is the weighted sum of the distances between objects is minimized as in (2). However in (2) both objects play the same role and both weights are membership values while in (4) one of the objects plays the role of a prototype and one of the weights is the degree to which it is a prototype for a type. By comparison when clustering is done using attribute vectors, the attribute vector for a prototype is the weighted average of attribute vectors. Generally the resulting average attribute vector does not represent an actual object. The solution to (4), found by using Lagrangian multipliers, is

$$\boldsymbol{M}_{i,k} = \frac{\dfrac{1}{\sum_j \boldsymbol{P}_{j,k}^2 \times \boldsymbol{D}_{i,j}}}{\sum_l \dfrac{1}{\sum_j \boldsymbol{P}_{j,l}^2 \times \boldsymbol{D}_{i,j}}}$$

$$\boldsymbol{P}_{i,k} = \frac{\dfrac{1}{\sum_j \boldsymbol{M}_{j,k}^2 \times \boldsymbol{D}_{j,i}}}{\sum_i \dfrac{1}{\sum_j \boldsymbol{M}_{j,k}^2 \times \boldsymbol{D}_{j,i}}} \tag{5}$$

The division by the denominator ensures that the values in the rows of $\boldsymbol{P}$ sum to 1 as before. If we substitute the right side of the second expression in the first expression in (5) we get a function of $\boldsymbol{M}$ only as before and therefore a fixed point expression in $\boldsymbol{M}$.

3.3 NERF C-means

The non-euclidian relational fuzzy c-means appears below as described by their originators, Hathaway and Bezdek in [7]. The method is identical to relational fuzzy c-means (RFCM) originated by the same authors but with the addition of a spreading transformation (step (8)) on the proximity matrix.

Algorithm 1 NERFCM

-given a similarity matrix $\boldsymbol{R}=\boldsymbol{R}^{(1)}$

-set the components of $\boldsymbol{M}$ to random values satisfactory for a membership matrix

-calculate pseudo-prototypes

$$\boldsymbol{v}_k = \frac{(\boldsymbol{M}_{k,1}^m, \boldsymbol{M}_{k,2}^m, ..\boldsymbol{M}_{k,n}^m)}{\boldsymbol{M}_{k,1}^m + \boldsymbol{M}_{k,2}^m + ..\boldsymbol{M}_{k,n}^m} \qquad k = 1..nk \tag{6}$$

-calculate the distances between pseudo-prototypes and objects by

$$\boldsymbol{D}_{k,p} = (\boldsymbol{R}^{(\beta)} \cdot \boldsymbol{v}_k)_p - \frac{\boldsymbol{v}_k \cdot \boldsymbol{R}^{(\beta)} \cdot \boldsymbol{v}_k}{2} \quad k=1..nk \quad p=1..np \tag{7}$$

-if $\boldsymbol{D}_{k,p} < 0$ for any k and p then

$$\Delta\beta = \max_{(k,p)}\left\{-2\times\frac{\boldsymbol{D}_{k,p}}{\left\|\boldsymbol{v}_k-\boldsymbol{e}_p\right\|^2}\right\}$$

$$\boldsymbol{D}_{k,p} \leftarrow \boldsymbol{D}_{k,p} + \frac{\Delta\beta}{2}\times\left\|\boldsymbol{v}_k-\boldsymbol{e}_p\right\|^2$$

$$\beta \leftarrow \beta + \Delta\beta$$

- modify the proximity matrix according to

$$\boldsymbol{R}^{(\beta)} \leftarrow \boldsymbol{R}^{(\beta)} + \beta\times(1\otimes 1-\boldsymbol{I})$$

if $\boldsymbol{D}_{k,p} > 0$ for all k, then

$$\boldsymbol{M}_{k,p} = \frac{1}{\left(\boldsymbol{D}_{k,p}\times\left(\frac{1}{\boldsymbol{D}_{1,p}}+\frac{1}{\boldsymbol{D}_{2,p}}+..+\frac{1}{\boldsymbol{D}_{nk,p}}\right)\right)^{\frac{1}{m-1}}}$$

otherwise

$$\boldsymbol{M}_{k,p}=0 \text{ if } \boldsymbol{D}_{k,p}>0, \boldsymbol{M}_{k,p}\in[0,1], \text{ and } (\boldsymbol{M}_{1,p}+\boldsymbol{M}_{2,p}+..\boldsymbol{M}_{nk,p})=1$$

if difference in $\boldsymbol{M}$ is small then stop

(9)

Logically this algorithm is much more complex than the other two.

4 Proposed Fixed Point Clustering Method

For a crisp partition it is desirable that each element belongs to the cluster with which it has the closest affinity or to which it is most similar. Formally we desire that

$$\boldsymbol{M}_{p,k} = (k \doteq \arg\max_{l} s(\boldsymbol{x}_p, C_l)) \qquad (10)$$

The operator $\doteq$ represent logical equality. Logic expressions evaluate to 0 and 1 for false and true respectively so that they can be used in arithmetic expressions. $s(\boldsymbol{x}_p, C_l)$ is the similarity or affinity between object x_p and cluster C_l that can be defined in several ways. $\boldsymbol{M}_{p,k}$ is a binary variable here denoting whether or not $\boldsymbol{x}_p$ belongs to cluster C_k. This is a winner take all situation with a cluster getting all or nothing of an object in terms of belongingness.

Let us now consider various expressions for $s(\boldsymbol{x}_p, C_k)$ in terms of similarities between two elements. One method is to define the similarity between clusters and elements as the similarity between the prototype for C_k and $\boldsymbol{x}_p$. This gives us the desired outcome of the k-means method of clustering.

For prototype-less clustering the similarity between C_k and $\boldsymbol{x}_p$ may be defined in terms of similarities between $\boldsymbol{x}_p$ and elements of C_k in various other ways such as maximum similarity, minimum similarity, or average similarity. In the following $\boldsymbol{S}_{p,q} \equiv s(\boldsymbol{x}_p, \boldsymbol{x}_q)$ where $s(\boldsymbol{x}_p, \boldsymbol{x}_q)$ is the similarity between $\boldsymbol{x}_p$ and $\boldsymbol{x}_q$. (8)

For minimum similarity we set

$$\begin{aligned} s(\boldsymbol{x}_p, C_k) &= \min_{\boldsymbol{x}_q \in C_k} \boldsymbol{S}_{p,q} \\ &= \min_{q} \lim_{\delta\to 0} \frac{\boldsymbol{S}_{p,q}}{(\boldsymbol{x}_q \in C_k)+\delta} \\ &= \min_{q} \lim_{\delta\to 0} \frac{\boldsymbol{S}_{p,q}}{\boldsymbol{M}_{q,k}+\delta} \end{aligned} \qquad (11)$$

