What is Fuzzy About Fuzzy Clustering? Understanding and Improving the Concept of the Fuzzifier

Frank Klawonn and Frank Höppner

Department of Computer Science
University of Applied Sciences Braunschweig/Wolfenbuettel
Salzdahlumer Str. 46/48
D-38302 Wolfenbuettel, Germany
f.klawonn@fh-wolfenbuettel.de

Abstract: The step from the well-known c-means clustering algorithm to the fuzzy c-means algorithm and its vast number of sophisticated extensions and generalisations involves an additional clustering parameter, the so called fuzzifier. This fuzzifier does not only control, how much clusters may or are assumed to overlap. It also has some negative effects causing problems for clusters with varying data density, noisy data and large data sets with a higher number of clusters. In this paper we take a closer look at what the underlying general principle of the fuzzifier is. Based on these investigations, we propose an improved more general framework that avoids the undesired effects of the fuzzifier.

1 Introduction

Clustering is an exploratory data analysis method applied to data in order to discover structures or certain groupings in a data set. Fuzzy clustering accepts the fact that the clusters or classes in the data are usually not completely well separated and thus assigns a membership degree between 0 and 1 for each cluster to every datum.

The most common fuzzy clustering techniques aim at minimizing an objective function whose (main) parameters are the membership degrees and the parameters determining the localisation as well as the shape of the clusters. Although the extension from deterministic (hard) to fuzzy clustering seems to be an obvious concept, it turns out that to actually obtain membership degrees between zero and one, it is necessary to introduce a so-called fuzzifier in fuzzy clustering. Usually, the fuzzifier is simply used to control how much clusters are allowed to overlap. In this paper, we provide a deeper understanding of the underlying concept of the fuzzifier and derive a more general approach that leads to improved results in fuzzy clustering.

Section 2 briefly reviews the necessary background in objective function-based fuzzy clustering. The purpose, background and the consequences of the additional parameter in fuzzy clustering – the fuzzifier – is examined in section 3. Based on these consideration and on a more general understanding, we propose an improved alternative to the fuzzifier in section 4 and outline possible other approaches in the final conclusions.

2 Objective Function-Based Fuzzy Clustering

Clustering aims at dividing a data set into groups or clusters that consist of similar data. There is large number of clustering techniques available with different underlying assumptions about the data and the clusters to be discovered. A simple and common popular approach is the so-called c-means clustering [4]. For the c-means algorithm it is assumed that the number of clusters is known or at least fixed, i.e., the algorithm will partition a given data set $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^p$ into c clusters. Since the assumption of a known or a priori fixed number of clusters is not realistic for many data analysis problems, there are techniques based on cluster validity considerations that allow to determine the number of clusters for the c-means algorithm as well. However, the underlying algorithm remains more or less the same, only the number of clusters is varied and the resulting clusters or the overall partition is evaluated. Therefore, it is sufficient to assume for the rest of the paper that the number of clusters is always fixed.

From the purely algorithmic point of view, the c-means clustering can be described as follows. Each of the c clusters is represented by a prototype $v_i \in \mathbb{R}^p$. These prototypes are chosen randomly in the beginning. Then each data vector is assigned to the nearest prototype (w.r.t. the Euclidean distance). Then each prototype is replaced by the centre of gravity of those data assigned to it. The alternating assignment of data to the nearest prototype and the update of the prototypes as cluster centres is repeated until the algorithm converges, i.e., no more changes happen.

This algorithm can also be seen as a strategy for minimizing the following objective function:

$$f = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij} d_{ij} \tag{1}$$

under the constraints

$$\sum_{i=1}^{c} u_{ij} = 1 \qquad \text{for all } j = 1, \ldots, n \tag{2}$$

where $u_{ij} \in \{0,1\}$ indicates whether data vector x_j is assigned to cluster i ($u_{ij} = 1$) or not ($u_{ij} = 0$). $d_{ij} =\parallel x_j - v_i \parallel^2$ is the squared Euclidean distance between data vector x_j and cluster prototype v_i.

Since this is a non-trivial constraint nonlinear optimisation problem with continuous parameters v_i and discrete parameters u_{ij}, the above mentioned algorithm, alternatingly optimising one set of parameters while the other set of parameters is considered as fixed, seems to be a reasonable approach for minimizing (1).

