Mathware & Soft Computing 7 (2000)

Fuzzy Clustering: Insights and a New Approach

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

Abstract

Fuzzy clustering extends crisp clustering in the sense that objects can belong to various clusters with different membership degrees at the same time, whereas crisp or deterministic clustering assigns each object to a unique cluster. The standard approach to fuzzy clustering introduces the so-called fuzzifier which controls how much clusters may overlap. In this paper we illustrate, how this fuzzifier can help to reduce the number of undesired local minima of the objective function that is associated with fuzzy clustering. Apart from this advantage, the fuzzifier has also some drawbacks that are discussed in this paper. A deeper analysis of the fuzzifier concept leads us to a more general approach to fuzzy clustering that can overcome the problems caused by the fuzzifier.

1 Introduction

Clustering is an exploratory data analysis technique and is applied in the data analysis process at a state where no precise model of the data is known. Therefore, it is necessary that the clustering process is self-guided as far as possible and will avoid unsuitable clustering results that do not reflect the structure of the data set properly.

Most fuzzy clustering techniques aim at minimizing an objective function that usually has a number of undesired local minima. After briefly reviewing the basic principles of fuzzy clustering, we illustrate in section 2, why fuzzy clustering can reduce the number of local minima in comparison to crisp clustering. However, the objective function associated with fuzzy clustering can sometimes also be misleading. We discuss and analyze some of the drawbacks of fuzzy clustering in section 3. Based on the analysis provided in section 3 we propose new approaches to fuzzy clustering to overcome these drawbacks in section 4 and conclude the paper with some recommendations for the implementation of the proposed algorithms in section 5.

2 Why Fuzzy Clustering

The most common approach to fuzzy clustering is the so called probabilistic clustering with the objective function

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

that should be minimized under the constraints

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

and $u_{ij} \in [0,1]$ for all $i \in \{1,\ldots,c\}$ and all $j \in \{1,\ldots,n\}$.

It is assumed that the number of clusters c is fixed. We will not discuss the issue of determining the number of clusters here and refer for an overview to [4, 10]. The set of data to be clustered is $\{x_1,\ldots,x_n\} \subset \mathbb{R}^p$. u_{ij} is the membership degree of datum x_j to the ith cluster. d_{ij} is some distance measure specifying the distance between datum x_j and cluster i, for instance the (quadratic) Euclidean distance of x_j to the ith cluster centre. The parameter $m > 1$, called fuzzifier, controls how much clusters may overlap. The constraints (2) lead to the name probabilistic clustering, since in this case the membership degree u_{ij} can also be interpreted as the probability that x_j belongs to cluster i.

The parameters to be optimised are the membership degrees u_{ij} and the cluster parameters that are not given explicitly here. They are hidden in the distances d_{ij}. Since this is a non-linear optimisation problem, the most common approach to minimize the objective function (1) is to alternatingly optimise either the membership degrees or the cluster parameters while considering the other parameter set as fixed. Of course, there are other strategies to minimize the objective function. However, the alternating optimisation scheme seems to be the most efficient algorithm for this objective function.

In this paper we are not interested in the great variety of 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 again to [4, 10].) We only concentrate on the aspect of the membership degrees.

Taking the constraints in equation (2) into account by Lagrange functions, the minimum of the objective function (1) with respect to the membership degrees is obtained at [2]

$$u_{ij} = \frac{1}{\sum_{k=1}^{c}\left(\frac{d_{ij}}{d_{kj}}\right)^{\frac{1}{m-1}}}, \tag{3}$$

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 (3) 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 fuzzy c-means algorithm, the minimum of the objective function (1) with respect to the cluster prototypes is obtained at [2]

$$v_i = \frac{\sum_{j=1}^{n} u_{ij}^m x_j}{\sum_{j=1}^{n} u_{ij}^m}, \tag{4}$$

when the membership degrees u_{ij} are considered to be fixed. The prototypes are still the cluster centres. The cluster prototypes are simply the weighted centres of gravity where the weighting is based on the membership degrees.

The fuzzy clustering scheme using alternatingly equations (3) and (4) 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 [5] 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 considerable membership degree to other clusters. Our investigations and our alternative approaches fit also perfectly to noise clustering.

We do not cover possibilistic clustering [13] 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 [15].

Another approach that emphasizes a probabilistic interpretation in fuzzy clustering is described in [7] where membership degrees as well as membership probabilities are used for the clustering. In this way, some of the problems of the standard FCM scheme can be avoided as well. However, this approach assumes the use of the Euclidean or a Mahalanobis distance and is not suitable for arbitrary cluster shapes as in shell clustering.

Before we take a closer a look at the problems caused by fuzzy clustering, we will examine the advantages of fuzzy clustering over crisp clustering. Of course, one of the main advantages of fuzzy clustering is the ability to express ambiguity in the assignment of objects to clusters. But apart from this, experimental results prove that fuzzy clustering seems also to be more robust in terms of local minima of the objective function. Although we cannot give a general proof that fuzzy clustering is more robust than deterministic (crisp) clustering, we can at least support this conjecture by analyzing some examples in detail.