For maximum similarity we set

$$\begin{aligned} s(\boldsymbol{x}_p, C_k) &= \max_{\boldsymbol{x}_q \in C_k} \boldsymbol{S}_{p,q} \\ &= \max_{q} \boldsymbol{S}_{p,q}\times(\boldsymbol{x}_q \in C_k) \\ &= \max_{q} \boldsymbol{S}_{p,q}\times\boldsymbol{M}_{q,k} \end{aligned} \qquad (12)$$

For average similarity we set

$$\begin{aligned} s(\boldsymbol{x}_p, C_k) &= \frac{\sum_{q=1}^{np} \boldsymbol{S}_{p,q}\times(\boldsymbol{x}_q \in C_k)}{\sum_{q=1}^{np}(\boldsymbol{x}_q \in C_k)} \\ &= \frac{\sum_{q=1}^{np} \boldsymbol{S}_{p,q}\times\boldsymbol{M}_{q,k}}{\sum_{q=1}^{np}\boldsymbol{M}_{q,k}} \end{aligned} \qquad (13)$$

Where we have replaced $(\boldsymbol{x}_q \in C_k)$ with $\boldsymbol{M}_{q,k}$.

A soft approach is to allow a pattern to belong to all clusters to some degree. This can be accomplished by replacing (10) with (14) to define the condition for a fuzzy partition.

$$\boldsymbol{M}_{p,k} = \frac{s(\boldsymbol{x}_p, C_k)}{\sum_{l=1}^{nk} s(\boldsymbol{x}_p, C_l)} \qquad (14)$$

If we replace $s(\boldsymbol{x}_p, C_k)$ using (11), (12) and (13) we get respectively the following.

For minimum similarity we get

$$\boldsymbol{M}_{p,k} = \frac{\min_q \lim_{\delta \to 0} \frac{\boldsymbol{S}_{p,q}}{\boldsymbol{M}_{q,k}^m + \delta}}{\sum_{l=1}^{nk} \min_q \lim_{\delta \to 0} \frac{\boldsymbol{S}_{p,q}}{\boldsymbol{M}_{q,l}^m + \delta}} \tag{15}$$

For maximum similarity we get

$$\boldsymbol{M}_{p,k} = \frac{\max_q \boldsymbol{S}_{p,q} \times \boldsymbol{M}_{q,k}^m}{\sum_{l=1}^{nk} \max_q \boldsymbol{S}_{p,q} \times \boldsymbol{M}_{q,l}^m} \tag{16}$$

For average similarity we get

$$\boldsymbol{M}_{p,k} = \frac{\frac{\sum_{q=1}^{np} \boldsymbol{S}_{p,q} \times \boldsymbol{M}_{q,k}^m}{\sum_{q=1}^{np} \boldsymbol{M}_{q,k}^m}}{\sum_{l=1}^{nk} \frac{\sum_{q=1}^{np} \boldsymbol{S}_{p,q} \times \boldsymbol{M}_{q,l}^m}{\sum_{q=1}^{np} \boldsymbol{M}_{q,l}^m}} \tag{17}$$

$\boldsymbol{M}_{p,k}$ is the membership of entity p in cluster k. An exponent m that effects the degree of fuzziness has been added also. Proximity or relational matrices are generally provided as distance matrices rather than as similarity matrices, **S**. Distances can be converted to similarities using

$$s = \frac{1}{1+d} \tag{18}$$

Each of the expressions on the right side of the equation in (15), (16), and (17) is a continuous function of $\boldsymbol{M}$ on the domain $\Re^{npxnk}$ and therefore Brouwer's fixed point theorem applies.
Therefore (15), (16), and (17) should have a solution.

5 Fixed Point Clustering and the Cluster Affinity Search Technique

The method proposed bears some similarity in principle to what is called the cluster affinity search technique (CAST) [20]. The input to CAST is a pair consisting of a similarity matrix and an affinity threshold. The algorithm is described in detail in [20]. Clusters are constructed one at a time. Let C_{open} be the current cluster under construction. The affinity of an element x with respect to the current cluster is defined by

$$a_p(C_{open}) = \sum_{l=1}^{n} \boldsymbol{S}_{p,l} \times (x_l \in C_{open}) \tag{19}$$

An element x_p has high affinity for C_{open} if $a_p \geq t \times \left| C_{open} \right|$, otherwise it has low affinity. Elements of low affinity are removed from C_{open} alternately with the addition of high affinity elements. When there is no further change the cluster is closed and a new cluster is opened.
A potential modification of the preceding algorithm is to start with a certain number of clusters, each with one element, and add a new element to the cluster with which it has maximum affinity with affinity defined as before. That is an element is added to the cluster with which it has maximum affinity. This is like k-means with affinity defined as the inverse of the distance to the cluster prototype.
We can modify (19) so that affinity of element x_p for a cluster, C_k, under consideration is a weighted average or an average over the elements in the cluster

$$a_p(C_k) = \frac{\sum_{l=1}^{n} \boldsymbol{S}_{p,l} \times (x_l \in C_k)}{\sum_{l=1}^{n} (x_l \in C_k)} \tag{20}$$

The element x_p is then added to the cluster with which it has maximum affinity. We can normalize the affinities by ensuring that the sum of the affinities of a pattern for a cluster is one which gives us

$$a_p(C_k) = \frac{\frac{\sum_{l=1}^{n} \boldsymbol{S}_{p,l} \times (x_l \in C_k)}{\sum_{l=1}^{n} (x_l \in C_k)}}{\sum_{r=1}^{nk} \frac{\sum_{l=1}^{n} \boldsymbol{S}_{p,l} \times (x_l \in C_r)}{\sum_{l=1}^{n} (x_l \in C_r)}} \tag{21}$$

It becomes natural now to interpret the affinity of an element for a cluster as the degree of membership in that cluster. We also replace $x_l \in C_k$ by its generalization $\boldsymbol{M}_{l,k}$ which denotes degree of membership and we get

$$\boldsymbol{M}_{p,k} = \frac{\dfrac{\sum_{l=1}^{n} \boldsymbol{S}_{p,l} \times (\boldsymbol{M}_{l,k})^m}{\sum_{l=1}^{n} (\boldsymbol{M}_{l,k})^m}}{\sum_{r=1}^{nk} \dfrac{\sum_{l=1}^{n} \boldsymbol{S}_{p,l} \times (\boldsymbol{M}_{l,r})^m}{\sum_{l=1}^{n} (\boldsymbol{M}_{l,r})^m}} \tag{22}$$

The exponent m is used to control the degree of fuzziness and is generally set to 2.

