# Visualization of Fuzzy Clustering Results by Modified Sammon Mapping

**Annamária Kovács and János Abonyi**

Department of Process Engineering, University of Veszprém, Veszprém, Hungary, P.O.Box 158, H-8200, www.fmt.vein.hu/softcomp, abonyij@fmt.vein.hu

*Abstract:*
*In many clustering problems high-dimensional data are involved. Hence, the resulting clusters are high-dimensional geometrical objects which are difficult to analyze and interpret. Cluster validity measures try to solve this problem, but they reduce the information into a single value. As the low dimensional graphical representation of the clusters could be much more informative than such a single number, this paper proposes a new tool for the visualization of fuzzy clustering results. The modified Sammon mapping is based on the basic properties of fuzzy clustering algorithms and maps the cluster centers and the data such that the distances between the clusters and the data-points will be preserved.*

*Keywords:Fuzzy clustering, Sammon projection, visualization*

## 1 Introduction

In our society the amount of data doubles almost every year. Hence, there is an urgent need for a new generation of computational techniques and tools to assist humans in extracting useful information (knowledge) from the rapidly growing volumes of data. Among the wide range of data-mining tools, the clustering-based computational intelligence methods are becoming increasingly popular, as they are able to learn the mapping of functions and systems, and can perform classification from labeled training data as well as explore structures and classes in unlabeled data.

Clustering algorithms always fit the clusters to the data, even if the cluster structure is not adequate for the problem. To analyze the adequateness of the cluster prototypes, cluster validity measures can be used to evaluate a single cluster or the whole partition of the data. However since validity measures reduce the overall evaluation to a single number, they cannot avoid a certain loss of information. Hence, the impact of visualization of fuzzy clustering results has been already realized in [1], when the membership values were simply projected

into the input variables of the model, and the resulted plots can serve for the same purpose as validity measures, but they are more informative than the simple numbers produced by validity measures.

To give more insight into the high-dimensional structures of the fuzzy clusters, in this paper we suggest using advanced pattern recognition algorithms developed for the visualization of high-dimensional data. These feature extraction and dimensionality reduction algorithms map the original features (variables) into fewer features which preserve the main information of the data structure. These tools are able to convert complex, nonlinear statistical relationships between the high-dimensional data items into simple geometric relationships on a low-dimensional display and compress information while preserving the most important topological and metric relationships of the primary data items. Nowadays Multi-dimensional Scaling means any method searching for a low (in particular two) dimensional representation of multi-dimensional data sets [2]. Sammon's non-linear mapping is a multi-dimensional scaling method [3]. It is a well-known procedure for mapping data from a high-dimensional space onto a lower-dimensional space by preserving the inter-pattern distances. This is achieved by minimizing an error criterion, called Sammon's stress, which penalizes differences in distances between points in the original space and the mapped space.

Fuzzy c-means cluster analysis has been already combined with this non-linear mapping method and successfully applied to map the distribution of pollutants and to trace their sources to access potentional environmental hazard on a soil database from Austria [4]. As Sammon mapping attempts to preserve the structure of high ($n$)-dimensional data by finding $N$ points in a much lower ($q$)-dimensional data space, such the interpoint distances measured in the $q$ dimensional space approximate the corresponding interpoint distances in the $n$ dimensional space, the algorithm involves a large number of computations as in every iteration step it requires the computation of $N \cdot (N-1)/2$ distances. Hence, the application of Sammon mapping becomes impractical for large $N$.

To avoid this problem in this paper we have modify the algorithm of Sammon mapping. By using the basic properties of fuzzy clustering algorithms the proposed tool maps the cluster centers and the data such that the distances between the clusters and the data-points will be preserved. During the iterative mapping process, the algorithm uses the membership values of the data and minimizes the objective function of the original clustering algorithm.

In the following, in Section 2, the general algorithm of fuzzy clustering is described. The proposed visualization tool will be described in Section 3. In Section 4, the proposed tool is applied to two data sets: classification of wines and iris flower types. The results show superior performance over the linear method (Principal Component Analysis) and the classical Sammon projection tools.