Crisp clustering uses the same objective function (1) including the constraints specified in (2). However, instead of allowing the membership degrees u_{ij} to assume

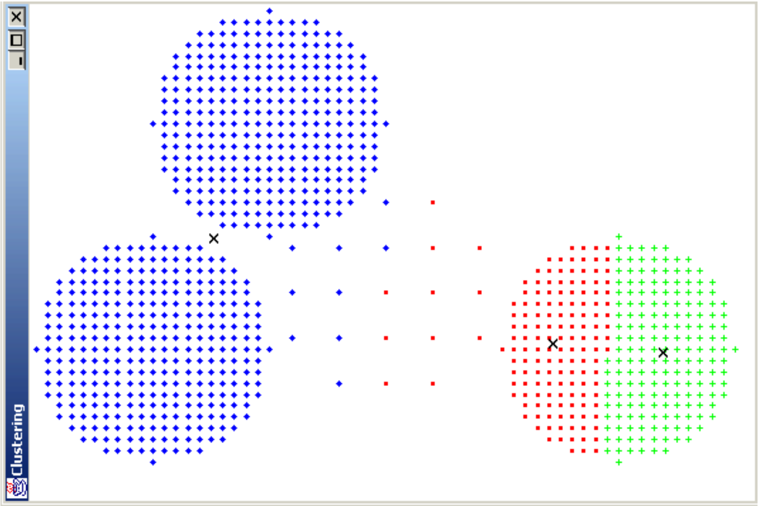


Figure 1: An undesired clustering result

values between 0 and 1, crisp clustering reduces the choice to either zero (not assigned to the corresponding cluster) or one (fully assigned to the corresponding cluster). In this case, the fuzzifier m has no effect at all. For crisp clustering the update equation (3) for the membership degrees is replaced by simply assigning each data vector x_j with membership degree one to the cluster i to which it has the smallest distance d_{ij} and choosing membership degree zero for the other clusters.

Figure 1 shows a simple artificial data set which we will use to illustrate one of the problems of crisp clustering. When we randomly initialise the cluster prototypes, the hard c-means algorithm (HCM) [6] – using the same form of the cluster prototypes and the same prototype update equation as FCM, but assuming crisp memberships $u_{ij} \in \{0, 1\}$ – tends to converge in almost 30% of the cases to the cluster partition and the prototypes indicated in the figure. This is definitely an undesired result. Although in the remaining 70% of the cases, HCM will converge to the desired clustering result, this example shows that the HCM objective function has – at least in this case – undesired local minima to which the alternating optimisation scheme might be attracted in a non-neglectable number of cases. When we carried out the same experiment with FCM with fuzzifier $m = 2$, we always obtained the desired prototypes located in the centres of the three spherical data clusters.

The undesired clustering result in figure 1 is obviously a local minimum of the HCM objective function. The left-most prototype is closer to the two data clusters on the left-hand side and all data from them are assigned to this prototype. Once the other two prototypes are too far away from the two data clusters on the left-hand side, they can only share the data from the one data cluster on the right-hand side. This means, in all cases where the randomly initialised prototypes are positioned in such a way that one of them is somewhere in the left half of figure 1 and the other two are positioned more to the lower right part, the alternating optimisation scheme will end up in this undesired local minimum. In this case the two prototypes on the right-hand side in figure 1 will never get a chance to capture one of the data points of the two data clusters on the left-hand side.

Why does the same effect not occur with FCM? The update equation (3) never yields zero membership degrees except for the case that the distance of a data object to a cluster prototype happens to be zero. For FCM this can only occur, when the coordinates of the data object coincide with the prototype coordinates. This means that all data (almost) always influence all prototypes. So even in the case that a single prototype captures the two data clusters on the left-hand side of figure 1, these data will still slightly attract the other two cluster prototypes in the FCM alternating optimisation scheme, in contrast to the HCM scheme where the other two prototypes will take no notice of the data in the two clusters on the left-hand side. Therefore, in the FCM case the two data clusters on the left-hand side seem to have enough power to attract a second prototype to them, even if there is one prototype very close to them and the other two are quite far away. So it seems that the introduction of $[0,1]$-valued membership degrees in FCM has a smoothing effect on undesired local minima in the objective function of HCM.

Unfortunately, we cannot illustrate this effect by looking at the objective function in the HCM and FCM case. The free parameters in these objective functions are the cluster prototypes and the membership degrees, the latter ones being also constraint. From the derivation of the alternating optimisation scheme (see for instance [2, 10]), we know how to choose the membership degrees, when we fix the cluster prototypes. This applies to crisp and fuzzy clustering. Therefore, in order to reduce the number of free parameters in the objective function, we can replace the membership degrees u_{ij} by the corresponding update equation. For example, in the case of FCM the objective function becomes

$$f = \sum_{i=1}^{c}\sum_{j=1}^{n}\left(\frac{1}{\sum_{k=1}^{c}\left(\frac{\| x_j - v_i \|^2}{\| x_j - v_k \|^2}\right)^{\frac{1}{m-1}}}\right)^{m} \| x_j - v_i \|^2$$

with the free parameter vectors $v_1, \ldots, v_c \in \mathrm{I\!R}^p$. In the example shown in figure 1 this means that we have $c = 3$ and $p = 2$, i.e. three prototype vectors v_1, v_2, v_3 each one having two components. All together we can think of an objective function with six free parameters, too many to graphically illustrate it.

In order to understand the smoothing effect of fuzzy membership degrees on undesired local minima in the objective function, we consider a simplified clustering problem. First of all, we restrict ourselves to a one-dimensional data set ($p = 1$). An objective function with two free parameters leads already to three-dimensional landscape as its graphical representation. However, to create a similar effect as in figure 1, we need at least three clusters. So even in the case of one-dimensional data, we already end up with at least three free parameters, making a graphical representation of the objective function impossible. Nevertheless, we can still construct an illustrating example, when we assume that we have three clusters, but