6 Quality of a Clustering Process and Quality of Clusters

6.1 Introduction

Many functions have been proposed for validation of fuzzy clustering [21-24] in case of fuzzy c-means when feature vectors and prototypes are available but ones for relational fuzzy clustering are rare.

A sample of some of the papers dealing with cluster validation is presented by the list [18, 22, 23, 25-30]. Most of them either treat only crisp clustering or they depend on determining centroids which is not possible unless IO data is available.

There are two aspects to measuring quality as it relates to clustering. One is the quality of the clustering process and the other is the quality of the clusters. Determining the quality of a clustering process is quite straightforward when there is a correct partition to which a comparison can be made. However that is generally not so. Bezdek etal. [27] separate tendency towards clustering (clusterability) from validation. The first is the tendency of the objects, based on data, towards clustering. Clusterability is difficult to determine unless the object's feature vectors are 2-dimensional and there is a visual representation of the interobject distances. Thus ultimately the validity of the result of clustering is best determined if visual clustering is supported by the result of the automatic clustering. Therefore in this work the OR data is also turned into two-dimensional IO data for visualization. The result of clustering, as expressed by the membership matrix, can then be compared to the visual representation.

Cluster(ing) quality indices can be divided into those that are only a function of the membership matrix that is produced by clustering and those that also depend on the proximity matrix. Cluster quality requires only the membership matrix and clustering quality requires also data that is input to the clustering process.

First we will discuss some cluster quality indices as ones that should not be used here because of their limitations. In these indices quality is measured by how close the membership matrix is to a discriminant matrix. A measure that is based only on the fuzzy membership as proposed by Bezdek [31] and also by Roubens [4], determines how close a fuzzy partition obtained by a clustering algorithm is to being crisp. It is expressed by

$$\frac{\sum_{p=1}^{np} \sum_{k=1}^{nk} (\boldsymbol{M}_{p,k})^2}{np} \tag{23}$$

Bezdek calls this the partition coefficient. It is the average amount of membership sharing. Bezdek [31] considers its monotonic tendency and the lack of a direct connection to the object data to be its major disadvantages. In normalized form it is equivalent to the normalized form of a partition coefficient by Dunn [22]. A measure proposed by Roubens [4] also determines how close a fuzzy partition is to a hard partition and is also equivalent to the normalized form of Dunns [22] partition coefficient. Another measure based only on the partition or membership matrix is classification entropy [32]. It expresses the amount of fuzziness in $\boldsymbol{M}$. Low values here means that the result of clustering is close to being crisp. It does not take into account the proximity matrix for the objects that are clustered.

A crisp partition is not however the desired outcome of the clustering process if it does not reflect the true partitioning structure of the data otherwise we would always turn every fuzzy partition into a crisp partition as its limiting case. (The membership matrix for the limiting case is obtained by replacing the maximum membership by one and the others by 0 for each object.) The quality of the clustering process should be a function of the proximity matrix(the input) and the membership matrix (the output). The membership matrix should be a reflection of the proximity matrix.

Following are some potential clustering quality indices that do take proximity matrices into account. Dunn's validity index [18] for crisp clustering is the distance between two clusters divided by the diameter of a cluster. In Dunn's particular case the distance between two clusters is taken to be the smallest distance between two objects with one object from each cluster and the diameter of a cluster is the largest distance between two objects from the same cluster. In the general expression (24) i and j identify different versions of distance between sets (clusters) and diameters of sets(clusters). The indices k,k_1,and k_2 identify clusters.

$$V = \frac{\min\limits_{k_1,k_2 \;\; k_1 \neq k_2} \delta_i(C_{k_1}, C_{k_2})}{\max\limits_{k} \Delta_j(C_k)} \tag{24}$$

Bezdek and Pal [27] consider a generalization of Dunn's index [18] for crisp clustering. They consider 6

definitions of set distance and 3 definitions of set diameter leading to 18 possibilities for crisp clustering quality indices. The first 2 set definitions are too much effected by outliers. The 4th and 5th require centroids of sets and the 6th which is the Hausdorff definition of distance between two sets is computationally intensive. This leaves the 3rd definition for inter-cluster distance that is defined as

$$\delta_{avg}(S,T) = \frac{1}{|S| \times |T|} \sum_{\substack{x \in S \\ y \in T}} d(x,y) \tag{25}$$

i.e. the average distance between the elements of the two sets.

The first definition of cluster diameter is too sensitive to outliers and the last requires calculation of centroids. This leaves the second that is defined by

$$\Delta_{avg}(S) = \frac{1}{\frac{|S| \times (|S|-1)}{2}} \sum_{\substack{x,y \in S \\ x \neq y}} d(x,y) \tag{26}$$

6.2 *Proposed Clustering Quality Index*

We can easily generalize(25) and (26) to fuzzy clustering by applying the membership matrix as in (27).

$$\delta_{avg}(C_{k_1}, C_{k_2}) = \frac{\sum_{1 \le p_1 < p_2 \le np} \boldsymbol{M}^m_{p_1,k_1} \times \boldsymbol{M}^m_{p_2,k_2} \times d(p_1,p_2)}{\sum_{1 \le p_1 < p_2 \le np} \boldsymbol{M}^m_{p_1,k_1} \times \boldsymbol{M}^m_{p_2,k_2}}$$

$$\Delta_{avg}(C_k) = \frac{\sum_{1 \le p_1 < p_2 \le np} \boldsymbol{M}^m_{p_1,k} \times \boldsymbol{M}^m_{p_2,k} \times d(p_1,p_2)}{\sum_{1 \le p_1 < p_2 \le np} \boldsymbol{M}^m_{p_1,k} \times \boldsymbol{M}^m_{p_2,k}} \tag{27}$$

For comparing the clustering methods in this paper the following measure is used. It takes into account both the proximity matrix and the result of clustering.

Expression (24) is modified to

$$V = \frac{\underset{k_1,k_2 \;\; k_1 \neq k_2}{average} \delta_{\boldsymbol{i}}(C_{k_1}, C_{k_2})}{\underset{k}{average} \; \Delta_j(C_k)} \tag{28}$$

with the definitions in (27) for fuzzy clustering with m replaced by 2.

7 Method for visualizing the relations between the objects

The purpose of the visual clustering is to permit comparison with the automatic clustering. For the visual clustering the space is 2 dimensional. The visual clustering permits a qualitative measure of the clustering process by comparing the visual clustering to the membership matrix that is obtained by automatic clustering. This qualitative measure in turn can be compared to the quantitative measure as obtained by using (28).

