16th World Congress of the International Fuzzy Systems Association (IFSA)
9th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT)

On the selection of m for Fuzzy c-Means

Vicenç Torra[1]

[1]University of Skövde, Sweden

Abstract

Fuzzy c-means is a well known fuzzy clustering algorithm. It is an unsupervised clustering algorithm that permits us to build a fuzzy partition from data. The algorithm depends on a parameter m which corresponds to the degree of fuzziness of the solution. Large values of m will blur the classes and all elements tend to belong to all clusters. The solutions of the optimization problem depend on the parameter m. That is, different selections of m will typically lead to different partitions.

In this paper we study and compare the effect of the selection of m obtained from the fuzzy c-means.

Keywords: Fuzzy clustering, Fuzzy c-means, parameters of FCM, m.

1. Introduction

Fuzzy clustering has been extensively used to extract knowledge from data. Fuzzy c-means [1] (see also e.g. [3, 4]) is one of the most used fuzzy clustering algorithms. It is a fuzzy generalization of k-means in which membership of elements to clusters is fuzzy. That is, the elements can belong to more than one cluster at the same time.

Fuzzy c-means is not the only fuzzy clustering algorithm. There are several families [5, 7, 2, 8] (e.g. entropy fuzzy c-means and possibilistic clustering) and variants (e.g. variable-size fuzzy c-means [6, 11]).

Fuzzy c-means depends on two parameters. One is the number of clusters. The other is the parameter m which stands for the degree of fuzziness in the solution. When m is close to one, the solution of the fuzzy c-means algorithm is similar to the one of k-means. Elements are basically assigned to only the nearest cluster and membership to others clusters is negligible. On the contrary, when m is large, fuzziness is also large and clusters are blurred. Elements tend to belong to all clusters with the same membership. Because of that, large values of m are not used. Typical values for the parameters m are between 1 and 2.

There are problems in which the membership degree of an element to the class is of relevance. This is the case of classification problems. In a multiclass problem we can e.g. select the k highest membership degrees. The membership values are used for this selection. In [10], we considered the problem of assigning elements to clusters according to a probability distribution based on the membership degrees. This application was in data privacy, and the assignment was to hide memberships and introduce some uncertainty to any intruder. That is, there is no certainty that elements are assigned to the nearest clusters.

When membership is of relevance, the selection of a value m is very important. Different values will lead to different memberships. At the same time, clusters will be shaped by the value selected. In addition, we can also prove that large values of m lead to solutions (partitions) that are not of interest.

In this work we study the effect of the parameter m in fuzzy c-means. We are interested on two types of effects. On the one hand, we are interested in knowing the general effect of the parameter m in the solution. Not only on the membership degrees of the data, but also on the effect on the centroids found. On the other hand, we want to know in what extent given a solution of fuzzy c-means for a given m, if we use later a different m' for computing membership degrees, this implies an important difference when comparing the respective objective functions. Expressing as in the first type of problem, this is to consider whether the approximate solution for m is acceptable for m'.

We do these analyses empirically, comparing the results of fuzzy c-means for a few data files using different values of c and m (the two parameters of fuzzy c-means). The comparison takes into account the difficulties of comparing fuzzy c-means results. It is well known that the practical application of these clustering algorithms poses an important problem. Fuzzy c-means algorithms typically lead to solutions that are a local optima of the optimization problem. In addition, several executions of the algorithm lead usually to different local optima. Because of that, when comparing different results we need to be sure that we are near enough to the global optima. We have taken this into account in our experiments.

The structure of this paper is as follows. In Section 2 we review the fuzzy c-means algorithm. In Section 3 we describe the experiments considered. The paper finishes with some conclusions and lines for future research.

2. Fuzzy c-means

Fuzzy clustering algorithms permit us to extract structures from data in which overlapping between clusters is permitted. Fuzzy c-means belongs to the family of algorithms that build fuzzy partitions. A

fuzzy partition is defined as follows.