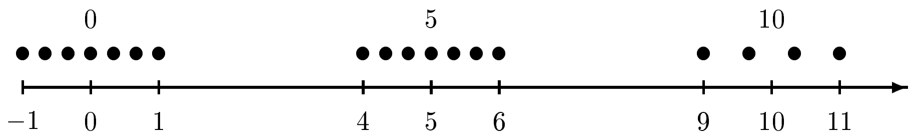
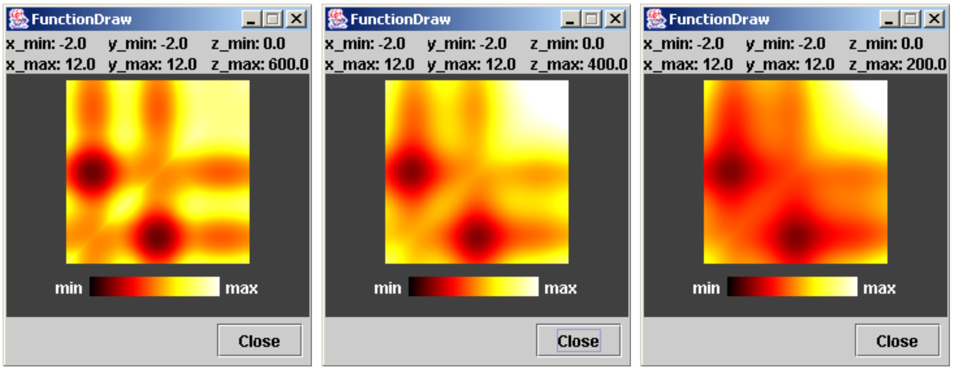


Figure 2: An extremely simple data set


FunctionDraw
x_min: -2.0 y_min: -2.0 z_min: 0.0
x_max: 12.0 y_max: 12.0 z_max: 600.0

FunctionDraw
x_min: -2.0 y_min: -2.0 z_min: 0.0
x_max: 12.0 y_max: 12.0 z_max: 400.0

FunctionDraw
x_min: -2.0 y_min: -2.0 z_min: 0.0
x_max: 12.0 y_max: 12.0 z_max: 200.0

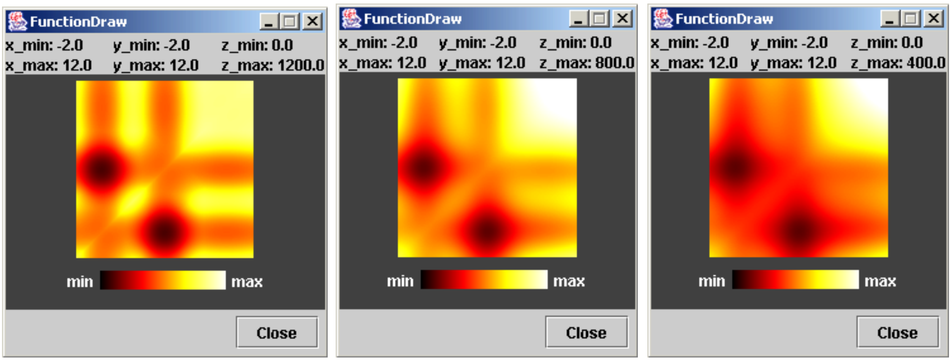
Figure 3: Objective functions for $N_{\text{cluster}} = 50$, $N_{\text{noise}} = 10$ of HCM, FCM with $m = 2$ and FCM with $m = 3$ (left to right)

one of them is a noise cluster.

We consider a data set of the structure illustrated in figure 2. In each of the intervals $[-1, 1]$ and $[4, 6]$ we place N_{cluster} equidistant data points and in the interval $[9, 11]$ we put N_{noise} equidistant data points, where $N_{\text{cluster}} \gg N_{\text{noise}}$. So we have two data clusters centred around 0 and 5 as well as some noise data near 10.

In figure 3 the objective functions for the data set with $N_{\text{cluster}} = 50$ and $N_{\text{noise}} = 10$ are illustrated. The noise distance was set to $\delta = 5$. The leftmost diagram shows the objective function for HCM, the middle one for FCM with $m = 2$ and the right-most one for FCM with $m = 3$. The values on the x- and y-axis determine the (one-dimensional) coordinates of the two cluster prototypes. A darker colour in each diagram indicates a lower (better) value of the objective function.

First of all, it is obvious that the objective functions must all be symmetric with respect to the main diagonal. When we exchange the two cluster prototypes, the value of the corresponding objective function will be the same. In all diagrams we can see strong local minima at approximately $(0, 5)$ and $(5, 0)$. This is exactly what we expect: The prototypes are correctly positioned into the centres 0 and 5 (or vice versa) of the data clusters. However, in addition to these desired (global) minima, there are other undesired local minima, namely at approximately $(0, 10), (5, 10), (10, 0), (10, 5)$. For these local minima one prototype covers one of



x_min: -2.0 y_min: -2.0 z_min: 0.0
x_max: 12.0 y_max: 12.0 z_max: 1200.0

x_min: -2.0 y_min: -2.0 z_min: 0.0
x_max: 12.0 y_max: 12.0 z_max: 800.0

x_min: -2.0 y_min: -2.0 z_min: 0.0
x_max: 12.0 y_max: 12.0 z_max: 400.0

Figure 4: Objective functions for $N_{\text{cluster}} = 100$, $N_{\text{noise}} = 10$ of HCM, FCM with $m = 2$ and FCM with $m = 3$ (left to right)

the data clusters, while the other prototype is mislead to the noise cluster. The alternating optimisation scheme starts with a random initialisation somewhere in these diagrams and will then slide down to the corresponding (usually the closest) local minimum of the objective function. From figure 3 it can been seen that the four undesired local minima are present in the HCM objective function, two of them gone in the case of FCM with fuzzifier $m = 2$ and all of them completely vanish for FCM with fuzzifier $m = 3$.

Figure 4 shows this effect even stronger, when the density of the data clusters is increased by setting N_{cluster} to 100, while keeping the other parameters as in figure 3.