A good visual representation should represent proximity information, by positioning similar elements close to one another and far from dissimilar ones. A visual representation of the proximity matrix van be obtained using multidimensional scaling (MDS) [33-39], which includes a set of algorithms, given a proximity matrix, for determining a set of vectors for the purpose of exploratory data analysis. MDS algorithms look for an arrangement of a set of objects such that the distance between any two objects matches their given proximities as closely as possible..

Only one of the many MDS algorithms, the SMACOF algorithm, is discussed and used here.

SMACOF-IB [33, 40-42] is considered to be one of the fastest, simplest, and elegant MDS algorithms. It optimizes the sum of the squares of the errors between the distances and the proximities. The algorithm below is as it appears on page 155 in [33]. It is based on majorizing the cost function (29).

$$\sigma_r(\boldsymbol{X}) = \sum_{i<j} w_{i,j} \times ((\delta_{i,j} - d_{i,j}(\boldsymbol{X}))^2$$

$$d_{i,j}(\boldsymbol{X}) = \sqrt{\sum_{a=1}^{m} (x_{i,a} - x_{j,a})^2} \tag{29}$$

The matrix, $\boldsymbol{X}$, contains the points as rows. The value of $\delta_{i,j}$ is the proximity (dissimilarity in this case) between objects $\boldsymbol{O}_i$ and $\boldsymbol{O}_j$. Fixed weight $w_{i,j}$ are 1 if $\delta_{i,j}$ are known and 0 otherwise. Following is the complete algorithm

Algorithm 2 SMACOF

The SMACOF algorithm may be summarized as

Initialization

Set components of $\boldsymbol{X}^{new}$ to random values

Calculate $\sigma(\boldsymbol{X}^{new})$ by (29)

Repeat

$\boldsymbol{X}^{old} \leftarrow X^{new}$

$\sigma(\boldsymbol{X}^{old}) \leftarrow \sigma(\boldsymbol{X}^{new})$

$\boldsymbol{X}^{new} \leftarrow$ Guttman transform of $\boldsymbol{X}^{old}$

Calculate $\sigma(\boldsymbol{X}^{new})$ by (29)

Until $\sigma(\boldsymbol{X}^{old}) - \sigma(\boldsymbol{X}^{new}) <$e or maximum allowable iterations

The Guttman transform is defined by (30)

$$X^{new} = \frac{1}{n} \times \boldsymbol{B} \cdot \boldsymbol{X}^{old}$$

$$\text{for } i \neq j$$

$$b_{i,j} = -w_{i,j} \times \frac{\delta_{i,j}}{d_{i,j}(\boldsymbol{X}^{old})} \qquad d_{i,j}(\boldsymbol{X}^{old}) \neq 0$$

$$b_{i,j} = 0 \qquad otherwise$$

$$\text{for } i = j$$

$$b_{i,i} = -\sum_{j=1, j \neq i}^{n} b_{i,j} \tag{30}$$

Fastmap [43] can also be used to provide a visual representation if computational time is a very important consideration. Unlike the MDS methods that have quadratic time complexity, it has linear time complexity but is not as accurate which in this case may not be an issue. Nevertheless SMAKOV is used since it is more common.

8 Simulations

8.1 Introduction

Simulations are done to compare the following 4 methods consisting of the proposed method and three existing methods.

1. Proposed - FRFP with average distance
2. Existing
    a. NERRFCM[3]
    b. Roubens [4]
    c. Windhams AP [5]

For fuzzy relational fixed point (FRFP) clustering all 3 formulations expressed by (15), (16) and (17) were tested. The ones based on minimum(15) and average (17) gave good results while the one based on maximum gave very poor results. The results reported here are with the use of (17) which is solved iteratively starting with a random matrix for ***M***. A value of 2 for *m* has been used throughout. For proximity dissimilarity is used rather than similarity.

The methods are compared on the basis of visual clustering and the proposed quantitative measure. As the ultimate test of clustering quality, the membership matrices that are produced through automatic clustering based on the proximity matrix, are examined to determine how well they correspond to the visual representation of the proximity matrices produced by MDS. The quantitative clustering quality measure is in turn validated by comparing its value to the result of the qualitative visual comparison.

8.2 Simulation 1 using proximity matrix 1

The first proximity matrix is Windham's, reproduced in Table 1 [5] , that is also used by Bezdek et. al. in their paper [6] in which the authors make comparisons between RFCM, Roubens and Windhams clustering algorithms. Rows and columns correspond to objects.

Table 1 Proximity matrix 1

0	**6**	**3**	**6**	**11**	25	44	72	69	72	100
6	**0**	**3**	**11**	**6**	14	28	56	47	44	72
3	**3**	**0**	**3**	**3**	11	25	47	44	47	69
6	**11**	**3**	**0**	**6**	14	28	44	47	56	72
11	**6**	**3**	**6**	**0**	3	11	28	25	28	44
25	14	11	14	3	**0**	3	14	11	14	25
44	28	25	28	11	3	**0**	**6**	**3**	**6**	**11**
72	56	47	44	28	14	**6**	**0**	**3**	**11**	**6**
69	47	44	47	25	11	**3**	**3**	**0**	**3**	**3**
72	44	47	56	28	14	**6**	**11**	**3**	**0**	**6**
100	72	69	72	44	25	**11**	**6**	**3**	**6**	**0**

The proximity matrix shows, by bolding of small proximity values, that there are 3 clusters with objects 1-5 in one cluster, object 6 in another cluster and objects 7-11 in the 3rd cluster. This was intended in the generation of the proximity matrix. Figure 1 shows a visual representation of the IO (inter-object) data obtained by the SMACOF algorithm. The visual representation supports the result of a visual inspection of the proximity matrix but also shows other possible partitions.

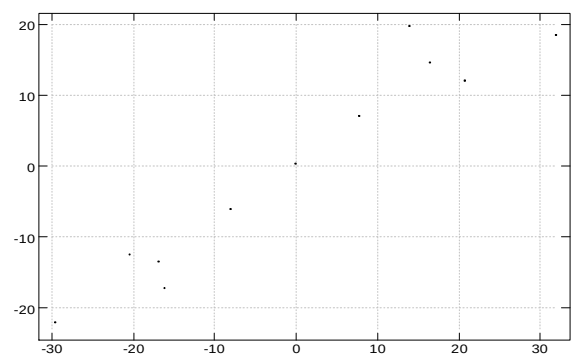

Figure 1 Representation of proximity matrix1 in 2-D

The visual representation shows that there are 7 clusters of sizes 1,3,1,1,1,3,1 or 5 clusters with sizes 4,1,1,1,4 or 3 clusters with sizes 4,3,4 or 3 clusters with sizes 5,1,5. If two clusters are permitted than one object belongs equally in both. The membership matrices in Table 2 for 2 clusters should bear this out. Bolded values are maximum values. Rows correspond to objects and columns correspond to clusters.