Definition 1 *Let X be a reference set. Then, a set of membership functions $\mathcal{M} = \{\mu_1, \ldots, \mu_c\}$ is a fuzzy partition of X if for all $x \in X$ it holds*

$$\sum_{i=1}^{c} \mu_i(x) = 1$$

We give now the formalization of fuzzy c-means as well as the typical algorithm to solve this optimization problem. We will consider the following notation. We have a set of objects $X = \{x_1, \ldots, x_n\}$ and we want to build c clusters from these data. The value c is one of the parameters of the algorithm and should be given by the user. Then, the method builds the clusters which are represented by membership functions μ_{ik}, where μ_{ik} is the membership of the kth object (x_k) to the ith cluster.

Besides of c, fuzzy c-means requires another parameter m. This value should be $m > 1$. The larger the m, the larger the fuzziness in the clusters. With values near to 1, solutions tend to be crisp. With large values of m, membership values tend to be all $\mu_{ik} = 1/c$.

FCM formulates the construction of μ from X as the solution of a minimization problem. This problem is formulated as follows using v_i to represent the cluster center, or centroid, of the i-th cluster.

$$\text{Minimize} J_{FCM}(\mu, V) = \{\sum_{i=1}^{c}\sum_{k=1}^{n}(\mu_{ik})^m ||x_k - v_i||^2\} \quad (1)$$

subject to the constraints $\mu_{ik} \in [0,1]$ and $\sum_{i=1}^{c} \mu_{ik} = 1$ for all k.

This optimization problem is usually solved by means of the iterative process described in Algorithm 1. In short, the algorithm interleaves two steps. One in which the membership values are estimated assuming known centroides, and the other in which the centroides are estimated assuming that memberships are known. The algorithm converges to a local optimal solution of the optimization problem formulated above.

We cannot ensure that this algorithm leads to a global optimum. Convergence is ensured because at each step the value of the objective function is reduced. Nevertheless, we can converge to a local optimum.

When we are interested in studying and comparing the solutions of FCM, this usually causes difficulties. More specifically, different executions will typically lead to different solutions, and their comparison will show divergences that may only be due to the convergence to different values in the space of solutions. We discussed this problem in [9]. In order to reduce the problems caused by computation, we will execute the FCM several times and among all solutions we will select the one with a lower objective function.

Algorithm 1 Fuzzy c-means

Step 1: Generate initial V

Step 2: Solve $min_{\mu \in M} J(\mu, V)$ computing:

$$\mu_{ik} = \Big(\sum_{j=1}^{c}\Big(\frac{||x_k - v_i||^2}{||x_k - v_j||^2}\Big)^{\frac{1}{m-1}}\Big)^{-1}$$

Step 3: Solve $min_V J(\mu, V)$ computing:

$$v_i = \frac{\sum_{k=1}^{n}(\mu_{ik})^m x_k}{\sum_{k=1}^{n}(\mu_{ik})^m}$$

Step 4: If the stop condition does not hold, go to step 2; otherwise, stop

Let

$$< OF, v >= FCM(X, m, c)$$

represent the application of FCM to data X with parameters m and c. We have that in general, t applications of this algorithm may lead to t different sets of cluster centers and t different values of the objective function. Let $FCM_s(X, m, c)$ represent the sth application of this algorithm. Then, we use

$$< OF, v >= argmin_{s \in S} FCM_s(X, m, c)$$

with minimum over all objective functions, to represent the selection of the best solution. Here, the best solution is the one with a lower objective function.

For a large enough set S, the probability of finding the global optimum can be large enough. As we detail later, we have been using sets S of between 20 and 2000 executions.

As stated above, fuzzy c-means uses the parameter m to settle the degree of fuzziness. This value should be $m > 1$. It is known that the larger the m, the larger the fuzziness in the clusters. More specifically, with values of m near to 1, solutions tend to be crisp and when m increases largely, membership values tend to be all $\mu_{ik} = 1/c$. The following toy example illustrates this fact.