# 2. Fuzzy Clustering

## 2.1 Clustering Algorithm

The aim of cluster analysis is the classification of objects according to similarities among them, and organizing data into groups. A cluster is a group of objects that are more similar to other ones than to other clusters. In metric spaces, similarity is often defined by means of distance based upon the length from a data vector to some prototypical object of the cluster. The prototypes are usually not known beforehand, and are sought by the clustering algorithm simultaneously with the partitioning of the data. Therefore, clustering techniques are among the unsupervised (learning) methods, since they do not use *a prior* class identifiers. The prototypes may be vectors (centers) of the same dimension as the data objects, but they can also be defined as "higher-level" geometrical objects, such as linear or non-linear subspaces or functions.

Since clusters can formally be seen as subsets of the data set, one possible classification method can be according to whether the subsets are fuzzy or crisp (hard). Hard clustering methods are based on classical set theory, and it requires an object that either does or does not belong to a cluster. Fuzzy clustering methods (FCM) allow objects to belong several clusters simultaneously with different degrees of membership [5]. The data set, $\mathbf{X}$, is thus partitioned into $c$ fuzzy subsets. In many real situations, fuzzy clustering is more natural than hard clustering, as objects on the boundaries between several classes are not forced to fully belong to one of the classes. However, they rather are assigned to membership degrees between 0 and 1 indicating their partial memberships.

In this paper, the clustering of quantitative data is considered. The data are typically observations of some physical phenomenon. Each observation consists of $n$ measured variables, grouped into an $n$-dimensional column vector $\mathbf{x}_k = [x_{k1},...,x_{kn}]^T, \mathbf{x}_k \in \Re^n$. A set of $N$ observations is denoted by $\mathbf{X} = \{\mathbf{x}_k = 1,2,...,N\}$ and represented as a $n \times N$ matrix:

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ x_{N1} & x_{N2} & \dots & x_{Nn} \end{bmatrix} \tag{1}$$

In the pattern recognition terminology, the columns of ***x*** called patterns or objects, the rows are called the features or attributes, and **X** is called the pattern matrix. The objective of clustering is to divide the data set **X** into *c* clusters.

A $c \times N$ matrix $\mathbf{U} = [\mu_{ik}]$ represents the fuzzy partitions, where $c$ is the number of the fuzzy clusters and $\mu_{ik}$ denotes the degree of the membership of the $\mathbf{x}_k$ -th observation belongs to the $1 \le i \le c$ -th cluster.

The objective of the FCM model [10] is to minimize the sum of the weighted squared distances between the data points, $\mathbf{x}_k$ and the cluster centers, $\mathbf{v}_i$. The distances $d^2(i,k)$ are weighted with the membership values $\mu_{ik}$. Therefore, the objective function is

$$J(X,U,V) = \sum_{i=1}^{c}\sum_{k=1}^{N}(\mu_{ik})^m d^2(i,k) \tag{5}$$

where $\mathbf{U} = [\mu_{ik}]$ is a fuzzy partition matrix of X,

$\mathbf{V} = [\mathbf{v_1}, \mathbf{v_2}, ..., \mathbf{v_c}]$ is a vector of cluster prototypes (centers),

$m \in \langle 1, \infty)$ is a weighting exponent that determines the fuzziness of the resulting clusters and it is often chosen as *m=2*.

$d^2(i,k)$ can be determined by any appropriate norm, e.g., an **A**-norm:

$$d^2(i,k) = \|\mathbf{x}_k - \mathbf{v}_i\|_A = \sqrt{(\mathbf{x}_k - \mathbf{v}_i)^T \mathbf{A} \ (\mathbf{x}_k - \mathbf{v}_i)} \tag{6}$$

The minimization of the c-means functional (Eq. 5) represents a non-linear optimization problem that can be solved by using a variety of available methods [5]. The most popular method, however, is the alternating optimization (AO), known as the fuzzy c-means algorithm (FCM-AO).