Table 2 Membership matrices for the 5 methods for a two cluster partition (rows correspond to objects) for proximity matrix 1

FRFP		NERFCM		Rouben		Winham	
0.05	**0.95**	**1.00**	0.00	**0.83**	0.17	0.09	**0.91**
0.08	**0.92**	**1.00**	0.00	**0.82**	0.18	0.10	**0.90**
0.07	**0.93**	**1.00**	0.00	**0.83**	0.17	0.05	**0.95**
0.08	**0.92**	**1.00**	0.00	**0.82**	0.18	0.10	**0.90**
0.17	**0.83**	**1.00**	0.00	**0.80**	0.20	0.15	**0.85**
0.50	**0.50**	0.33	**0.67**	**0.72**	0.28	**0.50**	**0.50**
0.83	0.17	0.00	**1.00**	**0.64**	0.36	**0.85**	0.15
0.92	0.08	0.00	**1.00**	**0.64**	0.36	**0.90**	0.10
0.93	0.07	0.00	**1.00**	**0.62**	0.38	**0.95**	0.05
0.92	0.08	0.00	**1.00**	**0.64**	0.36	**0.90**	0.10
0.95	0.05	0.00	**1.00**	**0.64**	0.36	**0.91**	0.09

Labeling of the objects in the membership matrices does not necessarily correspond to the labelling of the objects in the proximity matrix. The membership matrix for FRFP correlates, visually, very well with the visual representation. It is very easy to see the correspondence between objects in the membership matrices for these two methods and the points in Figure 1. There is a symmetry in the membership matrix that corresponds to the symmetry of the points representing the objects. Object 6 fits equally well in both of the 2 clusters that are allowed for in case of the proposed method and in case of Windhams method but not so in the other 2 cases. In case of the other objects all methods, except for NERFCM, show that objects, although belonging strongly to one cluster, also have some membership in the other cluster. NERFCM has tended towards a hard clustering. Table 3 shows the values for the clustering quality index. NERFCM scores best although the result is very close to that of the proposed method. The fact that, by comparing the membership matrices to the visual representation, the proposed method produced better results is not completely borne out by the values of the clustering index however. FRFP clearly provides the clustering result that is much closer to that of the visual representation. The average time over 20 runs for FRFP is also the best.

Table 3 Average quality measures and execution times for clustering using proximity matrix 1

Methods	Clustering Quality	times
FRFP	4.81	0.005
NERFCM	4.95	0.013
Rouben	1.34	0.019
Windham	2.21	0.011

8.2.1 Simulation 2 using proximity matrix 2

The second proximity matrix, shown in Table 4, is obtained using Euclidian distance between randomly generated feature vectors. The feature vectors are generated so that they fall into 5 clusters of 3 approximately. Bolding of relative small proximity values in the proximity matrix shows that there are 5 clusters of 3 as intended.

Table 4 Proximity matrix 2

0	**9**	**13**	51	45	53	90	90	93	134	134	139	179	180	186
9	**0**	**11**	52	47	55	92	92	95	136	136	140	181	182	188
13	**11**	**0**	46	42	49	85	85	88	130	129	133	174	176	181
51	52	46	**0**	**11**	**8**	40	41	43	84	84	89	129	130	136
45	47	42	**11**	**0**	**11**	46	46	49	90	89	94	134	136	141
53	55	49	**8**	**11**	**0**	37	38	40	82	81	86	126	128	133
90	92	85	40	46	37	**0**	**4**	**8**	45	45	49	90	91	97
90	92	85	41	46	38	**4**	**0**	**10**	45	45	49	90	91	97
93	95	88	43	49	40	**8**	**10**	**0**	42	41	46	86	88	93
34	136	130	84	90	82	45	45	42	**0**	**6**	**8**	46	47	52
134	136	129	84	89	81	45	45	41	**6**	**0**	**8**	46	48	53
139	140	133	89	94	86	49	49	46	**8**	**8**	**0**	42	44	49
179	181	174	129	134	126	90	90	86	46	46	42	**0**	**12**	**11**
180	182	176	130	136	128	91	91	88	47	48	44	**12**	**0**	**9**
186	188	181	136	141	133	97	97	93	52	53	49	**11**	**9**	**0**

Following in Figure 2 is a 2-dimensional representation of the proximity matrix produced by the SMACOF algorithm.

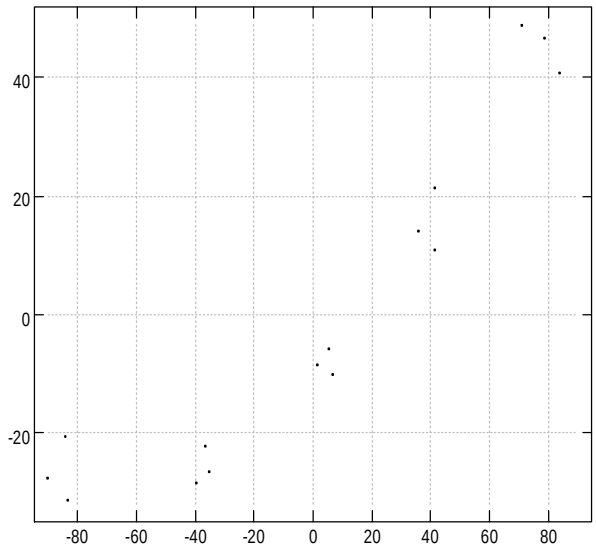

Figure 2 2-dimensional representation of proximity matrix 2 produced by the SMACOF algorithm

Visually there are 5 well separated clusters as is also apparent from the proximity matrix. The membership matrices are as in Table 5. Maximum values have been bolded.