Example 1 *Let us consider the data in Table 1. We consider their clustering with $c = 4$ and a few different values of m, and also the outcome of the objective function when we consider the following cases:*

- *Crisp partitions with cluster centers in positions $(1,1)$, $(1,5)$, $(5,1)$, and $(5,5)$. We say that there are four clouds (in order to avoid the use of the term cluster) around these four centers. Partitions are defined according to the nearest center. So, we obtain four distinctive clusters. This solution is what we would expect from k-means clustering.*

- *Fuzzy membership with value* $1/c$ *to all clusters and all cluster centers in the center of the data (i.e., formally, a single cluster with its center in position* $(3,3)$*). The objective function is*

$$\sum_{i=1}^{c}\sum_{k=1}^{n}(1/c)^m||x_k - c(3,3)||^2.$$

- *Fuzzy membership with value* $1/c$ *to all clusters but four different cluster centers, each at the center of the corresponding cloud (i.e., cluster centers in positions* $(1,1)$*,* $(1,5)$*,* $(5,1)$*, and* $(5,5)$*. The objective function is*

$$\begin{aligned}\sum(1/c)^m \quad [\quad & ||x_k - c(1,1)||^2 \\ + \quad & ||x_k - c(1,5)||^2 \\ + \quad & ||x_k - c(5,1)||^2 \\ + \quad & ||x_k - c(5,5)||^2]\end{aligned}$$

- *Fuzzy membership values according to the result of the fuzzy c-means algorithm.*

Tables 2 and 3 give the objective functions of these alternative solutions for a few values of m*. They are, respectively, in columns HCM, of1c (for objective function with one single cluster center), of4c (for objective function with four different cluster centers), and fcm.*

This simple example shows that on the solely basis of the values of the objective function the results of the fuzzy c-means are not always better than the ones of HCM. We will see in more detail in the next section that the solution of fuzzy c means obtained with large m *is not always optimal, and other approaches are preferable for finding the best partition.*

In addition, for $m = 3$ *and* $m = 4$*, the solutions of fuzzy c-means tend to be such that all cluster centers are in positions* $(3,3)$*. For* $m = 2$*, the best solution has cluster centers in* $(1.21, 1.21)$*,* $(1.21, 4.78)$*,* $(4.78, 1.21)$ *and* $(4.78, 4.78)$*.*

3. The effects of the parameter m

We have studied how the parameter m affects the outcome of the optimization problem doing two different types of experiments.

The first type of experiments consists of studying large values of the parameter m and how they influence the solution of the optimization process. As we will discuss later, the solutions are not accurate when m is large and we propose an alternative approach, instead of applying the clustering algorithm with the given m.

The second type of experiments is to compare the partitions the algorithm delivers with different m. We will see that the partitions disagree when m diverge. Nevertheless, when the divergence is small, the disagreement between partitions is small; and that the divergence tends to increase largely after a certain point.

x_1	x_2
0.5	0.5
0.5	1.5
1.5	0.5
1.5	1.5
0.5	4.5
0.5	5.5
1.5	4.5
1.5	5.5
4.5	0.5
5.5	0.5
4.5	1.5
5.5	1.5
4.5	4.5
5.5	4.5
4.5	5.5
5.5	5.5

Table 1: Example of 16 records.

m	HCM	of1c	of4c	fcm
1	8	136	264	8
2	8	34	66	8.518
3	8	8.5	16.5	8.5
4	8	2.125	4.125	2.124

Table 2: Example of 16 records. Values of the objective function for (i) the crisp partitions (column HCM for Hard c-means), for the case of all cluster center in the position (3,3) (column of1c), for the case of four different cluster centers (column of4c) and for the result of fuzzy c-means (column fcm).

m_2	$m_1 = 1$	$m_2 = 2$	$m_3 = 3$	$m_4 = 4$
1	8	9.48	135.5	135.99
2	7.40	8.51	33.99	34
3	3.65	3.87	8.50	8.5
4	1.20	1.22	2.12	2.12

Table 3: Example of 16 records. Results of the objective function when the set of centroids is computed with $m_1 = 1, 2, 3, 4$ and the objective function is then computed for $m_2 = 1, 2, 3, 4$.

3.1. The parameter m and the objective function

For large values of m, Algorithm 1 leads to solutions in which membership values are similar to $1/c$ for most data elements to all clusters. When this happens, the iterative process also causes that the centroids tend to be in the center of the data being clustered. This is so because the centroids (as can be seen in the equation of Step 3) are just calculated as the mean values of the elements in the clusters. So, when all elements are in all clusters with the same membership, they contribute equally to all clusters.

