Understanding and Controlling the Membership Degrees in Fuzzy Clustering

Frank Klawonn

Department of Computer Science
University of Applied Sciences Braunschweig/Wolfenbuettel
D-38302 Wolfenbuettel, Germany

Abstract. Fuzzy cluster analysis uses membership degrees to assign data objects to clusters in order to better handle ambiguous data that share properties of different clusters. However, the introduction of membership degrees requires a new parameter called fuzzifier. In this paper the good and bad effects of the fuzzifier on the clustering results are analysed and based on these considerations a more general approach to fuzzy clustering is proposed, providing better control on the membership degrees and their influence in fuzzy cluster analysis.

1 Introduction

A simple and common popular approach to cluster analysis is the so-called k-means clustering algorithm (see for instance Duda and Hart (1973)). In this approach each cluster is represented by a prototypical object that corresponds to the cluster centre. A data object is assigned to the cluster for which the distance of the prototype to the data object is smallest. In order to model partly overlapping clusters, the concept of membership degrees was proposed by Bezdek (1973) and Dunn (1974). Viewing k-means clustering and its fuzzy variant as an objective function-based clustering technique, it is necessary to introduce a new parameter, called fuzzifier. The behaviour and the properties of the clustering scheme differ significantly from the classical k-means one. A detailed analysis of the fuzzifier, its properties, its positive and negative effects leads to a more general approach to fuzzy clustering with a better understanding as well as better control of the clustering parameters and properties.

The paper is organized as follows. After a brief review of fuzzy clustering in section 2, section 3 provides a more detailed analysis and understanding of the fuzzifier concept, discussing also the question whether fuzzy clustering is more robust than crisp clustering. In section 4 various approaches are proposed that generalize the fuzzifier concept in order to have a better control on the properties of the clustering algorithm. Future perspectives are discussed in the conclusions.

2 Fuzzy cluster analysis

The c-means[1] clustering algorithm is designed to partition a data set $X = \{x_1, \ldots, x_n\} \subset R^p$ into c clusters. 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 $v_i \in R^p$. 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 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 ($u_{ij} = 0$). $d_{ij} = \| x_j - v_i \|^2$ is the squared Euclidean distance between data vector x_j and cluster prototype v_i. The parameters to be optimized are the cluster prototypes v_i, hidden in the distances d_{ij}, and the assignments u_{ij} to the clusters. Since there is no direct solution to this optimization problem, the above described strategy tries to minimize the objective function by alternatingly optimizing either the cluster prototypes or the assignments, while the other parameter set is considered to be fixed.

It should be noted that by replacing the Euclidean distance by other distance measures and enriching the cluster prototypes by further parameters, other shapes than just the spherical clusters as in standard c-means clustering can be discovered. Clusters might be ellipsoidal, linear manifolds, quadrics or even differ in volume (Keller and Klawonn (2003)). Since this paper is concerned with the assignment of the data objects to clusters, we refer to the literature (for instance Höppner et al. (1999)) for an overview. All our considerations are more or less independent of the chosen distance function d_{ij}.

The generalization from crisp assignments $u_{ij} \in \{0, 1\}$ to membership degrees $u_{ij} \in [0, 1]$ seems to be straight forward, by simply considering the latter relaxed constraint for the objective function. However, even when arbitrary values between zero and one are allowed for the assignment of the data objects to the clusters, it is easy to prove that a minimum of the objective

[1] In fuzzy cluster analysis, c is chosen to denote the number of clusters. In order to be coherent in notation, we therefore write c-means instead of k-means clustering.

function (1) can only be obtained, if the membership degrees are chosen in a crisp way, i.e. $u_{ij} \in \{0, 1\}$. The reason for this is quite obvious. Not assigning the full weight u_{ij} of a data object x_j to the closest cluster i, but instead raising the weight u_{kj} to a cluster with a larger distance, will definitely increase the value of the objective function. Therefore, for fuzzy clustering the objective function was modified in the following form, introducing a so-called fuzzifier $m > 1$:

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

Note that the fuzzifier m does not have any effects, when we use hard clustering. The fuzzifier $m > 1$ is not subject to the optimization 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 also 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.