Using points, as prototypes in the FCM, result in spherical clusters (corresponding to the A-norm). Different cluster shapes can be obtained with different norms as suggested in the Gustavson-Kessel algorithm, or with different kinds of prototypes, e.g., linear varieties (FCV), where the clusters are linear subspaces of the feature space. An *r*-dimensional linear variety is defined by the vector $\mathbf{v}_i$ and the directions $\mathbf{s}_j$, $j = 1,...,r$. In this case, the distance between the data $\mathbf{x}_k$ and the *i*th cluster is:

$$d^2(i,k) = \sqrt{\|\mathbf{x}_k - \mathbf{v}_i\|^2 - \sum_{j=1}^{r}\left((\mathbf{x}_k - \mathbf{v}_i)^T \mathbf{A} \ \mathbf{s}_{ij}\right)^2} \tag{7}$$

The corresponding fuzzy c-varieties alternating optimization (FCV-AO) brings up to determine the centers $\mathbf{v}_i$ in step 1. (see the Appendix), and it computes the directions $\mathbf{s}_{ij}$ as the unit eigenvectors of the *r* largest eigenvalues of the fuzzy scatter matrix:

$$\mathbf{S}_{iA} = A^{1/2}\left[\sum_{k=1}^{N} \mu_{ik}\left(\mathbf{x}_k - \mathbf{v}_i\right)\left(\mathbf{x}_k - \mathbf{v}_i\right)^T\right]\mathbf{A}^{1/2} \tag{8}$$

If *r=1,* this results in fuzzy c-lines (FCL) and FCL-AO algorithm.

The description of the clustering algorithm is given in Appendix.

## 2.2 Validity Measures

Cluster validity refers to the problem whether a given fuzzy partition fits to the data all. The clustering algorithm always tries to find the best fit for a fixed number of clusters and the parameterized cluster shapes. However this does not mean that even the best fit is meaningful at all. Either the number of clusters might be wrong or the cluster shapes might not correspond to the groups in the data, if the data can be grouped in a meaningful way at all. Cluster validity measures are used to validate a clustering result in general or also in order to determine the number of clusters [4]. Let us review two cluster validity measures. The partition coefficient (*F*) is defined in the following:

$$F = \frac{\sum_{i=1}^{c}\sum_{j=1}^{N}\mu_{ij}^{m}}{N} \tag{9}$$

The higher the value of the partition coefficient, the better the clustering result. The highest value of *F* 1 is obtained when the fuzzy partition is actually crisp, i.e. $\mu_{ij} \in \{0,1\}$. The lowest value 1/c is reached when all data are assigned to all clusters with the same membership degree 1/c. This means that fuzzy clustering result is considered better when it is more crisp.

The partition entropy (*H*) is defined in the following:

$$H = \frac{\sum_{i=1}^{c}\sum_{j=1}^{N}\mu_{ij}\ln(\mu_{ij})}{n} \tag{10}$$

The smaller the value of the partition entropy, the better the clustering result. This means that similar to *F* crisper fuzzy partitions are considered better.

# 3. Sammon Maping based Fuzzy Custer Visualization

## 3.1 Intoduction to Sammon Mapping

Sammon mapping is a feature extraction algorithm that is widely used for pattern recognition and exploratory data analysis. This tool is a simple yet very useful nonlinear projection algorithm maps the original features (measurements) into fewer variables by preserving the inherent structure of the data. While PCA attempts to preserve the variance of the data, Sammon's Mapping tries to preserve the interpattern distances. That is to preserve the structure of high (*n*)-dimensional data by finding *N* points in a much lower (*q*)-dimensional data space, such the interpoint distances measured in the *q* dimensional space approximate the corresponding interpoint distances in the *n* dimensional space.

Suppose the following: $\mathbf{X} = \left\{ \mathbf{x}_k \mid \mathbf{x}_k = (x_{k1}, x_{k2}, ..., x_{kn})^T, k = 1,2,..., N \right\}$ is the set of *n* input vectors, $\mathbf{Y} = \left\{ \mathbf{y}_k \mid \mathbf{y}_k = (y_{k1}, y_{k2}, ..., y_{kq})^T, k = 1,2,..., N \right\}$ is the unknown vectors to be found, $d_{ij} = d(x_i, x_j)$, $x_i, x_j \in \mathbf{X}$ and $d_{ij}^* = d(y_i, y_j)$, $y_i, y_j \in \mathbf{Y}$ where $d(x_i, x_j)$ is the Euclidian distance between $x_i$ and $x_j$.