It should be noted that choosing the (squared) Euclidean distance as a measure for the distance between data vector u_{ij} and cluster i is just one choice out of many. In this paper we are not interested in the great variety of specific cluster shapes (spheres, ellipsoids, lines, quadrics,...) that can be found by choosing suitable cluster parameters and an adequate distance function. (For an overview we refer to [2, 5].) Our considerations can be applied to all cluster shapes. In this paper we concentrate on the assignment of data to clusters specified by the u_{ij}-values, especially in fuzzy clustering where the assumption $u_{ij} \in \{0,1\}$ is relaxed to $u_{ij} \in [0,1]$. In this case, u_{ij} is interpreted as the membership degree of data vector x_j to cluster i. Especially, when ambiguous data exist and cluster boundaries are not sharp, membership degrees are more realistic than crisp assignments. However, it turned out that the minimum of the objective function (1) under the constraints (2) is still obtained, when u_{ij} is chosen in the same way as in the c-means algorithm, i.e. $u_{ij} \in \{0,1\}$, even if we allow $u_{ij} \in [0,1]$. Therefore, an additional parameter m, the so-called fuzzifier [1], was introduced and the objective function (1) is replaced by

$$f = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^{m} d_{ij}. \tag{3}$$

Note that the fuzzifier m does not have any effects, when we stick to hard clustering. The fuzzifier $m > 1$ is not subject of the optimisation process and has to be chosen in advance. A typical choice is $m = 2$. We will discuss the effects of the fuzzifier in the next section. The fuzzy clustering approach with the objective function (3) under the constraints (2) and the assumption $u_{ij} \in [0,1]$ is called probabilistic clustering, since due to the constraints (2) the membership degree u_{ij} can be interpreted as the probability that x_j belongs to cluster i.

This still leads to a nonlinear optimisation problem, however, in contrast to hard clustering, with all parameters being continuous. The common technique for minimizing this objective function is similar as in hard clustering, alternatingly optimise either the membership degrees or the cluster parameters while considering the other parameter set as fixed.

Taking the constraints (2) into account by Lagrange functions, the minimum of the objective function (3) w.r.t. the membership degrees is obtained at [1]

$$u_{ij} \;=\; \frac{1}{\sum_{k=1}^{c}\left(\frac{d_{ij}}{d_{kj}}\right)^{\frac{1}{m-1}}}, \tag{4}$$

when the cluster parameters, i.e. the distance values d_{ij}, are considered to be fixed. (If $d_{ij} = 0$ for one or more clusters, we deviate from (4) and assign x_j with membership degree 1 to the or one of the clusters with $d_{ij} = 0$ and choose $u_{ij} = 0$ for the other clusters i.)

If the clusters are represented by simple prototypes $v_i \in \mathbb{R}^p$ and the distances d_{ij} are the squared Euclidean distances of the data to the corresponding cluster prototypes as in the hard c-means algorithm, the minimum of the objective function (3) w.r.t. the cluster prototypes is obtained at [1]

$$v_i \;=\; \frac{\sum_{j=1}^{n} u_{ij}^m x_j}{\sum_{j=1}^{n} u_{ij}^m}, \tag{5}$$

when the membership degrees u_{ij} are considered to be fixed. The prototypes are still the cluster centres. However, using $[0,1]$-valued membership degrees means that we have to compute weighted cluster centres. The fuzzy clustering scheme using alternatingly equations (4) and (5) is called fuzzy c-means algorithm (FCM). As mentioned before, more complicated cluster shapes can be detected by introducing additional cluster parameters and a modified distance function. Our considerations apply to all these schemes, but it would lead too far to discuss them in detail. However, we should mention that there are alternative approaches to fuzzy clustering than only probabilistic clustering. Noise clustering [3] maintains the principle of probabilistic clustering, but an additional noise cluster is introduced. All data have a fixed (large) distance to the noise cluster. In this way, data that are near the border between two clusters, still have a high membership degree to both clusters as in probabilistic clustering. But data that are far away from all clusters will be assigned to the noise cluster and have no longer a high membership degree to other clusters. Our investigations and our alternative approach fit also perfectly to noise clustering. We do not cover possibilistic clustering [6] where the probabilistic constraint is completely dropped and an additional term in the objective function is introduced to avoid the trivial solution $u_{ij} = 0$ for all i, j. However, the aim of possibilistic clustering is actually not to find the global optimum of the corresponding objective function, since this is obtained, when all clusters are identical [7].