Table 5 Membership matrices produced by the various clustering methods for proximity matrix 2. Membership values have been multiplied by 100, to reduce space. The symbol * represents the value 100

O	FRFP					NERFCM					Rouben					Winham				
1	1	3	**9**	2	5	*	0	0	0	0	1	1	**3**	2	1	1	1	2	1	**2**
2	1	3	**8**	2	5	*	0	0	0	0	1	1	**3**	2	1	1	1	2	1	**2**
3	1	2	**9**	2	5	*	0	0	0	0	1	1	**3**	2	1	1	1	2	1	**2**
4	2	5	5	2	**8**	0	0	0	0	*	1	1	**3**	2	1	1	1	2	1	**2**
5	2	5	4	2	**8**	0	0	0	0	*	1	1	**3**	2	1	1	1	2	1	**2**
6	2	5	4	2	**8**	0	0	0	0	*	1	1	**3**	2	1	1	1	2	1	**2**
7	2	**8**	2	4	4	0	*	0	0	0	1	1	**2**	2	1	**2**	1	2	2	1
8	2	**8**	2	5	4	0	*	0	0	0	1	1	**2**	2	1	**2**	1	2	2	1
9	2	**8**	2	4	5	0	*	0	0	0	1	1	**2**	2	1	**2**	1	2	2	2
1	5	4	1	**8**	2	0	0	0	*	0	2	7	2	**2**	1	2	**2**	1	2	1
1	5	4	1	**8**	2	0	0	0	*	0	2	7	2	**2**	1	2	**2**	1	2	1
1	5	5	2	**8**	2	0	0	0	*	0	2	7	2	**2**	1	2	**2**	1	2	1
1	**8**	2	1	5	2	0	0	*	0	0	**2**	6	2	2	1	2	**2**	1	2	1
1	**8**	2	1	5	2	0	0	*	0	0	**2**	7	2	2	1	2	**2**	1	2	1
1	**8**	3	1	6	2	0	0	*	0	0	**2**	7	2	2	1	2	**2**	1	2	1
1	1	3	**9**	2	5	*	0	0	0	0	1	1	**3**	2	1	1	1	2	1	**2**
1	1	3	**8**	2	5	*	0	0	0	0	1	1	**3**	2	1	1	1	2	1	**2**
1	1	2	**9**	2	5	*	0	0	0	0	1	1	**3**	2	1	1	1	2	1	**2**
1	2	5	5	2	**8**	0	0	0	0	*	1	1	**3**	2	1	1	1	2	1	**2**
2	2	5	4	2	**8**	0	0	0	0	*	1	1	**3**	2	1	1	1	2	1	**2**
2	2	5	4	2	**8**	0	0	0	0	*	1	1	**3**	2	1	1	1	2	1	**2**
2	2	**8**	2	4	4	0	*	0	0	0	1	1	**2**	2	1	2	1	2	**2**	1
2	2	**8**	2	5	4	0	*	0	0	0	1	1	**2**	2	1	2	1	2	**2**	1
2	2	**8**	2	4	5	0	*	0	0	0	1	1	**2**	2	1	2	1	2	**2**	2
2	5	4	1	**8**	2	0	0	0	*	0	2	7	2	**2**	1	2	**2**	1	2	1
2	5	4	1	**8**	2	0	0	0	*	0	2	7	2	**2**	1	2	**2**	1	2	1
2	5	5	2	**8**	2	0	0	0	*	0	2	7	2	**2**	1	2	**2**	1	2	1
2	**8**	2	1	5	2	0	0	*****	0	0	**2**	6	2	2	1	2	**2**	1	2	1
2	**8**	2	1	5	2	0	0	*****	0	0	**2**	7	2	2	1	2	**2**	1	2	1
3	**8**	**3**	1	6	2	0	0	*****	0	0	**2**	7	2	2	1	2	**2**	1	2	1

Rouben's and Windhams methods clearly fail in this case. NERFCM has produced a crisp clustering rather than a fuzzy clustering. The clustering quality index supports the fact that, visually, FRFP produces a membership matrix more in line with the visual demonstration that shows that objects belong to more than one cluster even though the degree is small except for one cluster.

Table 6 Average Clustering Quality and times for data set 2

Methods	Clustering Quality	Times
FRFP	12.08	0.0050
NERFCM	9.45	0.0125
Rouben	1.07	0.0200
Windham	1.05	0.145

Clearly FRPC is better both in terms of clustering quality and computational times.

8.3 Simulation 3 proximity matrix 3

The third proximity matrix is randomly generated such that the magnitude of the proximities falls approximately into 3 ranges such that 3 clusters are apparent in the proximity matrix as in Table 7.

Table 7 Proximity matrix 3

0	**4**	**4**	**5**	**5**	**5**	13	19	10	16	17	15	25	23	27	24	24	26
4	**0**	**6**	**8**	**1**	**1**	1	1	1	1	1	1	2	2	2	2	2	2

						2	7	3	9	2	6	4	7	1	7	7	6
4	**6**	**0**	**4**	**3**	**4**	15	14	12	16	17	13	22	25	22	25	27	24
5	**8**	**4**	**0**	**3**	**2**	11	16	13	14	17	13	26	24	27	24	27	27
5	**1**	**3**	**3**	**0**	**5**	13	17	16	13	15	14	26	21	26	24	20	25
5	**1**	**4**	**2**	**5**	**0**	16	15	12	16	12	12	23	27	28	26	24	21
13	12	15	11	13	16	**0**	**7**	**4**	**1**	**1**	**7**	12	15	13	15	14	17
19	17	14	16	17	15	**7**	**0**	**1**	**3**	**1**	**3**	11	10	12	11	15	15
10	13	12	13	16	12	**4**	**1**	**0**	**9**	**8**	**4**	11	15	12	11	17	16
16	19	16	14	13	16	**1**	**3**	**9**	**0**	**4**	**7**	13	15	18	17	14	15
17	12	17	17	15	12	**1**	**1**	**8**	**4**	**0**	**3**	15	18	13	15	12	12
15	16	13	13	14	12	**7**	**3**	**4**	**7**	**3**	**0**	15	15	17	17	10	10
25	24	22	26	26	23	12	11	11	13	15	15	**0**	**4**	**2**	**4**	**4**	**3**
23	27	25	24	21	27	15	10	15	15	18	15	**4**	**0**	**3**	**7**	**3**	**9**
27	21	22	27	26	28	13	12	12	18	13	17	**2**	**3**	**0**	**4**	**4**	**5**
24	27	25	24	24	26	15	11	11	17	15	17	**4**	**7**	**4**	**0**	**2**	**6**
24	27	27	27	20	24	14	15	17	14	12	10	**4**	**3**	**4**	**2**	**0**	**1**
26	26	24	27	25	21	17	15	16	15	12	10	**3**	**9**	**5**	**6**	**1**	**0**

There are 18 objects with 6 in each cluster according to the proximity matrix. Figure 3 is a plot of the objects obtained by the SMACOF algorithm.

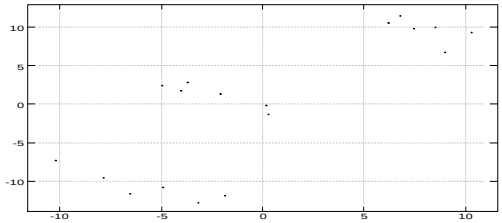

Figure 3 Plot of points corresponding to objects with similar inter object distances as in proximity matrix 3

Both the proximity data and the visual representation show 3 distinct clusters that we will number 1,2, and 3 from the bottom left corner to the top right corner. A fuzzy membership matrix should show the following properties to match the visual representation. Members of visual cluster 1 should show a very dominant membership in that cluster and a small membership in cluster 2 and a still smaller membership in cluster 3. The members of cluster 2 should show a strong membership in that cluster and small but approximately equal memberships in the other 2 clusters. Members of cluster 3 are like members of cluster 1.