3 Bad Effects of the Fuzzifier

In the previous section we have seen that fuzzy clustering has advantages over hard clustering, at least for the examples we have discussed. This is mainly due to the fact that in fuzzy clustering we allow the data to have some influence on a prototype, even though they might only be assigned to the corresponding cluster to a small degree. As we have mentioned already in the previous section, the standard (probabilistic) fuzzy clustering approach even leads to the effect that all data have influence on all cluster prototypes, no matter how far they are away from them.

Figure 5 shows an undesired side-effect of the probabilistic fuzzy clustering approach. There are obviously three data clusters. However, the upper cluster has a much higher data density than the other two. This single dense cluster attracts the cluster prototype of the lower left data cluster so that this prototype migrates completely into the dense cluster. Even the prototype covering the data cluster in the far right is slightly drawn in the direction of the dense cluster. This effect will even happen, when we position the cluster prototypes in the corresponding centres of the data clusters. Although each single data object in the dense cluster has only

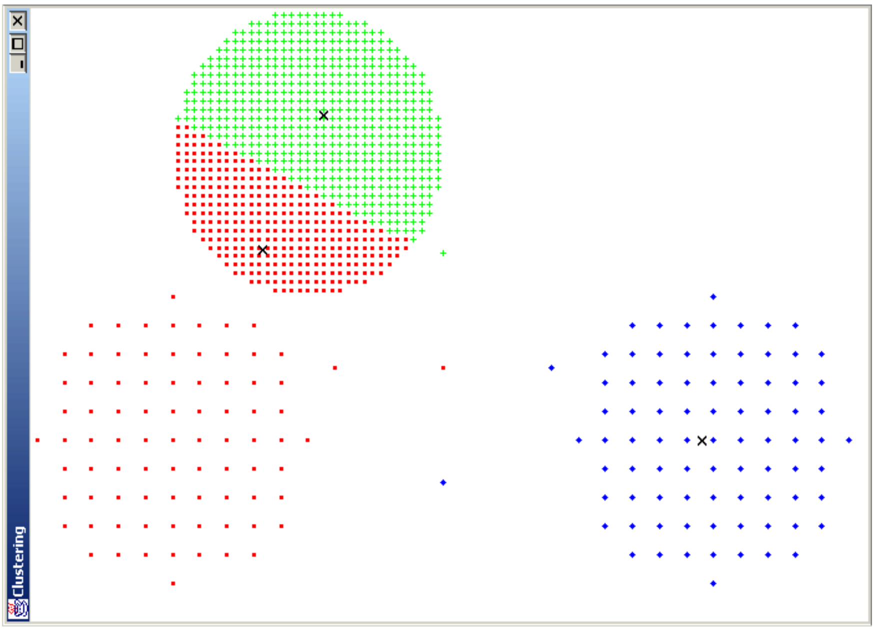


Figure 5: Clusters with varying density

a small membership degree to the prototypes of the other clusters, there are so many data objects in the dense cluster that they still manage to attract the other prototypes.

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, well separated 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.

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. And even a small fuzzifier will still lead to the effect that all data have a certain influence on all clusters.

As we have mentioned in the previous section, the fuzzifier has no effect, when we consider crisp clustering ($u_{ij} \in \{0,1\}$). Therefore, the most obvious generalisation from crisp to fuzzy clustering is not to use a fuzzifier at all, i.e. choose $m = 1$ for the fuzzifier. However, it is well known [2] that in this case even though we allow membership degrees between zero and one, a (local) minimum of the objective function can only be obtained, when we stick to crisp memberships. This was probably the main motivation for introducing the fuzzifier in the first step. Looking at the fuzzifier from a more general point of view, we have to deal with the objective function

$$f \;=\; \sum_{i=1}^{c}\sum_{j=1}^{n} g(u_{ij})d_{ij} \tag{5}$$

under the constraints (2). In fuzzy clustering we have

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

In other words, the fuzzifier is a special transformation function g applied to the membership degrees within the objective function. In [11] a detailed general analysis of the required properties and the effects of such a transformation function g was carried out. We need the following result from [11] here. When we want to minimize the objective function (5) under the constraints (2) with respect to 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

$$\frac{\partial L}{\partial u_{ij}} \;=\; g'(u_{ij})d_{ij} - \lambda_j. \tag{6}$$

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

$$g'(u_{ij})d_{ij} \;=\; g'(u_{kj})d_{kj} \tag{7}$$

for all i, k at a minimum. This actually means that these products must be balanced during the minimization process.

The balance equation (7) explains immediately, why we (nearly) never obtain zero membership degrees in standard fuzzy clustering. Since in this case we have $g(u) = u^m$ and therefore $g'(u) = m \cdot u^{m-1}$, we obtain $g'(0) = 0$. Satisfying the balance equation (7) with at least one value $u_{ij} = 0$ means that all other products $g'(u_{kj})d_{kj}$ must be zero as well. This can only be obtained in the rare case, when we have $d_{kj} = 0$ for some k or when we set all u_{kj} to zero. But the latter case is not in accordance with constraint (2).

The balance equation can also be understood in another way. Assume, we do not know the explicit solution for the u_{ij}-values for the chosen transformation g in the alternating optimisation scheme. In order to minimize the objective function we might start with some "good" first guess for the u_{ij}-values. We can then compare the products occurring in the balance equation. When we have $g'(u_{ij})d_{ij} < g'(u_{kj})d_{kj}$, we know the following (assuming that g' is continuous). When we increase u_{ij} by some very small value $\varepsilon > 0$ and decrease u_{kj} by ε (maintaining constraint (2) in this way), the value of the objective function will change approximately by the following value:

$$\Delta f \;\approx\; \varepsilon \cdot g'(u_{ij})d_{ij} - \varepsilon \cdot g'(u_{kj})d_{kj} \;=\; \varepsilon \cdot (g'(u_{ij})d_{ij} - g'(u_{kj})d_{kj}) \;<\; 0.$$

Thus increasing u_{ij} and decreasing u_{kj} slightly leads to a decrease in the objective function and therefore to a better solution. At the end of the following section we will develop a clustering algorithm that is based on this idea exploiting the balance equation.