Due to this effect, we have studied whether for large m it is preferable to compute the optimal of the fuzzy c-means with the given m, or if it is preferable to compute an optimal solution with a smaller m and then use the large m to compute the membership values and, also, the objective function.

In order to make explicit our computations, we start describing the computation of the objective function. Let $OF(X, v_i, m)$ denote the computation of the objective function of fuzzy c-means when we use a given data set X, the centroids v_i and a given m. That is, it computes the value given by Equation 1 taking into account that the membership functions are determined using the Equation in Step 2 of the algorithm.

Then, we will consider two values of m. Let m_1 represent a value near to 1, and let m_2 represent a larger value ($m_2 > m_1$). Then, we consider on the one hand the objective function of the problem $FCM(X, m_2, c)$. Formally, let us denote by

$$< OF, v >= FCM(X, m_1, c)$$

that the algorithm is applied to the data set X with parameters m_1 and c and it returns the vectors of centroids v as well as the value of the objective function OF.

On the other hand we consider the solution of the problem for m_1 and then recompute the objective function using the centroids of m_1 but using m_2 in the expression for the objective function. That is, we compute first

$$< OF, v_1 >= FCM(X, m_1, c)$$

and then use v_1 to compute

$$OF' = OF(X, v_1, m_2).$$

We then can compare OF and OF'.

Tables 3 and 4 display the results of applying this approach to the set of 16 records from Example 1. For this purpose we used our own implementation of the algorithm using the programming language R.

Table 3 shows the objective function for $m_2 = 1, 2, 3$ and 4 when the centroids have been computed with $m_1 = 1, 2, 3, 4$. It can be seen that the best solution for a given m_2 is not the one with $m_1 = m_2$ but with a $m_1 = 1$. This table has been computed applying fuzzy c-means 10 times and selecting the best solution (best in terms of the objective function).

m_1	$m_2 = 4$	$m_2 = 1.7$
1.0	1.20469	7.902152
1.1	1.20469	7.902152
1.2	1.20469	7.902152
1.3	1.20469	7.902156
1.4	1.20462	7.902245
1.5	1.20433	7.903276
1.6	1.20366	7.911207
1.7	1.20307	7.949283
1.8	1.20417	8.078378
1.9	1.21035	8.431363
2	1.22762	9.293082
3	2.125	51.53413
4	2.125	51.53436

Table 4: Example of 16 records. Values of the objective function for $m_2 = 4$ and $m_2 = 1.7$ and different values of m_1.

In Table 4, the column $m_2 = 4$ represents the computation of the objective function for $m_2 = 4$ when we use the centroids obtained, respectively, with $m_1 = 1, 1.1, 1.2, \ldots$. It can be seen that the centroids that minimize the objective functions are the ones computed with low values of m_1 and not the ones of $m_1 = 4$. The column $m_2 = 1.7$ represents the objective functions computed for $m_2 = 1.7$ with the centroids obtained with $m_1 = 1, 1.1, 1.2, \ldots$.

We can see that the larger is m, in general, the worse are the results, being the values between 1 and 2 resulting in a similar objective function, and, in some cases, decreasing.

In order to compute Table 4, we have applied the fuzzy c-means algorithm 200 times for each m and selected the solution with a minimal objective function.

We have applied the same study to two datasets. One is the IRIS dataset, which consists of 150 records with 4 numerical variables. The second one is the QUAKES dataset. It consists of 1000 records each one described in terms of 5 numerical variables. We used these data files as provided by R.

Tables 5 and 6 display the values obtained for the objective function OF and OF'. It can be seen that OF' leads to the best results in most of the experiments. The larger the difference between m_1 and m_2, the better the results obtained by OF'.