Following, in Table 8, are the various membership matrices obtained by clustering using proximity matrix 3.

Table 8 Membership matrices for proximity matrix 3. Membership values have been multiplied by 100, to reduce space except in the case where it is equal to 1.

FRFP			NERFCM			Rouben			Windham		
19	**70**	11	**1**	0	0	28	**49**	23	34	**34**	32
15	**76**	9	**1**	0	0	28	**49**	24	34	**34**	32
18	**71**	11	**1**	0	0	28	**49**	23	34	**34**	32
18	**72**	10	**1**	0	0	28	**49**	23	34	**35**	31
16	**75**	9	**1**	0	0	28	**48**	24	34	**34**	32
16	**75**	9	**1**	0	0	28	**49**	23	34	**35**	31
70	15	15	0	0	**1**	34	**41**	26	33	33	**33**
71	13	16	0	0	**1**	36	**39**	25	33	33	**34**
64	18	18	0	0	**1**	33	**42**	25	33	33	**33**
70	15	15	0	0	**99**	35	**40**	26	33	33	**33**
71	14	15	0	0	**1**	34	**40**	26	33	33	**33**
62	18	19	0	0	**1**	34	**41**	25	33	33	**33**
19	10	**71**	0	**1**	0	**46**	30	24	33	32	**35**
19	11	**70**	0	**1**	0	**44**	32	24	33	32	**35**
18	10	**72**	0	**1**	0	**45**	31	24	33	32	**35**
19	10	**71**	0	**1**	0	**45**	31	24	33	32	**35**
16	9	**75**	0	**1**	0	**46**	30	24	33	32	**35**
17	10	**73**	0	**1**	0	**45**	31	24	33	32	**35**

The membership matrix for NERFCM is obviously a poor expression of visual clustering as are the ones for Rouben and Windham methods. This is supported by the clustering quality values in this case as shown in Table 9. If we examine Table 7 we see 3 sets of points. In one set, the points belong very strongly to one cluster and much less so , but equally so, to the other 2 clusters. In case of the other 2 sets the points belong very strongly to one cluster, a little to the other cluster, and hardly at all to the remaining cluster. This is borne out by the membership matrix for FRFDC.

Table 9 Clustering and cluster quality measures for proximity matrix 3

Methods	Clustering Quality	Times
FRFP	3.04	0.0000
NERFCM	3.81	0.0100
Rouben	1.08	0.0200
Windham	1.00	0.0100

.

FRFP produced a better membership matrices than NERFCM which produced a crisp partition that is not supported by a visual inspection of the proximity matrix.

8.4 Simulation 4

For this simulation the clustering was repeated 100 times for each of the 3 types (proximity matrices 1 to 3) of proximity matrices used in the previous simulations. The results are shown in Table 10 to Table 12.

Table 10 Averages of 100 trials using proximity matrix 1 each time

Methods	Clustering Quality	Times
FRFP	4.81	0.00180
NERFCM	4.94	0.00872
Rouben	1.73	0.00810
Windham	2.54	0.00651

.

Table 11 Averages of 100 trials using proximity matrices similar to proximity matrix 2

Methods	Clustering Quality	Times
FRFP	13.96	0.00342
NERFCM	11.14	0.00651
Rouben	1.08	0.01020
Windham	1.06	0.00661

Table 12 Averages of 100 trials using proximity matrices similar to proximity matrix 3 but with 72 objects each time.

Methods	Clustering Quality	Times
FRFP	2.01	0.01171
NERFCM	2.98	0.01644
Rouben	1.01	0.02105
Windham	1.00	0.01644

On average FRFP and NERFCM are about equivalent in terms of the quantitative measure of clustering quality.

8.5 Simulation 5

Following are execution times when the clustering methods are applied to proximity matrices generated the same way as proximity matrix 3 with different numbers of objects to get an idea of time complexity. The times are averages over 100 trials for each object set cardinality.

Table 13 Execution times for several trials for various sized proximity matrices of type 3

# objects	FRFP	NERFCM	Rouben	Windham
18	0.00370	0.01063	0.01562	0.01250
36	0.00510	0.01141	0.01582	0.01191
72	0.00411	0.01080	0.01471	0.01204
144	0.03863	0.05939	0.05374	0.06102
288	0.15090	0.25831	0.21871	0.25101

This again shows that FRFP is most efficient.

9 Summary and Conclusion

A method (FRFP) of relational clustering that is based on solving a function of the membership matrix for a fixed point is in several cases superior to NERFCM. FRFP produces membership matrices that are a better representation of the proximity matrices, that is the input to the automatic clustering, according to a visual inspection of the proximity matrix, the 2–dimensional representation of the proximity matrix and the membership matrix that is the output of the automatic clustering. A distinction has been made between clustering quality and cluster quality. NERFCM has a strong tendency to produce crisp clusters that may not accurately reflect the inter-object proximities. The quantitative clustering quality measure proposed in this paper also appears to favour crisp clusters to some degree and needs to be improved. As has been demonstrated here, ultimately a good way of evaluating the result of clustering is to produce a visual representation and compare it to the membership matrix that is produced.

In addition to providing improved clustering quality, FRFP is also more computationally efficient than the other algorithms. Future work may consist of using general knowledge about solving for fixed points to further improve the efficiency of the clustering algorithm.

Acknowledgements

The support of the Canadian Government in terms of an NSERC research grant (227338-04) and support of Thompson Rivers University in terms of use of facilities is gratefully acknowledged.

References

1. Jain, A.K. and R.C. Dubes, *Algorithms for Clustering Data*. 1988, Upper Saddle River, NJ: Prentice Hall. 321.
2. Anthony, A. and M. desJardins. *Open problems in relational data clustering*. in *ICML Workshop on Open Problems in Statistical Relational Learning*. 2006. Pittsburgh.
3. Hathaway, R.J., J.W. Davenport, and J.W. Bezdek, *Relational duals of the c-means clustering algorithms.* Pattern Recognition, 1989. **22**(2): p. 205-12.
4. Roubens, M., *Pattern classification problems and fuzzy sets.* Fuzzy Sets and Systems, 1978. **1**(4): p. 239-53.