4 Replacing the Fuzzifier

This section is devoted to possible alternatives for the transformation $g(u) = u^m$ used in standard fuzzy clustering. In [14] the alternating optimisation scheme is changed into an alternating (cluster) estimation scheme. The cluster prototypes are still computed according to the update equation, whereas for the membership degrees fixed functions depending only on the distances are prescribed. Thus the idea of the objective function is completely dropped in favour of a purely heuristic algorithm. This makes sense in the context of building fuzzy models with restricted types of fuzzy sets. However, the price to be paid is loosing the interpretability of the objectives the clustering has to meet in terms of the objective function as well as some proven convergence properties [1, 3, 9].

In this paper we want to maintain the idea of the objective function and generalise the transformation function g. It is obvious that g should be increasing (a higher membership degree leads to an increase in the objective function) and that we want $g(0) = 0$ and $g(1) = 1$. Finally, from the balance equation (7) we can see the following. For a minimum in the objective function we should have in any case (also guaranteed by the requirement that g is increasing) $g(u_{ij}) \leq g(u_{kj})$, if $d_{ij} \geq d_{kj}$ holds. In order to avoid crisp clustering, the balance equation tells us that g' must also be increasing.

There is also a technical constraint for g. The alternating optimisation scheme is already a price that we have to pay for the non-linearity of the objective function in the parameters to be optimised. In order to keep the computational complexity feasible, it is important that we can find the minimum of the objective function with respect to the considered parameter set (either the membership degrees or the cluster prototype parameters) in each single step of the alternating optimisation scheme directly.

In [11] it was proposed to use the quadratic transformation

$$g(u) = \alpha u^2 + (1-\alpha)u, \qquad (0 < \alpha < 1) \tag{8}$$

leading to

$$u_{ij} = \frac{1}{1-\beta}\left(\frac{1+(\hat{c}-1)\beta}{\sum_{k:u_{kj}\neq 0}\frac{d_{ij}}{d_{kj}}} - \beta\right) \tag{9}$$

as the update equation. This update equation requires to determine first which u_{kj} should be non-zero. $\hat{c}$ is the number of clusters for data object x_j to which x_j has a non-zero membership degree. The clusters with non-zero membership degrees are determined in the following way. (For a mathematical derivation see [11].) For a fixed j we can 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. (9) does not apply to these u_{ij}-values. Therefore, we have to find the smallest index i_0 to which (9) 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 (9) with $\hat{c} = c + 1 - i_0$.

At the first glance this seems to contradict the balance equation (7), since the balance equation is not satisfied for those j with $u_{ij} = 0$. However, we should mention that (6) is not valid for the case $u_{ij} = 0$. Therefore, the balance equation applies only to those u_{ij} with $u_{ij} \neq 0$.

The main structural difference between the standard transformation in fuzzy clustering $g_{\text{standard}}(u) = u^m$ (with $m > 1$) and the quadratic transformation g in (8) can be found in the derivative at zero, namely $g'_{\text{standard}}(0) = 0$, whereas $g'(0) = 1 - \alpha > 0$. This explains again the effect, why in standard fuzzy clustering zero membership degrees (nearly) never occur, whereas the quadratic transformation g in (8) allows zero membership degrees. Consider a data object x_j and assume without loss of generality that $d_{1j} \geq \ldots \geq d_{cj}$ holds. We know that at least the membership degree to the closest cluster c cannot be zero. Taking into account that g' is increasing, the highest possible value for the balance equation (6) for j is

$$d_{cj} \cdot g'(1) = d_{cj} \cdot (1+\alpha). \tag{10}$$

When we consider a cluster i that is further away from x_j than the closest cluster c, the smallest possible value for the balance equation is

$$d_{ij} \cdot g'(0) = d_{ij} \cdot (1-\alpha). \tag{11}$$

If (11) yields already a larger value than (10), we can immediately conclude that $u_{ij} = 0$ must hold. In other words,

$$\frac{d_{cj}}{d_{ij}} < \frac{g'(0)}{g'(1)} = \frac{1-\alpha}{1+\alpha} \tag{12}$$

implies $u_{ij} = 0$.

As an example, let us choose $\alpha = 0.5$. This means that the right-hand side of (12) is $1/3$. If the closest cluster to data object x_j is cluster c with distance d_{cj}, then x_j will have zero membership degree to any other cluster whose distance is at least three times as big as d_{cj}. Therefore, data objects that are close to one cluster will not influence the prototypes of clusters far away from it. In the case of noise clustering data that are far away from all clusters (and therefore relatively "close" to the noise cluster), will have no influence on the proper clusters.

When we insert the transformation (8) into the objective function

$$\begin{aligned} f &= \sum_{i=1}^{c}\sum_{j=1}^{n} g(u_{ij})d_{ij} \\ &= \alpha\sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^2 d_{ij} \;+\; (1-\alpha)\sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}d_{ij} \\ &= \alpha f_{\mathrm{FCM},m=2} \;+\; (1-\alpha)f_{\mathrm{HCM}}, \end{aligned}$$

we can see that this objective function represents a convex combination of the FCM objective function with fuzzifier $m = 2$ and the HCM objective function. In this sense, this approach tries to combine the advantages of fuzzy and crisp clustering and to avoid their disadvantages.