The Sammon mapping is looking for **Y** by minimizing the error function *E*:

$$E = \frac{1}{\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} d(i,j)} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \frac{\left( d(i,j) - d^*(i,j) \right)^2}{d(i,j)} \tag{11}$$

Minimization of *E* is an optimization problem in the *nq* variables $y_{ij}$ (*i*=1,2,..,*N*; *j*=1,2,...,*q*). Sammon applied the method of steepest decent to minimizing this function. Let $y_i(t)$ to be the estimate of $y_i$ at the *t*th iteration, $\forall i$. Then $y_i(t+1)$ is given by

$$y_{ij}(t+1) = y_{ij}(t) - \alpha \left[ \frac{\frac{\partial E(t)}{\partial y_{ij}(t)}}{\frac{\partial^2 E(t)}{\partial y_{ij}(t)^2}} \right] \tag{12}$$

where $\alpha$ is a nonnegative scalar constant (recommended $\alpha \approx 0.3$ *or* $0.4$), this is the step size for gradient search.

Now

$$\frac{\partial E(t)}{\partial y_{ij}(t)} = -\frac{2}{\lambda} \sum_{k=1,k\neq i}^{N} \left[ \frac{(d_{ik} - d_{ik}^*)}{(d_{ik} d_{ik}^*)} \right] (y_{ij} - y_{kj}) \tag{13}$$

and

$$\frac{\partial^2 E(t)}{\partial y_{ij}(t)^2} = -\frac{2}{\lambda} \sum_{k=1,k\neq i}^{N} \left[ \frac{1}{d_{ik} d_{ik}^*} \right] \left[ \left(d_{ik} - d_{ik}^*\right) - \left( \frac{\left(y_{ik} - y_{kj}\right)^2}{d_{ik}^*} \right) \left( 1 + \frac{(d_{ik} - d_{ik}^*)}{d_{ik}} \right) \right] \tag{14}$$

where $\lambda = \sum_{i<j} d_{ij}$ . It is not necessary to maintain $\lambda$ in (13) and (14) for a successful solution of the optimization problem, since the minimization of $\left( 1/\sum_{i<j} d_{ij} \right) \sum_{i<j} \left( \left(d_{ij} - d_{ij}^*\right)^2 / d_{ij} \right)$ and $\sum_{i<j} \left( \left(d_{ij} - d_{ij}^*\right)^2 / d_{ij} \right)$ yield the same solution.

When the gradient-descent method is applied to search for the minimum of Sammon's stress, a local minimum in the error surface could be reached. Therefore a significant number of experiments with different random initializations may be necessary. Nevertheless the initialization could be based on information which is obtained from the data, such as the first and second norms of the feature vectors or the principal axes of the covariance matrix of the data.

## 2.2 Modified Sammon Mapping

A disadvantage of the original Sammon mapping is that when a new data point has to be mapped, the whole mapping procedure has to be repeated [6]. It means computational load, because in each iteration $N \cdot (N-1)/2$ distances as well as the error derivatives, must be calculated where *N* represents the number of data points. Hence, the application of Sammon mapping becomes impractical for large *N*. To avoid this problem in this section we have modify the previously presented algorithm of Sammon mapping. By using the basic properties of fuzzy clustering algorithms where only the distance between the data points and the cluster centers are considered to be important, with the modified algorithm only $N \cdot c$ distances are calculated in every iteration, where *c* represents the number of clusters, so the cost function is:

$$E = \sum_{i=1}^{c} \sum_{k=1}^{N} \mu_{i,k} \left( d(i,k) - d^*(i,k) \right)^2 \tag{15}$$

In every iteration, after the adaptation of the projected data points, the projected cluster centers are calculated based on the weighted mean formula of the fuzzy clustering algorithms:

$$\mathbf{z}_i = \sum_{k=1}^{N} \mu_{ik} \mathbf{y}_k \Big/ \sum_{k=1}^{N} \mu_{ik} \tag{16}$$

As in the low dimensional space the distances are measured by the Eucledian norm, $d^*(i,k) = \sqrt{(\mathbf{y}_k - \mathbf{z}_i)^T (\mathbf{y}_k - \mathbf{z}_i)}$, and the dimension of the output map is $q=2$, the result of the original clustering algorithms can be easily analyzed.

Based on these mapped distances, the membership values of the projected data can be also evaluated

$$\mu_{ik}^* = \frac{1}{\sum_{j=1}^{c} \left( \frac{d^*(i,k)}{d^*(j,k)} \right)^{\frac{2}{m-1}}} \tag{17}$$

The quality of the mapping can be easily evaluated based on the mean square error of the original and the re-calculated membership values.