3 Understanding the Fuzzifier

The fuzzifier controls how strong clusters may overlap. Figure 1 illustrates this effect by placing two cluster centres on the x-axis at 0 and 1. The leftmost plot shows the membership functions for the two clusters, when the fuzzifier is set to $m = 2$, the middle one for $m = 1.5$ (and the rightmost will be discussed later). It is well known and can be seen easily that for a larger m the transition from a high membership degree from one cluster to the other is more smooth than for a smaller m. Looking at the extremes, we obtain crisp $\{0, 1\}$-valued membership degrees for $m \to 1$ and equal membership degrees to all clusters for $m \to \infty$ with all cluster centres converging to the centre of the data set. Note that we have chosen a non-symmetric w.r.t. to the cluster centres, so that images do not look symmetric, although the membership functions are complimentary.

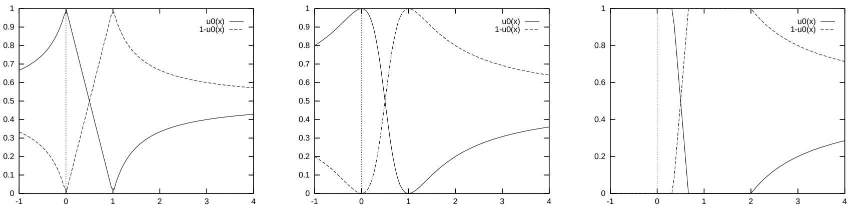

Fig. 1. Effects of the fuzzifier (from left to right): $m = 2$, $m = 1.5$, $\beta = 0.5$

The update equation (4) for the membership degrees derived from the objective function (3) can lead to undesired or counterintuitive results, because zero membership degrees never occur (except in the extremely rare case, when a data vector coincides with a cluster centre). No matter, how far away a data vector is from a cluster and how well it is covered by another cluster, it will still have nonzero membership degrees to all other clusters.

Figure 2 shows an undesired side-effect of the probabilistic fuzzy clustering approach. There are obviously three clusters. However, the upper cluster has a much higher data density than the other two. This single dense cluster attracts all other cluster prototypes so that the prototype of the left cluster is slightly drawn away from the original cluster centre and the prototype we would expect in the centre of the lower left cluster migrates completely into the dense cluster. In the figure we have also indicated for which cluster a data vector has the highest membership degree.

Another counterintuitive effect of probabilistic fuzzy clustering occurs in the following situation. Assume we have a data set that we have clustered already. Then we add more data to the data set in the form of a new cluster that is far away from all other clusters. If we recluster this enlarged data set with one more cluster as the original data set, we would expect the same result, except that the new data are covered by the additional cluster, i.e., we would assume that the new cluster has no influence on the old ones. However, since we never obtain zero membership degrees, the new data (cluster) will influence the old clusters.