In [12] an exponential transformation

$$g : [0,1] \to [0,1], \qquad u \mapsto \frac{1}{e^{\alpha} - 1}\left(e^{\alpha u} - 1\right) \tag{13}$$

was proposed, leading to the update equation

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

As in (9) $\hat{c}$ is the number of clusters for which data object x_j has non-zero membership degrees. The clusters to which x_j has zero membership degree are determined analogously as in the case of (9).

When we compute the characteristic value (12) for the transformation (13), we obtain

$$\frac{g'(0)}{g'(1)} \;=\; e^{-\alpha}.$$

Finally, we present a new approach to fuzzy clustering that uses a piecewise linear transformation that exploits the balance equation directly. We consider a piecewise linear transformation of the structure shown in figure 6. We assume that the horizontal axis is divided into T ($T = 3$ in figure 6) intervals of equal length: $0 = u_0 < u_1 < \ldots < u_T = 1$ with $u_{t+1} - u_t = 1/T$. On each of these intervals the transformation is linear. On the interval $[u_t, u_{t+1}]$ the transformation is a line segment with $g(u_t) = g_t$ and $g(u_{t+1}) = g_{t+1}$. We require that g' is increasing where it is defined, i.e. everywhere, except at $u_0, \ldots, u_T$. Therefore, the line segments must become steeper from left to right, i.e.

$$g_t - g_{t-1} < g_{t+1} - g_t. \tag{14}$$

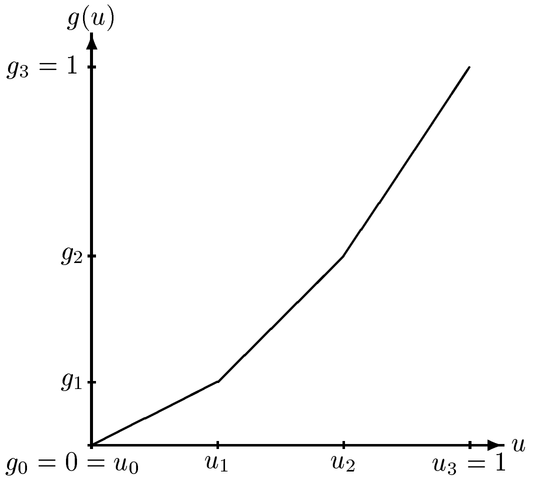


Figure 6: A piecewise linear transformation

Instead of taking derivatives to obtain the update equations for the u_{ij} when using the piecewise linear transformation function, we exploit the idea of using the balance equation at the end of section 3. Since g' is not continuous, the balance equation cannot be satisfied exactly. A simple heuristic strategy to implement the balancing idea would be the following. We consider a data object x_j and sort the distances in decreasing order: $d_{1j} \geq \ldots \geq d_{cj}$. We start with $u_{cj} = 1$ and $u_{1j} = \ldots = u_{c-1,j} = 0$. Now we compare the values from the balance equation, i.e.

$$\frac{g_T - g_{T-1}}{1/T} d_{cj} \qquad \text{and} \qquad \frac{g_1 - g_0}{1/T} d_{ij}.$$

When we decrease u_{cj} by $1/T$ and increase u_{ij} by $1/T$ instead, the objective function will be changed by the value

$$\Delta f \;=\; (g_1 - g_0)d_{ij} - (g_T - g_{T-1})d_{cj}.$$

This means, if $(g_T - g_{T-1})d_{cj} > (g_1 - g_0)d_{ij}$ holds, then Δf will become negative and the objective function will be decreased. In this case we would decrease u_{cj} and increase u_{ij} by $1/T$.

In principle, we could continue this balancing scheme, until the value of the objective function can no longer be decreased. However, this is not very efficient from the computational point of view. Instead, we apply the following scheme.

For a data object x_j we first set all values $u_{ij} = 0$. We then carry out T steps. In each step one of the u_{ij}-values will be increased by $1/T$. (Some of the u_{ij}-values might be increased more than once.) Assume we have already carried out t steps ($t = 0$ in the beginning). Assume u_{ij} has reached the value $u_s \in \{0, 1/T, 2/T, \ldots, 1\}$, after we had carried out the t steps. Increasing u_{ij} from u_s to

$u_{s+1} = u_s + 1/T$ leads to an increase of the value of the objective function of

$$\Delta f \;=\; (g_{s+1} - g_s) d_{ij}. \tag{15}$$

We now increase this u_{ij}-value ($i = 1, \ldots, c$) by $1/T$ for which (15) yields the smallest value and continue with the next step in the same way until we have finished all T steps.

We can also show that this procedure minimizes the objective function, when we fix the d_{ij} (the cluster centres in case of FCM). Consider a data object x_j. Note that we can treat the data objects independently, when we update the membership degrees. Since our membership transformation is piecewise linear, for a minimum of the objective function we should choose the membership degrees from the set $\{0, 1/T, 2/T, \ldots, 1\}$. Otherwise applying our trade-off concept, we could further reduce the value of the objective function. We prove by induction over t that in each step

$$\sum_{i=1}^{c} g(u_{ij}) d_{ij} \tag{16}$$

(with fixed distances d_{ij}) is minimized under the constraint

$$\sum_{i=1}^{c} u_{ij} \;=\; \frac{t}{T} \tag{17}$$

and, of course, requiring $u_{ij} \in \{0, 1/T, 2/T, \ldots, 1\}$. For $t = 0$ this is obviously true. For the induction step, let us assume as the induction hypothesis that (16) is minimized by our procedure in all steps including step t. We now have to show in the induction step that this also holds for step $(t+1)$. Let $u_{ij}^{(t)}$ denote the membership values we obtain after step t of our procedure. In our procedure only one of the values $u_{ij}^{(t)}$ is changed (increased by $1/T$), when we go from t to $(t+1)$. Without loss of generality, let us assume that $u_{1j}^{(t+1)} = u_{1j}^{(t)} + 1/T$ and $u_{ij}^{(t+1)} = u_{ij}^{(t)}$ for $i > 1$.