The pseudo-code of the proposed algorithm is given in Table I.

# 3. Application Example

In order to examine the performance of the proposed visualization method two examples are presented in this section. The first example is the visualization of the results of the clustering of the well known Iris data, while the second one deals with the analysis of the Wine data, coming from the UCI Repository of Machine Learning Databases (http://www.ics.uci.edu). These studies are performed to evaluate the performance of the proposed method, the e.g., the mean square error of the re-calculated membership values $\left\| \mathbf{U} - \mathbf{U}^* \right\|$, the difference between the original and the re-calculated cluster validity measures (see Eq.(9)), and the Sammon stress coefficient (11). For comparison, the data and the cluster centers were projected by principal component analysis (PCA) and standard Sammon projection. The results are summarized in Table II and III and show that the proposed tool has superior performance over the linear method and the classical Sammon projection tools.

**Table I.** The proposed FUZZSAMMVIS algorithm

Input *n,q* and $\mathbf{X} = \{\mathbf{x}_k \in R^n : k = 1,2,...N\}$; $\varepsilon > 0$; maxstep;

Clustering the data by Fuzzy c-means to obtain the membership values $\mathbf{U} = [\mu_{ik}]$ and the cluster centers, **V**

Generate randomly $\mathbf{Y} = \{\mathbf{y}_k \in R^q : k = 1,2,...N\}$; and calculate the projected cluster centers by $\mathbf{z}_i = \sum_{k=1}^{N} \mu_{ik} \mathbf{y}_k \Big/ \sum_{k=1}^{N} \mu_{ik}$

Compute $D = [d_{ij} = d(\mathbf{x}_i, \mathbf{v}_j)]_{N \times N}$ and $D^* = [d_{ij}^* = d(y_i, \mathbf{z}_j)]_{N \times N}$;

error=High value; t=1;

while((error> $\varepsilon$ ) and ( $t \leq$ maxstep)

{for $(i = 1 : i \leq c; i++)$

{for $(j = 1 : j \leq N; j++)$

{Compute $\frac{\partial E(t)}{\partial y_{ij}(t)}$ using (13) and $\frac{\partial^2 E(t)}{\partial y_{ij}(t)^2}$ using (14);

$$y_{ij}(t+1) = y_{ij}(t) - \alpha \left[ \frac{\frac{\partial E(t)}{\partial y_{ij}(t)}}{\frac{\partial^2 E(t)}{\partial y_{ij}(t)^2}} \right];$$

Compute $\mathbf{z}_i = \sum_{k=1}^{N} \mu_{ik} \mathbf{y}_k \Big/ \sum_{k=1}^{N} \mu_{ik}$

}

}

Compute $D^* = [d_{ij}^* = d(\mathbf{y}_i, \mathbf{z}_j)]_{N \times N}$

}

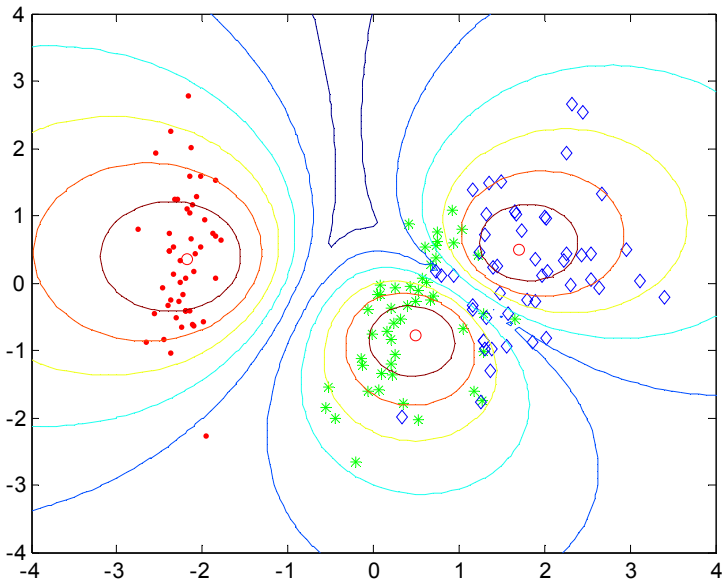


Figure 1. Projection of the results of the clustering of Iris data

**Table II.** Results of the mapping of the Iris clustering results

| | $\|\mathbf{U} - \mathbf{U}^*\|$ | F | F* | Sammon str. ($E$) |
|---|---|---|---|---|
| PCA | 0.0184 | 0.7052 | 0.7445 | 0.0098 |
| Sammon | 0.0128 | 0.7052 | 0.7272 | 0.0063 |
| FUZZSAMMVIS | **0.0030** | 0.7052 | 0.7076 | 0.0105 |

