An Alternative Approach to the Fuzzifier in Fuzzy Clustering to Obtain Better Clustering Results

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

Frank Höppner
Department of Economics
University of Applied Sciences BS/WF
Robert-Koch-Platz 10-14
D-38440 Wolfsburg
Germany
f.hoeppner@fh-wolfenbuettel.de

Abstract

The most common fuzzy clustering algorithms are based on the minimization of an objective function that evaluates (fuzzy) cluster partitions. The generalisation step from hard clustering to crisp clustering requires the introduction of an additional parameter, the so called fuzzifier. This fuzzifier does not only control, how much clusters may overlap, but has also some undesired consequences. For example, data have (almost) always non-zero membership degrees to all clusters, no matter how far they are away from a cluster. We propose a concept that generalizes the idea of the fuzzifier and solves the mentioned problems.

Keywords: Fuzzy clustering, noise clustering, fuzzifier.

1 Introduction

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 crisp 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 may 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 considerations and on a more general understanding of the fuzzifier, 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

Fuzzy clustering is suited for finding structures in data. A data set is divided into a set of clusters and – in contrast to hard clustering – a datum is not assigned to a unique cluster. In order to handle noisy and ambiguous data, membership degrees of the data to the clusters are computed. Most fuzzy clustering techniques are designed to optimise an object function with constraints.

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

$$f = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^{m} d_{ij} \quad \text{constrained by} \quad \sum_{i=1}^{c} u_{ij} = 1 \qquad (1)$$

and the constraints

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

that should be minimized.

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 [2, 5].

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 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.

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 [2, 5].) 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) w.r.t. the membership degrees is obtained at [1]

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

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) w.r.t. the cluster prototypes is obtained at [1]

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

when the membership degrees u_{ij} are considered to be fixed. The prototypes are still the cluster centres. The cluster prototypes are simply cluster weighted centres 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 [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 [7] 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 [8].

Another approach that emphasizes a probabilistic interpretation in fuzzy clustering is described in [4] where membership degrees as well as membership probabilities are used for the clustering. In this way, some of the problems of the standard FCM 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.

3 Effects of the Fuzzifier

The update equation (3) for the membership degrees derived from the objective function (1) can lead to undesired or counterintuitive results, because zero membership degrees never occur (except in the extremely

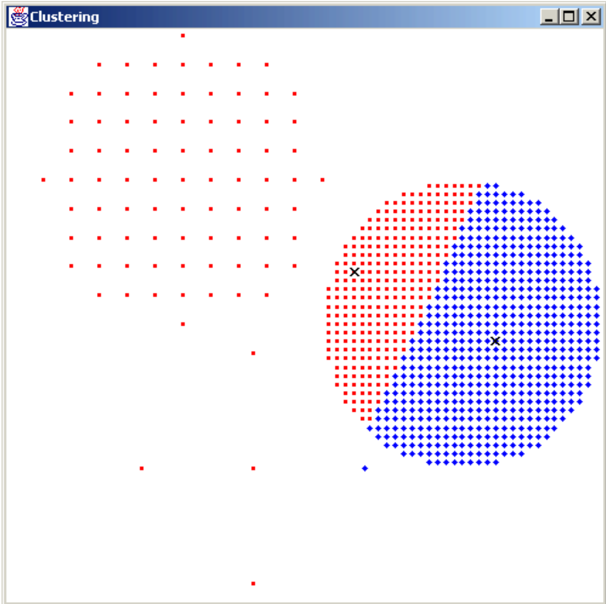

Figure 1: Clusters with varying density

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 non-zero membership degrees to all other clusters.

Figure 1 shows an undesired side-effect of the probabilistic fuzzy clustering approach. There are obviously two clusters. However, the right-hand cluster has a much higher data density than the other one. This single dense cluster attracts the other cluster prototype so that the prototype of the left-hand 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, 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.

4 Replacing the Fuzzifier Function

Viewing the objective function (1) from a more general point of view, the fuzzifier carries out a transformation

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

of the membership degrees. This transformation should be increasing and $t(0) = 0$ and $t(1) = 1$ should hold. This is definitely satisfied by the transformation $t(u) = u^m$. However, there might be better choices. Let us consider an arbitrary transformation t satisfying the afore mentioned conditions and let us take a closer look which effect the transformation t has on the objective function. Assume, we want to minimize the objective function

$$f = \sum_{i=1}^{c}\sum_{j=1}^{n} t(u_{ij})d_{ij} \qquad (5)$$

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

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

and the partial derivatives

$$\frac{\partial L}{\partial u_{ij}} = t'(u_{ij})d_{ij} - \lambda_j. \qquad (6)$$

At a minimum of the objective function the partial derivatives must be zero, i.e. $\lambda_j = t'(u_{ij})d_{ij}$. Since λ_j is independent of i, we must have

$$t'(u_{ij})d_{ij} = t'(u_{kj})d_{kj} \qquad (7)$$

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

Considering the standard transformation $t(u) = u^m$ used in fuzzy clustering, it yields very small values for the derivative near zero and even zero at $u = 0$. This is the reason, why zero membership degrees nearly never occur.