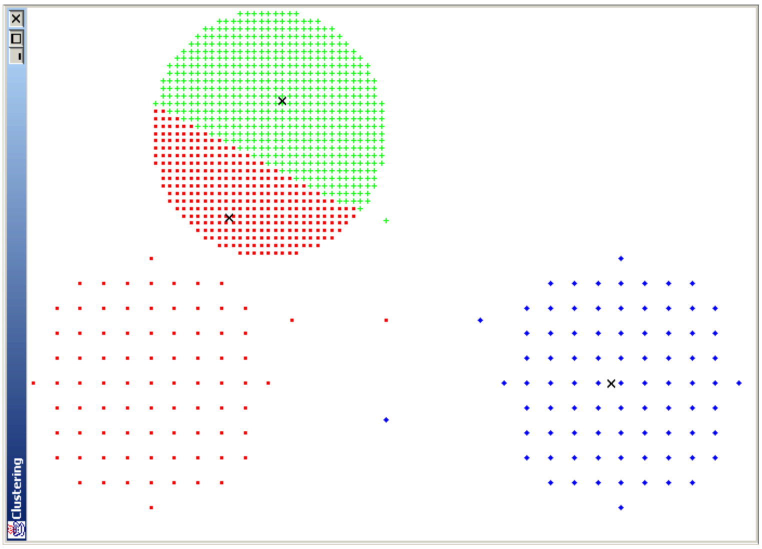

Fig. 2. Clusters with varying density

This means also that, if we have many clusters, clusters far away from the centre of the whole data set tend to have their computed cluster centres drawn into the direction of the centre of the data set.

These effects can be amended, when a small fuzzifier is chosen. The price for this is that we end up more or less with hard clustering again and even neighbouring clusters become artificially well separated, although there might be ambiguous data between these clusters.

As can be seen in figure 1, the membership degrees tend to increase again, when we move far away from all clusters. This undesired effect can be amended by applying noise clustering. Nevertheless, even in the case of noise clustering, noisy data, no matter how far away they are from all other clusters, will still have nonzero membership degrees to all clusters.

In order to propose an alternative to the fuzzifier approach, we examine more closely what impact the fuzzifier has on the objective function. When we want to generalise the idea of deterministic or hard clustering to fuzzy clustering, using the original objective function (1) of hard clustering simply allowing the u_{ij} values to be in $[0, 1]$ instead of $\{0, 1\}$, still leads to crisp partitions, as we have already mentioned before. In order to better understand why, let us consider the following situation. We fix the cluster prototypes, i.e. the distance values d_{ij}, for the moment – we might even assume that we have already found the best prototypes – and want to minimize the objective function (1) by choosing the appropriate membership degrees taking the constraints (2) into account. A good starting point seems to be to choose $u_{i_0j} = 1$, if $d_{i_0j} \leq d_{ij}$ for all $i = 1, \ldots, c$ and $u_{ij} = 0$ otherwise. When we try to further reduce the resulting value of the object function by decreasing an u_{i_0j}-value that was set to one, we have to increase another u_{ij}-value to satisfy the constraint (2). When we reduce u_{i_0j} by ε and increase u_{ij} by ε instead, the change in the objective function will be

$$\Delta \;=\; \varepsilon \cdot d_{ij} - \varepsilon \cdot d_{i_0j} \;=\; \varepsilon(d_{ij} - d_{i_0j}).$$

Since $d_{i_0j} \leq d_{ij}$, this can only lead to an increase and therefore never to an improvement of the objective function. The trade-off by reducing u_{i_0j} and therefore increasing u_{ij} always means a bad pay-off in terms of the objective function. We can turn the pay-off into a good one, if we modify the objective function in the following way: A reduction of a u_{i_0j}-value near 1 by ε must have a higher decreasing effect than the increment of a u_{ij}-value near 0. Since the factor d_{i_0j} of u_{i_0j} is smaller than the factor d_{ij} of u_{ij}, we apply a transformation to the membership degrees in the objective function, such that a decrease of a high membership degree has a stronger decreasing effect than the increasing effect caused by an increase of a small membership value. One transformation, satisfying this criterion, is

$$g : [0,1] \to [0,1], \qquad u \mapsto u^m$$

with $m > 1$ that is commonly used in fuzzy clustering. However, there might be other choices as well. Which properties should such a transformation satisfy? It is obvious that g should be increasing and that we want $g(0) = 0$ and $g(1) = 1$. If g is differentiable and we apply the above mentioned reduction of the membership degree u_{i_0j} by ε, trading it in for an increase of the membership degree u_{ij} by a small value ε, the change in the objective function is now approximately

$$\Delta \;\approx\; \varepsilon \cdot g'(u_{ij}) \cdot d_{ij} - \varepsilon \cdot g'(u_{i_0j}) \cdot d_{i_0j} \;=\; \varepsilon(d_{ij} \cdot g'(u_{ij}) - d_{i_0j} \cdot g'(u_{i_0j})). \quad (6)$$

In order to let this decrease the objective function, we need at least $g'(u_{ij}) < g'(u_{i_0j})$, since $d_{i_0j} \leq d_{ij}$. More generally, we require that the derivative of g is increasing on $[0,1]$. $g(u) = u^m$ with $m > 1$ definitely has this property. Especially, for this transformation we have $g'(0) = 0$, so that it always pays off to get away from zero membership degrees. Figure 3 shows the identity (the line which does not transform the membership degrees at all), the transformation u^2 (the lower curve) and another transformation that also satisfies the requirements for g, but has a nonzero derivative at 0, so that it only pays off to have a nonzero membership degree, if the distance values d_{ij} is not too large in comparison to d_{i_0j}.