**Table III.** Results of the mapping of Wine clustering results

| | $\|\mathbf{U} - \mathbf{U}^*\|$ | F | F* | Sammon str. ($E$) |
|---|---|---|---|---|
| PCA | 0.1357 | 0.4761 | 0.7170 | 0.1468 |
| Sammon | 0.0622 | 0.4761 | 0.5650 | 0.0647 |
| FUZZSAMMVIS | **0.0427** | 0.4761 | 0.5137 | 0.1007 |

## Conclusions

By using the basic properties of fuzzy clustering algorithms in this paper a new tool has been proposed that maps the cluster centers and the data such that the distances between the clusters and the data-points will be preserved. During the iterative mapping process, the algorithm uses the membership values of the data and minimizes the objective function of the original clustering algorithm. Comparing to the original Sammon mapping not only reliable cluster shapes are obtained but the numerical complexity of the algorithm is also drastically reduced. The proposed tool is applied on different data sets: classification of wines, iris flower types. The results show superior performance over the linear method (Principal Component Analysis) and the classical Sammon projection tools.

## References

[1] Frank Klawonn: Visual Inspection of Fuzzy Clustering Results, Department of Computer Science, University of Applied Sciences Braunschweig/Wolfenbuettel, Germany

[2] J. Mao and A. K. Jain: Artificial neural networks for feature extraction and multivariate data projection, IEEE Trans. Neural Networks 629-637 (1995)

[3] Dick de Ridder, Robert P. W. Duin: Sammon's mapping using neural networks: A comparison, Pattern Recognition Letters 18. 1307-1316 (1997)

[4] M Hanesch, R. Scholger and M. J. Dekkers: The application of Fuzzy c-means cluster analysis and non-linear mapping to a soil data set for the detection of polluted sites, Phys. Chem. Earth 26. 885-891 (2001)

[5] J.C. Bezdek. Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, New York, (1981)

[6] N.R. Pal and V.K. Eluri, Two Efficient Connectionist Schemes for Structure Preserving Dimensionality Reduction, IEEE Transactions on Neural Networks, 9, 1143-1153 (1998)

## Acknowledgements

The authors would like to acknowledge the support of the Hungarian Ministry of Education (FKFP-0073/2001) and OTKA (Hungarian National Research Foundation), No. T037600. Janos Abonyi is grateful for the financial support of the Janos Bolyai Research Fellowship of the and Hungarian Academy of Sciences.

## Appendix: Clustering Algorithm

Initialization:

Given the data set **X**, choose the number of clusters *c*, the weighting exponent *m,* the termination tolerance $\varepsilon > 0$ and initialize the partition matrix randomly.

Repeat for $l = 1,2,...$

Step 1.: Compute the cluster centers:

$$v_i^{(l)} = \frac{\sum_{k=1}^{N} \left(\mu_{ik}^{(l-1)}\right)^m \mathbf{x}_k}{\sum_{k=1}^{N} \left(\mu_{ik}^{(l-1)}\right)^m}, \quad 1 \le i \le c$$

Step 2.: Compute the distances: $d^2(i,k) = \left\| \mathbf{x}_k - \mathbf{v}_i \right\|_A^2$

Step 3.: Update the partition matrix:

$$\text{If } d(i,k) > 0 \text{ for } \quad 1 \le i \le c, \ 1 \le k \le N,$$

$$\mu_{ik}^{(l)} = \frac{1}{\sum_{j=1}^{c} \left( \frac{d(i,k)}{d(j,k)} \right)^{\frac{2}{m-1}}} \quad \text{otherwise } \mu_{ik}^{(l)} = 0$$

until $\left\| \mathbf{U}^{(l)} - \mathbf{U}^{(l-1)} \right\| < \varepsilon$