The objective function (3) is also optimized by an alternating optimization scheme. It can be shown that the membership degrees have to be chosen as

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

unless there exists a cluster i with zero distance d_{ij} to x_j. In this case $u_{ij} = 1$ and $u_{kj} = 0$ for $i \neq k$ is chosen. If d_{ij} is the squared Euclidean distance, then the cluster centres v_i are computed as the weighted mean

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

3 Properties of the fuzzifier

The fuzzifier m controls, how much clusters may overlap. For $m \to 1$ the membership degrees tend to the values 0 and 1, i.e. fuzzy clustering is turned into crisp clustering. For $m \to \infty$ clusters become completely merged because of $u_{ij} \to \frac{1}{c}$.

In addition to be able to better handle ambiguous data, it seems that fuzzy c-means clustering is more robust than standard crisp c-means. Although there is no proof for this hypothesis, the use of membership degrees might be able to eliminate undesired local minima in the objective function and can therefore prevent fuzzy clustering from converging to counter-intuitive results. Figure 1 shows the objective functions of c-means and fuzzy c-means clustering for a simple one-dimensional data set. The data set consists of

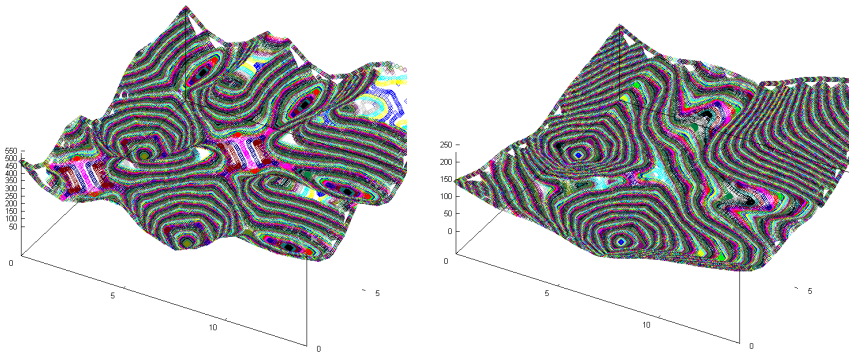

Fig. 1. Objective functions for crisp (left) and fuzzy (right) clustering

two clusters centred around 0 and 5. There is also a cluster with very few data around 10. The clustering was carried out using noise clustering (Davé (1991)). This means that in addition to the two clusters for which the prototypes must be computed there is a third noise cluster that has no specific prototype, but a fixed large distance to all data. The noise cluster is supposed to collect those data that are far away from all other clusters, in our case the few data around 10. In figure 1 the x- and the y-axis refer to the location of the two cluster prototypes. The membership degrees are assumed to be chosen according to (4) for fuzzy clustering and for crisp clustering, the data objects are assigned to the closest cluster (including the noise cluster). The objective function for fuzzy clustering on the right hand side has two local minima, both representing correct clustering results. The difference between them is that the first and second cluster prototype are exchanged. The objective function for crisp clustering has – in addition to the two "correct" local minima – four more undesired local minima. For a detailed discussion of this problem we refer to Klawonn (2004).

Another explanation for the higher robustness of fuzzy clustering is that a bad initialisation is more difficult to overcome for crisp clustering. In order to illustrate this effect we consider the artificial data set in figure 2. When (crisp) c-means is initialized by random cluster centres as they are indicated by the three squares, the left prototype will grab immediately the two clusters on the left hand side, while the other two prototypes have to share the third cluster. They will never obtain any information about the existence of the other two clusters. The situation is different for fuzzy c-means. Although all data from the two clusters on the left hand side are closest to the left initial prototype, they will still have a non-zero membership degree to the other clusters according to equation (4). Therefore, there is a higher chance that one of the prototypes on the right hand side will be attracted by one of the two clusters on the left hand side. However, although fuzzy clustering benefits in this case from the non-zero membership degrees, they can have

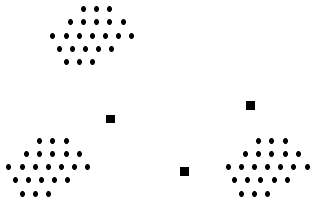

Fig. 2. A bad initialisation for c-means

bad effects in other cases. First of all, it is counter-intuitive to have non-zero membership degrees, no matter how far away a data object lies from a cluster prototype and how well it might be covered by another prototype. Secondly, when clusters have different data densities, clusters with higher densities tend to influence or even completely attract other cluster prototypes than the one that is closest as well.