We therefore propose to use another transformation t that does not assume the value zero for the derivative at zero. In order not to end up with crisp membership degrees again, we should make sure that the derivative t' is increasing, but not starting at the value zero at zero.

When we consider a data vector x and the cluster i nearest to x and another cluster k further away from x, taking (7) into account, we can see that the quotient

$$\frac{t'(0)}{t'(1)} \tag{8}$$

indicates which value the quotient

$$\frac{d_{ij}}{d_{kj}}$$

must exceed, in order to yield a non-zero membership for the cluster k further away from x. The larger the value (8) is, the more will the clustering tend to prefer crisp clusters. Although (8) yields always zero for the standard fuzzy clustering with $t(u) = u^m$, we can still use a similar measure, when we replace the derivative at zero in (8) by a value of the derivative at $\varepsilon > 0$ near zero.

$$\frac{t'(\varepsilon)}{t'(1)}$$

gets larger for a smaller fuzzifier m, when we choose $t(u) = u^m$.

A more detailed analysis of this problem can be found in [6], where also an alternative quadratic transformation t is discussed. In this paper we propose to replace the transformation t by the following one:

$$t_\alpha : [0,1] \to [0,1], \qquad u \mapsto \frac{1}{e^\alpha - 1}\left(e^{\alpha u} - 1\right) \tag{9}$$

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

$$f = \sum_{i=1}^{c}\sum_{j=1}^{n} \frac{1}{e^\alpha - 1}\left(e^{\alpha u_{ij}} - 1\right) d_{ij} \tag{10}$$

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

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

and solving for u_{ij} we obtain

$$u_{ij} = \frac{1}{\alpha} \ln\left(\frac{\lambda_j (e^\alpha - 1)}{\alpha \cdot d_{ij}}\right) \tag{11}$$

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

$$\lambda_j = e^{\alpha/\hat{c}} \cdot \frac{1}{e^\alpha - 1} \cdot \alpha \cdot \prod_{k: u_{kj} \neq 0} \left(d_{kj}\right)^{1/\hat{c}}$$

where $\hat{c}$ is the number of clusters to which data vector x_j has non-zero membership degrees. Replacing λ_j in (11), we finally obtain the update equation for

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

We still have to determine which u_{ij} are zero. Since we want to minimize the objective function (10), we can make the following observation. If $u_{ij} = 0$ and $d_{ij} < d_{tj}$, then at a minimum of the objective function, we must have $u_{tj} = 0$ as well. Otherwise we could reduce the value by setting u_{tj} to zero and letting u_{ij} assume the original value of u_{tj}. This implies the following. For a fixed j we can sort the distances d_{ij} in decreasing order. Without loss of generality let us assume $d_{1j} \ge \ldots \ge d_{cj}$. 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. (12) does not apply to these u_{ij}-values. Therefore, we have to find the smallest index i_0 to which (12) is applicable, i.e. for which it yields a positive value. For $i < i_0$ we have $u_{ij} = 0$ and for $i \ge i_0$ the membership degree u_{ij} is computed according to (12) with $\hat{c} = c + 1 - i_0$.

When using our modified algorithm, the update equations for the cluster prototype remain the same – for instance, in the case of FCM cluster centres as in (4) – except that we have to replace u_{ij}^m by

$$t_\alpha(u_{ij}) = \frac{1}{e^\alpha - 1}\left(e^{\alpha u_{ij}} - 1\right).$$

In addition to using the slightly more complicated update equation (12) instead of (3), we also have to sort the distances to the cluster centres for each data vector in every iteration step. However, from the point of view of computational complexity, this does not lead to a significant decrease in performance, since we only have to sort as many distances as we have clusters each time. And the number of clusters is usually quite small so that we mainly have a linear increase of the computational complexity in the number of data. For reasons of efficiency, we also recommend to compute (12) in the following way:

$$u_{ij} = \frac{1}{\hat{c}} + \frac{1}{\alpha\hat{c}}\left(\sum_{k:u_{kj}\neq 0} \ln(d_{kj})\right) - \alpha\ln(d_{ij}). \quad (13)$$

For a fixed data vector x_j only the last term $\ln(d_{ij})$ varies for the clusters, the rest needs only to be computed once in each iteration step.

With this modified algorithm the data set in figure 1 is clustered correctly (for instance with $\alpha = 0.5$). 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.

The value $\alpha > 0$ in our approach has a similar effect as the fuzzifier m in standard fuzzy clustering: The smaller α is chosen, the crisper the fuzzy partition tends to be. This is also obvious from the quotient (8). We obtain

$$\frac{t'_\alpha(0)}{t'_\alpha(1)} = e^{-\alpha}.$$

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. As a future work, we will extend our approach to more general transformations t other than the exponential one proposed in this paper and the quadratic approach presented in [6].

References

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

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

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

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

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

[6] F. Klawonn, F. Höppner. What is Fuzzy About Fuzzy Clustering? Understanding and Improving the Concept of the Fuzzifier. In: Proceedings Intelligent Data Analysis 2003. Springer, Berlin (to appear).

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

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