So, the experiments show that for large values of m, the standard approach for fuzzy c means makes the optimization problem hard and the results are not optimal. In these cases it is preferable to use a lower value of m, apply the algorithm and converge using this lower value, and, finally, if needed, compute the membership values and/or the objective

m_1	m_2	OF'	OF
1.1	4	0.329	0.681
1.3	4	0.336	0.681
1.5	4	0.341	0.681
1.7	4	0.341	0.681
1.9	4	0.346	0.681
2.0	4	0.324	0.681
2.2	4	0.357	0.681
2.4	4	0.428	0.681
2.6	4	0.515	0.681
2.8	4	0.627	0.681
3.0	4	0.672	0.681
4.0	4	0.681	0.681
1.1	2	14.163	18.911
1.3	2	13.657	18.956
1.5	2	13.886	19.070
1.7	2	14.143	18.470
1.9	2	17.470	18.622

Table 5: Best OF after 10 different executions of FCM with the IRIS dataset and with $c = 5$. $OF = OF(X, FCM(X, m_1), m_2)$ and $OF = FCM(X, m_2)$.

m_1	m_2	OF'	OF
1.1	2.5	14.162	18.910
1.3	2.5	13.656	18.956
1.5	2.5	13.886	19.070
1.7	2.5	15.144	18.470
1.9	2.5	17.478	18.623

Table 6: Bests OF after 10 different executions of FCM with the QUAKES dataset and with $c = 10$. $OF = OF(X, FCM(X, m_1), m_2)$ and $OF = FCM(X, m_2)$.

function with the actual large value of m.

3.2. The partitions and the parameter m

We have studied how the partitions obtained by fuzzy c-means diverge when m increases. To do so we have compared the outcome of the algorithm for different values of m applied to the same data set with the same number of clusters.

The partitions are compared using the cluster centers. A distance is computed between the two sets of cluster centers. Let v_i and w_i represent the two sets of cluster centers. Then, we first align the two cluster centers (assigning each one in v_i to the nearest one in w_i), and then define their distance as the maximum distance between pairs. That is, for aligned centers, we compute

$$d(v, w) = \max_i d(v_i, w_i).$$

We denote by $d(v_{1.1}, w_m)$ the distance between the optimal solution for $m_1 = 1.1$ and the optimal solution for $m_2 = m$. We compared the results for $m_1 = 1.1$ and $m_2 = 1 + t * 0.1$ for $t = 1, \ldots, 20$.

In order that the comparison is meaningful, the centroids v_i and w_i should be the global minima (or near enough to the global minima) of the optimization problem. Otherwise, the comparison is not meaningful, and the analysis of the function $d(v_{1.1}, w_m)$ can lead to misleading information. In particular, if different executions lead to different local optima, each execution will be comparing absolutely different solutions.

In order to ensure that we are really comparing the partitions of the global optima, we applied at least 20 executions of the fuzzy c-means. In some of the experiments this was not enough and we applied up to 100 executions.

For the experiments in this section we used three data files. We used the two described in the previous section (IRIS and QUAKES). In addition, we used the THEOP file, which consists of 132 records described in terms of 5 numerical variables. This file is also provided by the software package R. For the computation of these experiments we used the fuzzy c-means algorithm (cmeans) provided by the package e1071 in R.

The results show how the cluster centers diverge. We can see that in most of Figures 1, 2, 3, 4, 5, and 6 small values have small differences, and that for larger values the differences increase largely. We see that in some experiments (Figures 2, 4, 5, and 6) the difference is not monotonic with respect to the difference in m. This is usually due to solutions that are not optimal. In most of the cases, increasing the number of applications of fuzzy c-means the monotonicity of the curve improves.

Note that we have better curves with low c than with large c, where it is more difficult to find the global optima.

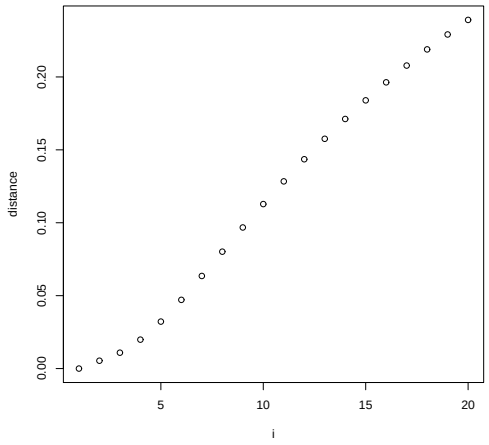

Figure 1: Distance between the centroids obtained with $m = 1.1$ and the ones of $m = 1 + 0.1t$ for $t = 1, \ldots, 20$. Case of the Iris file with $c = 3$, and 20 executions for each m.