4 Alternatives to the fuzzifier

Viewing the fuzzifier in fuzzy clustering from a more general point of view, its main effect is a transformation of the membership degrees. Instead of using the terms $u_{ij}d_{ij}$ as in the objective function (1) for c-means, fuzzy clustering (3) replaces this by $g(u_{ij})d_{ij}$ where $g(u) = u^m$. It is an obvious question, whether this type of transformation g is the only reasonable one or whether there are better alternatives. In order to better understand the role of the transformation g, we follow Klawonn and Höppner (2003a) and consider the objective function

$$f \;=\; \sum_{i=1}^{c}\sum_{j=1}^{n} g(u_{ij})d_{ij} \tag{6}$$

under the constraints (2) that we want to minimize w.r.t. the values u_{ij}, considering the distances d_{ij} to be 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

$$\frac{\partial L}{\partial u_{ij}} \;=\; g'(u_{ij})d_{ij} - \lambda_j. \tag{7}$$

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

$$g'(u_{ij}) \cdot d_{ij} \;=\; g'(u_{kj}) \cdot d_{kj} \tag{8}$$

for all i, k at a minimum. This actually means that these products must be balanced during the minimization process, unless at least one of the two membership degrees is zero or one. Equation (8) also explains why it is necessary to introduce the fuzzifier and why zero membership degrees (nearly) never occur. When we simply use the identity $g(u) = u$ as the transformation, i.e. we consider the objective function (1), then it is obvious that (8) cannot be achieved, since $g'(u) = 1$ is constant. On the other hand, when we use $g(u) = u^m$ with $m > 1$, we have $g'(0) = 0$ and $g'(1) = m > 0$. Therefore, in order to balance the two products in (8), no matter how large d_{ik} and how small d_{ij} is, u_{kj} must be chosen greater than zero and u_{ij} smaller than one. When we replace $g(u) = u^m$ by another transform g with $g'(0) > 0$, this will definitely yield a zero membership degree for a cluster k, if $d_{ij}/d_{kj} < g'(0)/g'(1)$ holds. Of course, it is not possible to choose arbitrary functions $g : [0, 1] \to [0, 1]$. It is obvious that g should be increasing and that we want $g(0) = 0$ and $g(1) = 1$. The above argument also requires that g should be differentiable. Equation (8) will only let clusters with a larger distance to a data object than the closest cluster participate in the membership degree if $g'(u) < g'(\tilde{u})$ for $u < \tilde{u}$. This means g' should also be increasing. Another important aspect is that we are still able to derive an analytical solution for the membership degrees, when we want to minimize the objective function (6) while fixing the cluster prototypes (and therefore the distances d_{ij}). Without an analytical solution, a numerical solution, i.e. an iterative scheme would be needed within the alternating optimization leading to high computational costs. Klawonn and Höppner (2003a/b) proposed a quadratic transform

$$g_\alpha(u) \;=\; \alpha u^2 + (1-\alpha)u \qquad (0 \leq \alpha \leq 1) \tag{9}$$

and an exponential transform

$$g_\alpha(u) \;=\; \frac{1}{e^\alpha - 1}\,(e^{\alpha u} - 1) \qquad (0 < \alpha). \tag{10}$$

The objective function using the transformation (9) represents a convex combination of standard crisp c-means and fuzzy c-means clustering with a fuzzifier $m = 2$. When using these transforms, the update equations for the alternating optimization scheme have to be altered. For the cluster prototypes (5) the terms u_{ij}^m have to be replaced by $g_\alpha(u_{ij})$. The update equations for the membership degrees are

$$u_{ij} \;=\; \frac{1}{1-\beta}\left(\frac{1+(\hat{c}-1)\beta}{\sum_{k:u_{kj}\neq 0}\frac{d_{ij}}{d_{kj}}} - \beta\right) \qquad \left(\text{where } \beta = \frac{1-\alpha}{1+\alpha}\right) \tag{11}$$

and

$$u_{ij} \;=\; \frac{1}{\alpha\hat{c}}\left(\alpha + \sum_{k:u_{kj}\neq 0} \ln\left(\frac{d_{kj}}{d_{ij}}\right)\right), \tag{12}$$

respectively. $\hat{c}$ is the number of clusters with zero membership degree for data object x_j. These clusters with zero membership degree are determined in the following way. For a fixed data object x_j the distances d_{ij} are sorted 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. The corresponding update equation does not apply to these u_{ij}-values. Therefore, we have to find the smallest index i_0 to which (11), respectively (12), 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 (11), respectively (12), with $\hat{c} = c + 1 - i_0$. Note that updating the membership degrees requires additional sorting. However, the sorting has to be carried out for a relatively small number of elements, namely as many elements as there are clusters, so that the additional computational costs are acceptable.