Let us take a closer look which effect the transformation g has on the objective function. Assume, we want to minimize the objective function

$$f \;=\; \sum_{i=1}^{c}\sum_{j=1}^{n} g(u_{ij})d_{ij} \qquad (7)$$

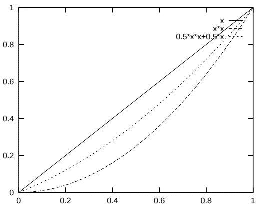

Fig. 3. Transformations for probabilistic clustering

under the constraints (2) w.r.t. the values u_{ij}, i.e., we consider the distances as fixed. The constraints lead to the Lagrange function

$$L = \sum_{i=1}^{c}\sum_{j=1}^{n} g(u_{ij})d_{ij} + \sum_{j=1}^{n} \lambda_j \left(1 - \sum_{i=1}^{c} u_{ij}\right)$$

and the partial derivatives

$$\frac{\partial L}{\partial u_{ij}} = g'(u_{ij})d_{ij} - \lambda_j. \tag{8}$$

At a minimum of the objective function the partial derivatives must be zero, i.e. $\lambda_j = g'(u_{ij})d_{ij}$. Since λ_j is independent of i, we must have $g'(u_{ij})d_{ij} = g'(u_{kj})d_{kj}$ for all i,k at a minimum. This actually means that these products must be balanced during the minimization process. In other words, the minimum is not reached unless the Δ-values in (6) are all zero.

4 An Alternative for the Fuzzifier

Taking into account the analysis carried out in the previous section, we propose a new approach to fuzzy clustering that replaces the transformation $g(u) = u^m$ by another transformation. In principle, we can think of any differentiable function satisfying the requirements stated in the previous section. However, when we want to maintain the computationally more efficient alternating optimisation scheme for fuzzy clustering with explicit update equations for the membership degrees, we easily run into problems for general functions g. From (8) we can immediately see that we will need the inverse of g' in order to compute the u_{ij}. Therefore, we restrict our considerations here to quadratic transformations. Since we require $g(0) = 0$ and $g(1) = 1$ and an increasing first derivative, the choices reduce to quadratic functions of the form $g(u) = \alpha u^2 + (1-\alpha)u$ with $0 \leq \alpha \leq 1$.

Instead of α we use the parameter

$$\beta = \frac{g'(0)}{g'(1)} = \frac{1-\alpha}{1+\alpha}.$$

We can easily compute $\alpha = \frac{1-\beta}{1+\beta}$. Let us assume that $d_{i_0 j}$ is the distance of data vector x_j to the nearest cluster and d_{ij} is the distance of x_j to another cluster further away. Then β indicates the lower bound that the quotient $\frac{d_{i_0 j}}{d_{ij}}$ must exceed, in order to have a nonzero membership degree of x_j to the cluster that lies further away. For $\beta = 0$ we obtain standard fuzzy clustering with fuzzifier $m = 2$ and $\beta = 1$ leads to crisp clustering.