Now assume that we do not minimize (16) in step $(t+1)$ anymore, i.e. there is a configuration $\tilde{u}_{ij}$ satisfying the constraint (17) with

$$\sum_{i=1}^{c} g(\tilde{u}_{ij}) d_{ij} \;<\; \sum_{i=1}^{c} g\left(u_{ij}^{(t+1)}\right) d_{ij}. \tag{18}$$

In this case we must have $g(\tilde{u}_{1j}) < g\left(u_{1j}^{(t+1)}\right)$. Otherwise, according to (14) we obtain

$$\begin{aligned} g\left(\tilde{u}_{1j} - \frac{1}{T}\right) d_{1j} + \sum_{i=2}^{c} g(\tilde{u}_{ij}) d_{ij} \;&<\; g\left(u_{1j}^{(t+1)} - \frac{1}{T}\right) d_{1j} + \sum_{i=2}^{c} g\left(u_{ij}^{(t+1)}\right) d_{ij} \\ &=\; \sum_{i=1}^{c} g(u_{ij}^{(t)}) d_{ij}, \end{aligned}$$

saying that we did not minimize (16) in step t which is a contradiction to the induction hypothesis. Because of $g(\tilde{u}_{ij}) < g\left(u_{ij}^{(t+1)}\right)$ and since the $\tilde{u}_{ij}$ and the $u_{ij}^{(t+1)}$ both must obey (17), there must exist an index $i \in \{2, \ldots, c\}$ with $\tilde{u}_{ij} > u_{ij}^{(t+1)}$. Without loss of generality let us assume $\tilde{u}_{2j} > u_{2j}^{(t+1)}$, at least by $1/T$, since $\tilde{u}_{2j}, u_{2j}^{(t+1)} \in \{0, 1/T, 2/T, \ldots, 1\}$. Taking (17) and our iterative procedure into account, we have

$$\begin{aligned}\left(g\left(\tilde{u}_{2j}\right) - g\left(\tilde{u}_{2j} - \frac{1}{T}\right)\right) d_{2j} &\geq \left(g\left(u_{2j}^{(t)} + \frac{1}{T}\right) - g\left(u_{2j}^{(t)}\right)\right) d_{2j} \\ &\geq \left(g\left(u_{1j}^{(t)} + \frac{1}{T}\right) - g\left(u_{1j}^{(t)}\right)\right) d_{1j}.\end{aligned}$$

With these inequalities and our induction hypothesis implying

$$\sum_{i=1}^{c} g\left(u_{ij}^{(t)}\right) d_{ij} \leq g\left(\tilde{u}_{2j} - \frac{1}{T}\right) d_{2j} + \sum_{i=1, i \neq 2}^{c} g(\tilde{u}_{ij}) d_{ij}$$

we finally obtain

$$\begin{aligned}\sum_{i=1}^{c} g\left(u_{ij}^{(t+1)}\right) d_{ij} &= \left(g\left(u_{1j}^{(t)} + \frac{1}{T}\right) - g\left(u_{1j}^{(t)})\right)\right) d_{1j} + \sum_{i=1}^{c} g\left(u_{ij}^{(t)}\right) d_{ij} \\ &\leq \left(g\left(u_{2j}^{(t)} + \frac{1}{T}\right) - g\left(u_{2j}^{(t)}\right)\right) d_{2j} + \sum_{i=1}^{c} g\left(u_{ij}^{(t)}\right) d_{ij} \\ &\leq \left(g\left(\tilde{u}_{2j}\right) - g\left(\tilde{u}_{2j} - \frac{1}{T}\right)\right) d_{2j} + \sum_{i=1}^{c} g\left(u_{ij}^{(t)}\right) d_{ij} \\ &\leq g(\tilde{u}_{2j}) d_{2j} + \sum_{i=1, i \neq 2}^{c} g(\tilde{u}_{ij}) d_{ij} \\ &= \sum_{i=1}^{c} g(\tilde{u}_{ij}) d_{ij}\end{aligned}$$

which contradicts (18). This completes the induction proof.

Thus we can guarantee for convergence of our algorithm for the part of the alternating optimization scheme, when membership degrees are updated. At least for algorithms like FCM and the Gustafson-Kessel algorithm [8] convergence can also be guaranteed for the update scheme of the cluster prototypes [1, 9].

When we use the piecewise linear transformation, we will not obtain arbitrary membership degrees between zero and one, but only the values $0, 1/T, 2/T, \ldots, 1$.

With the piecewise linear transformation we can directly adjust how much a cluster further away from a data object should be influenced by this data object in comparison to a closer cluster. We can also avoid zero-membership degrees as they occur in standard fuzzy clustering, if this is desired. We simply have to choose