5. Windham, M.P., *Numerical classification of proximity data with assignment measures.* Journal of Classification, 1985. **2**(1): p. 157-72.
6. Bezdek, J.C., R.J. Hathaway, and M.P. Windham, *Numerical comparison of the RFCM and AP Algorithms for clustering relational data.* Pattern Recognition, 1991. **24**(8): p. 783-91.
7. Hathaway, R.J. and J.C. Bezdek, *Nerf c-means: Non-Euclidean relational fuzzy clustering.* Pattern Recognition, 1994. **27**(3): p. 429-37.
8. Kaufman, L. and P.J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. 1990, New York: John Wiley and Sons.
9. Dave, R.N. and S. Sen, *Robust fuzzy clustering of relational data.* Fuzzy Systems, IEEE Transactions on, 2002. **10**(6): p. 713-27.
10. Bezdek, J., *Pattern Recognition with Fuzzy Objective Function Algorithms*. 1981: Plenum.
11. Nasraoui, O., R. Krishnapuram, and A. Joshi. *Relational Clustering Based on a New Robust Estimator with Application to Web Mining*. in *NAFIPS* 1999.
12. Krishnapuram, R., et al., *Low-Complexity Fuzzy Relational Clustering Algorithms for Web Mining.* IEEE Transactions on Fuzzy Systems, 2001. **9**(4): p. 595-607.
13. Sen, S. and R.N. Dave. *Agglomerative model for fuzzy relational clustering (FRC).* in *19th International Conference of the North American Fuzzy Information Processing Society*. 2000.
14. Cimino, M.G.C.A., B. Lazzerini, and F. Marcelloni. *A novel approach to robust fuzzy clustering of relational data*. in *NAFIPS '04 IEEE Annual Meeting of the Fuzzy Information Processing Society* 2004.
15. Hathaway, R.J., J.M. Huband, and J.C. Bezdek. *A Kernelized Non-Euclidean Relational Fuzzy c-Means Algorithm*. in *FUZZ '05 The 14th IEEE International Conference on Fuzzy Systems*. 2005.
16. Denoeux, T. and M.H. Masson, *EVCLUS: evidential clustering of proximity data.* IEEE Transactions onSystems, Man, and Cybernetics Part B, 2004. **34**(1): p. 95-109.
17. Borgelt, C. *Prototype-less Fuzzy Clustering*. in *FUZZ-IEEE 2007 IEEE International Fuzzy Systems Conference*. 2007.
18. Dunn, J.C., *A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters.* Journal of Cybernetics, 1973. **3**(3): p. 32-57.
19. Ruspini, E., *Optimization in sample descriptions: Data reduction and pattern recognition using fuzzy clustering.* IEEE Transactions on Systems, Man and Cybernetics, 1972. **2**: p. 541.
20. Ben-Dor, A., R. Shamir, and Z. Yakhini, *Clustering Gene Expression Patterns.* Journal of Computational Biology, 1999. **6**(3-4): p. 281-297.
21. Kwon, S.H., *Cluster validity index for fuzzy clustering.* Electronics Letters, 1998. **34**(22): p. 2176-7.
22. Dunn, J.C., *Indices of partition fuzziness and detection of clusters in large data sets*, in *Fuzzy automata and decision processes*, M.M. Gupta, Editor. 1976, Elsevier: New York.
23. Pal, N.K. and J.C. Bezdek, *On clustering validity for the fuzzy c-means model.* IEEE Transactions on Fuzzy Systems, 1995. **3**(3): p. 370-9.
24. Fukuyama, Y. and M. Sugeno. *A new method of choosing the number of clusters for the fuzzy c-means method*. in *5th Fuzzy Systems Symposium*. 1989.
25. Xie, X.L. and G. Beni, *A validity measure for fuzzy clustering.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1991. **13**(8): p. 841-7.
26. Bezdek, J.C. and N.R. Pal. *Cluster validation with generalized Dunn's indices*. in *Artificial Neural Networks and Expert Systems, 1995. Proceedings., Second New Zealand International Two-Stream Conference on*. 1995.
27. Bezdek, J.C. and N.R. Pal, *Some new indexes of cluster validity.* Systems, Man and Cybernetics, Part B, IEEE Transactions on, 1998. **28**(3): p. 301-315.
28. Maulik, U. and S. Bandyopadhyay, *Performance evaluation of some clustering algorithms and validity indices.* Transactions on Pattern Analysis and Machine Intelligence, 2002. **24**(12): p. 1650-1654.
29. Stein, B., S.M. Eissen, and F. Wissbrock. *On cluster validity and the information need of users.* in *3rd IASTED International Conference on Artificial Intelligence and Applications*. 2003. Benalmadena, Spain.
30. Hubert, L. and P. Arabie, *Comparing partitions.* Journal of Classification, 1985. **2**: p. 193-8.
31. Bezdek, J.C., *Cluster validity with fuzzy sets.* Journal of Cybernetics, 1974: p. 58-73.
32. Dave, R.N., *Validating fuzzy partitions obtained through c-shells clustering.* Pattern Recognition Letters, 1996. **17**(6): p. 613-623.
33. Borg, I. and P. Groenen, *Modern Multidimensional Scaling Theory and Applications*. 1997: Springer.
34. Hair, J.F., et al., *Multivariate data analysis.* 1998, Upper Sadle River: Prentice Hall.

35. Cox, T.F. and M.A.A. Cox, *Multidimensional Scaling*. 1994: Chapman and Hall.
36. Kruskal, J.B. and M. Wish, *Multidimensional Scaling*. Quantitative Applications in the Social Sciences 07-001, ed. J.L. Sullivan and R.G. Niemi. 1989: Sage Publications.
37. Young, F.W. and R.M. Hamer, *Theory and Applications of Multidimensional Scaling*. 1994, Hillsdale, NJ. : Eribaum Associates.
38. Kruskal, J.B. and M. Wish, *Multidimensional Scaling*. 1977, Beverly Hills. CA. : Sage Publications.
39. Young, F.W., *Research Methods for Multimode Data Analvsis in the Behavioral Sciences*, ed. H.G. Law, et al. 1984.
40. Stoop, L. and J. de Leeuw, *How to Use SMACOF-IB (A complete user's guide.)*, in *Department of Data Theory. University of Leiden. The Netherlands* 1982.
41. Deleeuw, J., in *Recent Developments in Statistics*, R. Barra, Editor. 1977: Amsterdam.
42. Deleeuw, J. and W.J. Heisernn, *Geometrical Representatioms of Relational Data*, J.C. Lingoes, Editor. 1977, Mathesis Press: Ann Arbor, M.
43. Faloutsos, C. and K.-I. Lin. *FastMap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets.* in *ACM SIGMOD '95 Conference on Management of Data*. 1995. San Jose, CA.