We now derive the update equations for our new clustering approach. We have to minimize the objective function

$$f \;=\; \sum_{i=1}^{c}\sum_{j=1}^{n}\left(\frac{1-\beta}{1+\beta}u_{ij}^2 + \frac{2\beta}{1+\beta}u_{ij}\right) d_{ij}$$

under the constraints (2) as well as $0 \leq u_{ij} \leq 1$. Computing the partial derivatives of the Lagrange function

$$L \;=\; \sum_{i=1}^{c}\sum_{j=1}^{n}\left(\frac{1-\beta}{1+\beta}u_{ij}^2 + \frac{2\beta}{1+\beta}u_{ij}\right) d_{ij} \;+\; \sum_{j=1}^{n}\lambda_j\left(1 - \sum_{i=1}^{c} u_{ij}\right)$$

and solving for u_{ij} we obtain

$$u_{ij} \;=\; \frac{1}{1-\beta}\left(\frac{(1+\beta)\lambda_j}{2d_{ij}} - \beta\right) \tag{9}$$

if $u_{ij} \neq 0$. Using $\sum_{k:u_{kj}\neq 0} u_{kj} = 1$, we can compute

$$\lambda_j \;=\; \frac{2(1+(\hat{c}-1)\beta)}{(1+\beta)\sum_{k:u_{kj}\neq 0}\frac{1}{d_{kj}}}$$

where $\hat{c}$ is the number of clusters to which data vector x_j has nonzero membership degrees. Replacing λ_j in (9), we finally obtain the update equation for

$$u_{ij} \;=\; \frac{1}{1-\beta}\left(\frac{1+(\hat{c}-1)\beta}{\sum_{k:u_{kj}\neq 0}\frac{d_{ij}}{d_{kj}}} - \beta\right). \tag{10}$$

We still have to determine which u_{ij} are zero. This can be done in the following way. For a fixed j sort the distances d_{ij} in decreasing order. Without loss of generality let us assume $d_{1j} \geq \ldots \geq d_{cj}$. If there are zero membership degrees at all, we know that for minimizing the objective function the u_{ij}-values with larger distances have to be zero. (10) does not apply to these u_{ij}-values. Therefore, we have to find the smallest index i_0 to which (10) is applicable, i.e. for which it yields a positive value. For $i < i_0$ we have $u_{ij} = 0$ and for $i \geq i_0$ the membership degree u_{ij} is computed according to (10) with $\hat{c} = c + 1 - i_0$.

With this modified algorithm the data set in figure 2 is clustered correctly (for instance with $\beta = 0.5$, the corresponding transformation g is the curve in the middle in figure 3). Especially, when our modified approach is coupled with noise clustering, most of the undesired effects of fuzzy clustering can be avoided and the advantages of a fuzzy approach can be maintained.

When using our modified algorithm, the update equations for the cluster prototype remain the same – for instance, in the case of FCM as in (5) – except that we have to replace u_{ij}^m by

$$g(u_{ij}) \;=\; \alpha u_{ij}^2 + (1-\alpha)u_{ij} \;=\; \frac{1-\beta}{1+\beta}u_{ij}^2 + \frac{2\beta}{1+\beta}u_{ij}.$$

5 Conclusions

We have proposed a new approach to fuzzy clustering that overcomes the problem in fuzzy clustering that all data tend to influence all clusters. From the computational point of view our algorithm is slightly more complex than the standard fuzzy clustering scheme. The update equations are quite similar to standard (probabilistic) fuzzy clustering. However, for each data vector we have to sort the distances to the clusters in each iteration step. Since the number of clusters is usually quite small and the order of the distances tends to converge quickly, this additional computational effort can be kept at minimum.

As a future work, we will extend our approach to more general transformations g and apply the balancing scheme induced by (6) directly to compute the membership degrees.

References

1. Bezdek JC (1981) Pattern recognition with fuzzy objective function algorithms. Plenum Press
2. Bezdek JC, Keller J, Krishnapuram R, Pal NR (1999) Fuzzy models and algorithms for pattern recognition and image processing. Kluwer, Boston
3. Davé, RN (1991) Characterization and detection of noise in clustering. Pattern Recognition Letters 12: 657–664
4. Duda, R, Hart, P (1973) Pattern classification and scene analysis. Wiley, New York
5. Höppner F, Klawonn F, Kruse R, Runkler T (1999) Fuzzy cluster analysis. Wiley, Chichester
6. Krishnapuram R, Keller J (1993) A possibilistic approach to clustering. IEEE Trans. on Fuzzy Systems 1: 98–110
7. Timm H, Borgelt C, Kruse R (2002) A modification to improve possibilistic cluster analysis. IEEE Intern. Conf. on Fuzzy Systems, Honululu (2002)