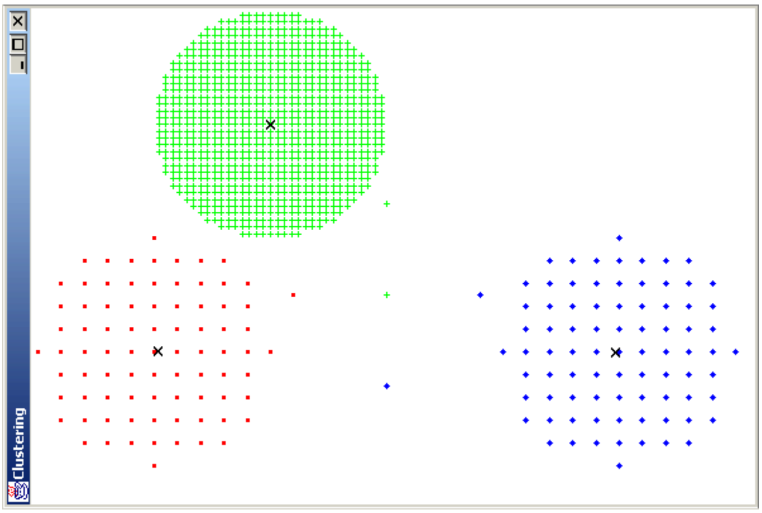


Figure 7: Clustering result using transformations

$g_1 = 0$, so that the membership degrees will always be at least $1/T$ (unless we have more than T clusters). HCM can be viewed as a special case of our balancing algorithm with $T = 1$.

Using a piecewise linear transformation has the advantage that various properties of the clustering algorithm can be controlled at the time. We can control, when zero membership degrees should occur (for which relative distances), and we can also adjust how strong clusters should overlap. The first property is controlled by the form of the transformation near zero, the second by the form of the transformation near one. Although the approach using a piecewise linear transformation is computationally less efficient than the ones with quadratic or exponential transformations, we gain more flexibility. The quadratic and the exponential transformation both have only one parameter to control zero membership degrees and cluster overlap at the same time. Using the piecewise linear transformation, we can control both properties almost independently. However, in order to keep the number of parameters small and the computation feasible, T should be chosen small. (We recommend $T < 10$.) In order to control zero membership degrees and cluster overlap at the same time, $T = 3$ is already sufficient and should work in most of the applications.

It should be mentioned that our fuzzy clustering algorithms with the quadratic, exponential as well as the piecewise linear transformation can cope with the data set shown in figure 5 and find the expected (correct) cluster centres as shown in figure 7.

5 Conclusions

We have discussed some advantages and disadvantages of probabilistic fuzzy (including noise) clustering. It seems that the non-zero membership degree property of probabilistic clustering has a smoothing effect on undesired local minima of the objective function as we have illustrated in section 2. However, the same property causes also problems in fuzzy clustering as we have shown in section 3. These bad effects can be avoided by the modified transformations replacing the fuzzifier that we have discussed in section 4. Nevertheless, there seems to be a certain trade-off between the good and bad effects of fuzzy clustering. We have seen that our approach using the quadratic transformation can be viewed as a combination of HCM and FCM. In this sense we have to find a compromise between the smoothing effect on undesired local minima and the bad effects of non-zero membership degrees. In order to increase the speed of convergence and to avoid undesired local minima, we recommend to initialise our algorithm with the result from a standard probabilistic (or better noise) clustering analysis.

References

[1] J.C. Bezdek: A Convergence Theorem for the Fuzzy ISODATA Clustering Algorithm. IEEE Trans. Pattern Analysis and Machine Intelligence 2 (1980), 1-8

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

[3] J.C. Bezdek, R.H. Hathaway, M.J. Sabin, W.T. Tucker: Convergence Theory for Fuzzy c-Menas: Counterexamples and Repairs. IEEE Trans. Systems, Man, and Cybernetics 17 (1987), 873-877

[4] J.C. Bezdek, J. Keller, R. Krishnapuram, N.R. Pal: Fuzzy Models and Algorithms for Pattern Recognition and Image Processing. Kluwer, Boston (1999)

[5] R.N. Davé: Characterization and Detection of Noise in Clustering. Pattern Recognition Letters 12 (1991) 657-664

[6] R. Duda, P. Hart: Pattern Classification and Scene Analysis. Wiley, New York (1973)

[7] A. Flores-Sintas, J.M. Cadenas, F. Martin: Membership Functions in the Fuzzy c-Means Algorithm. Fuzzy Sets and Systems 101 (1999), 49-58.

[8] E.E. Gustafson and W.C. Kessel: Fuzzy Clustering with a Fuzzy Covariance Matrix. In Proc. 18th IEEE Conference on Decision and Control (IEEE CDC, San Diego), IEEE Press, Piscataway (1979), 761-766.

[9] F. Höppner, F. Klawonn: A Contribution to Convergence Theory of Fuzzy c-Means and its Derivatives. IEEE Trans. on Fuzzy Systems 11 (2003), 682-694

[10] F. Höppner, F. Klawonn, R. Kruse, T. Runkler: Fuzzy Cluster Analysis. Wiley, Chichester (1999)

[11] F. Klawonn, F. Höppner: What is Fuzzy About Fuzzy Clustering? Understanding and Improving the Concept of the Fuzzifier. In: M.R. Berthold, H.-J. Lenz, E. Bradley, R. Kruse, C. Borgelt (eds.): Advances in Intelligent Data Analysis V. Springer, Berlin (2003), 254-264

[12] F. Klawonn, F. Höppner: An Alternative Approach to the Fuzzifier in Fuzzy Clustering to Obtain Better Clustering Results. In: Proc. 3rd Eusflat Conference, Zittau (2003), 730-734

[13] R. Krishnapuram, J. Keller: A Possibilistic Approach to Clustering. IEEE Trans. on Fuzzy Systems 1 (1993) 98-110

[14] T.A. Runkler, J.C. Bezdek: Alternating Cluster Estimation: A New Tool for Clustering and Function Approximation. IEEE Trans. on Fuzzy Systems 7 (1999), 377-393

[15] H. Timm, C. Borgelt, R. Kruse: A Modification to Improve Possibilistic Cluster Analysis. IEEE Intern. Conf. on Fuzzy Systems, Honululu (2002)