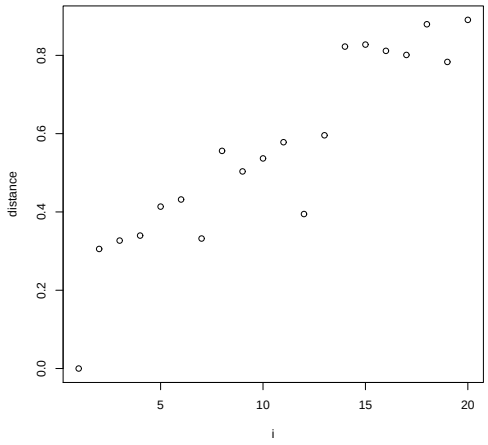

Figure 2: Distance between the centroids obtained with $m = 1.1$ and the ones of $m = 1 + 0.1t$ for $t = 1, \ldots, 20$. Case of the Iris file with $c = 20$, and 1000 executions for each m.

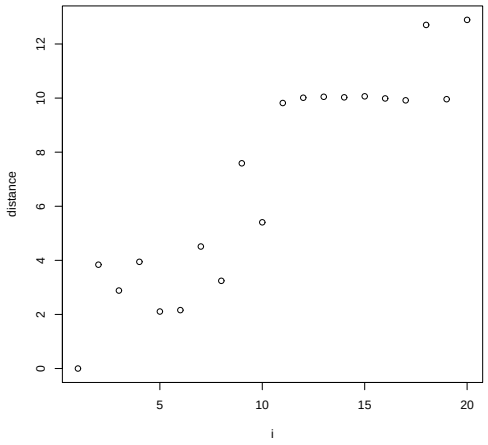

Figure 4: Distance between the centroids obtained with $m = 1.1$ and the ones of $m = 1 + 0.1t$ for $t = 1, \ldots, 20$. Case of the THEOP file with $c = 15$, and 100 executions for each m.

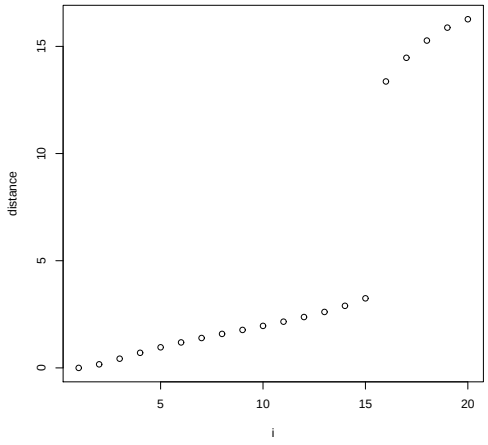

Figure 3: Distance between the centroids obtained with $m = 1.1$ and the ones of $m = 1 + 0.1t$ for $t = 1, \ldots, 20$. Case of the THEOP file with $c = 4$, and 100 executions for each m.

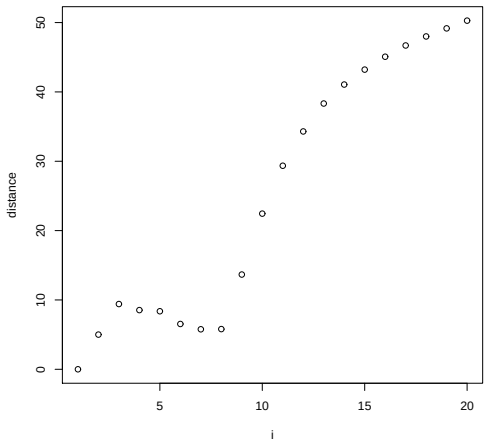

Figure 5: Distance between the centroids obtained with $m = 1.1$ and the ones of $m = 1 + 0.1t$ for $t = 1, \ldots, 20$. Case of the Quakes file with $c = 4$, and 100 executions for each m.