Klawonn (2004) proposed to drop the differentiability of the transformation g completely and to consider a piecewise linear transformation g as it is shown in figure 3. This gives more freedom to control the behaviour of the membership degrees than just by one parameter α. With this kind of

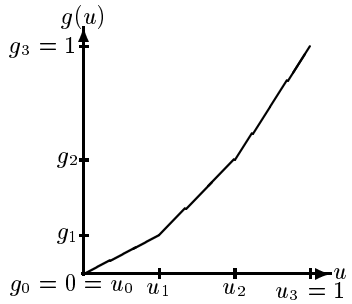

Fig. 3. A piecewise linear transformation

transformation, the update equation for membership degrees can no longer be determined by taking derivatives. Klawonn (2004) describes an efficient update scheme for such piecewise linear transformations with guaranteed convergence. A piecewise linear transformation will lead to discrete membership degrees in the sense that only the membership degrees corresponding to the bends in the curve will occur. Although such a piecewise linear transformation is not differentiable, it still satisfies the condition that its derivative exists at least almost everywhere and is non-decreasing. We can even give up this monotonicity condition for the derivative. This means that membership degrees in parts where the transformation becomes flatter will not be assigned to clusters. In this way, making the curve flatter around 0.5, we can avoid ambiguous membership degrees forcing them to tend more to either zero or one.

5 Conclusions

Understanding the fuzzifier in fuzzy clustering as a specific type of transformation opens the door to new approaches to fuzzy clustering. The undesired effect of non-zero membership degrees, no matter how far away data objects might be from a cluster, can be avoided in this way. This also enables fuzzy clustering to cope with clusters of different densities. The proposed transforms allow a specific adjustment of the properties of the fuzzy clustering algorithm like: When should a data object have a non-zero membership degree? Are completely ambiguous data acceptable? In most cases this can be achieved by a piecewise linear transformation with only three or four segments. Future work will be devoted to algorithms whose membership transformation is changed over time or might even be adaptive. For instance, as we have already mentioned in section 3, standard fuzzy clustering can easier overcome a bad initialisation, because all data objects have at least a small influence on all clusters, no matter how large the distance is. However, this property is not desired for the final clustering result. Therefore, it seems advisable, to start the clustering with a transformation similar to the one used in standard fuzzy clustering and then modify it, so that non-zero membership degrees become less probable.

References

BEZDEK, J.C. (1973): Fuzzy Mathematics in Pattern Classification. Ph.D. thesis, Appl. Math. Center, Cornell University, Ithaca.

DAVÉ, R.N. (1991): Characterization and Detection of Noise in Clustering. *Pattern Recognition Letters, 12, 657–664.*

DUDA, R. and HART, P. (1973): *Pattern Classification and Scene Analysis.* Wiley, New York.

DUNN, J.C. (1974): A Fuzzy Relative of the ISODATA Process and its Use in Detecting Compact, Well Separated Clusters. *Journal of Cybernetics, 3, 95–104.*

HÖPPNER, F., KLAWONN, F., KRUSE, R. and RUNKLER, T. (1999): *Fuzzy Cluster Analysis.* Wiley, Chichester.

KELLER, A. and KLAWONN , F. (2003): Adaptation of Cluster Sizes in Objective Function Based Fuzzy Clustering. In: C.T. Leondes (ed.): *Intelligent Systems: Technology and Applications vol. IV: Database and Learning Systems.* CRC Press, Boca Raton, 181–199.

KLAWONN, F. (2004): Fuzzy Clustering: Insights and a New Approach. *Mathware and Soft Computing, 11, 125–142.*

KLAWONN, F., HÖPPNER, F. (2003a): 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, 254–264.

KLAWONN, F., HÖPPNER, F. (2003b): An Alternative Approach to the Fuzzifier in Fuzzy Clustering to Obtain Better Clustering Results. In: *Proc. 3rd Eusflat Conference.* Zittau, 730–734.