4. Summary and future work

In this work we have studied the effect of the parameter m in the results obtained by fuzzy c-means.

We have seen that for large values of m the partitions tend to be blurred and the centroids concentrated in the center of the data. Let m be the large parameter. In this case, in order to have a better result of the objective function it is preferable to apply the algorithm with a value of m' such that is smaller $m' < m$ and then use the original m to compute the objective function (and the membership degrees). This seems to lead to better results than applying the algorithm with m.

We have also shown that given two values m_1 and m_2, when their difference is small, the clusters obtained by the fuzzy c-means algorithm are similar. Then, after a point (in some cases $|m_1 - m_2| > 0.5$) the clusters centers start to be rather different. This can be taken into account when applying fuzzy c-means.

We consider two lines for future work. One is the extension of the study described here to other types of fuzzy clustering algorithms as e.g. entropy based fuzzy c-means. The other is to study the relationship between clustering results, the parameter m, and the fuzzy cluster validity indices.

5. Acknowledgments

Partial support by the Spanish MEC project and COPRIVACY (TIN2011-27076-C03-03) is acknowl-

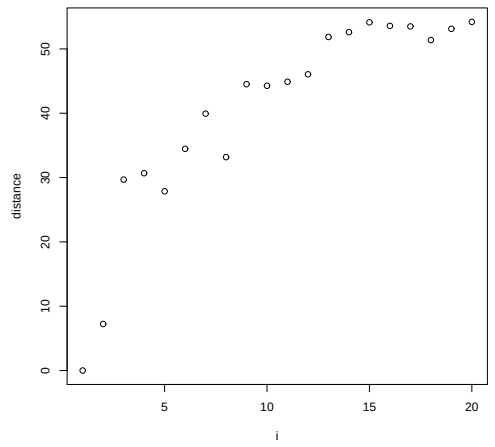

Figure 6: Distance between the centroids obtained with $m = 1.1$ and the ones of $m = 1 + 0.1t$ for $t = 1, \ldots, 20$. Case of the Quakes file with $c = 20$, and 100 executions for each m.

edged. Partial support of the FP7 EU project Data Without Boundaries is also acknowledged.

References

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

[2] Honda, K., Ichihashi, H., Masulli, F., Rovetta, S. (2007) Linear Fuzzy Clustering With Selection of Variables Using Graded Possibilistic Approach, IEEE Trans. on Fuzzy Systems 15 878-889.

[3] Höppner, F., Klawonn, F., Kruse, R., Runkler, T. (1999) Fuzzy cluster analysis, Wiley.

[4] Miyamoto, S. (1999) Introduction to fuzzy clustering (in Japanese), Ed. Morikita, Japan.

[5] Miyamoto, S., Mukaidono, M. (1997) Fuzzy c - means as a regularization and maximum entropy approach, Proc. of the 7th IFSA Conference, Vol.II 86-92.

[6] Miyamoto, S., Umayahara, K. (2000) Fuzzy c-means with variables for cluster sizes (in Japanese), Proc. of the 16th Fuzzy System Symposium, 537-538.

[7] Pal, N. R., Pal, K., Keller, J. M., Bezdek, J. C. (2005) A possibilistic fuzzy c-means clustering algorithm, IEEE Trans. on Fuzzy Systems 13:4 517-530.

[8] Szilágyi, L., Varga, Z. R., Szilágyi, S. M. (2014) Application of the Fuzzy-Possibilistic Product Partition in Elliptic Shell Clustering, Proc. MDAI 2014, LNCS 8825 158-169.

[9] Torra, V., Endo, Y., Miyamoto, S. (2011) Computationally Intensive Parameter Selection for Clustering Algorithms: the Case of Fuzzy c-Means with Tolerance, Int. J. of Intel. Systems 26:4 313-322.

[10] Torra, V., Miyamoto, S. (2004) Evaluating fuzzy clustering algorithms for microdata protection, PSD 2004, Lecture Notes in Computer Science 3050 175-186.

[11] Torra, V., Miyamoto, S. (2006) On the use of variable-size fuzzy clustering for classification, Proc. MDAI 2006, LNCS.