Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT®

INTERNATIONAL JOURNAL OF APPROXIMATE REASONING

ELSEVIER International Journal of Approximate Reasoning 32 (2003) 85–102

www.elsevier.com/locate/ijar

Improved fuzzy partitions for fuzzy regression models ☆

Frank Höppner *, Frank Klawonn

Department of Computer Science, University of Applied Sciences, Braunschweig/Wolfenbüttel, Salzdahlumer Str. 46/48, D-38302 Wolfenbüttel, Germany

Received 1 January 2002; accepted 1 April 2002

Abstract

Fuzzy clustering algorithms like the popular fuzzy *c*-means algorithm (FCM) are frequently used to automatically divide up the data space into fuzzy granules. When the fuzzy clusters are used to derive membership functions for a fuzzy rule-based system, then the corresponding fuzzy sets should fulfill some requirements like boundedness of support or unimodality. Problems may also arise in the case, when the fuzzy partition induced by the clusters is intended as a basis for local function approximation. In this case, a local model (function) is assigned to each cluster. Taking the fuzziness of the partition into account, continuous transitions between the single local models can be obtained easily. However, unless the overlapping of the clusters is very small, the local models tend to mix and no local model is actually valid.

By rewarding crisp membership degrees, we modify the objective function used in fuzzy clustering and obtain different membership functions that better suit these purposes. We show that the modification can be interpreted as standard FCM using distances to the Voronoi cell of the cluster rather than using distances to the cluster prototypes. In consequence, the resulting partitions of the modified algorithm are much closer to those of the crisp original methods. The membership functions can be generalized to a fuzzified minimum function. We give some bounds on the approximation quality of this fuzzification.

We apply this modified fuzzy clustering approach to building fuzzy models of the Takagi–Sugeno (TS) type automatically from data.

☆ This work was supported by the Deutsche Forschungsgemeinschaft (DFG) under grant Kl 648/1-1.

* Corresponding author. Tel.: +49-1709414096; fax: +49-49218071843.
E-mail address: frank.hoeppner@ieee.org (F. Höppner).

0888-613X/02/$ - see front matter © 2002 Elsevier Science Inc. All rights reserved.
PII: S0888-613X(02)00078-6

Keywords: Fuzzy minimum function; Fuzzy clustering; Fuzzy regression; k-Means; Voronoi diagram; Takagi–Sugeno model

1. Introduction

When building fuzzy systems automatically from data, we are in need of procedures that automatically divide up the input space in fuzzy granules. These granules are the building blocks for the fuzzy rules. When modeling an input–output relationship, the membership functions of these rules play the same role as basis functions in conventional function approximation tasks. To keep interpretability we usually require that the fuzzy sets are specified in *local regions*, that is, the membership functions have bounded support or decay rapidly. If this requirement is not fulfilled, many rules must be applied and aggregated simultaneously, such that the final result becomes more difficult to grasp – one is not allowed to interpret a fuzzy system *rule by rule* any longer. A second requirement is that the fuzzy sets of the primitive linguistic values should be simple and unimodal. It would be counterintuitive if the membership of the linguistic term "young", which is high for "17 years", would be higher for "23 years" than for "21 years".

To gain such fuzzy granules clustering algorithms can be used. Especially fuzzy clustering algorithms seem well suited, because they provide the user with a fuzzy membership function which could be used directly for the linguistic terms. Unfortunately, the family of the fuzzy c-means (FCM) clustering algorithms [1] and derivatives produce membership functions that do not fulfill the above-mentioned requirements [10]. Fig. 1(c) shows an example for FCM membership functions for a partition of the real line with cluster representatives $c_1 = 1$, $c_2 = 3$ and $c_3 = 8$. We can observe that the support of the membership functions is unbounded for all clusters, in particular for the cluster whose center is located at $c_2 = 3$. While for $c_1 = 1$ and $c_3 = 8$ one allows even in the context of fuzzy systems for an unbounded support if $x < 1$ and $x > 8$, respectively, but at least the membership function for $c_2 = 3$ should be defined locally. Furthermore, we can observe that the membership degree for the cluster at $c_1 = 1$ increases near 5, the FCM membership functions are not unimodal. These undesired properties can be reduced by tuning a parameter of the FCM algorithm, the so-called fuzzifier, however, then we also decrease the fuzziness of the partition and finally end up with crisp indicator functions as shown in Fig. 1(a). The problem of unimodality can be solved by using possibilistic memberships [4], but the possibilistic c-means is not a partitional but a mode-seeking algorithm. In [10] the objective function has been completely abandoned to allow user-defined membership functions, thereby also loosing the partitional property. For further litera-

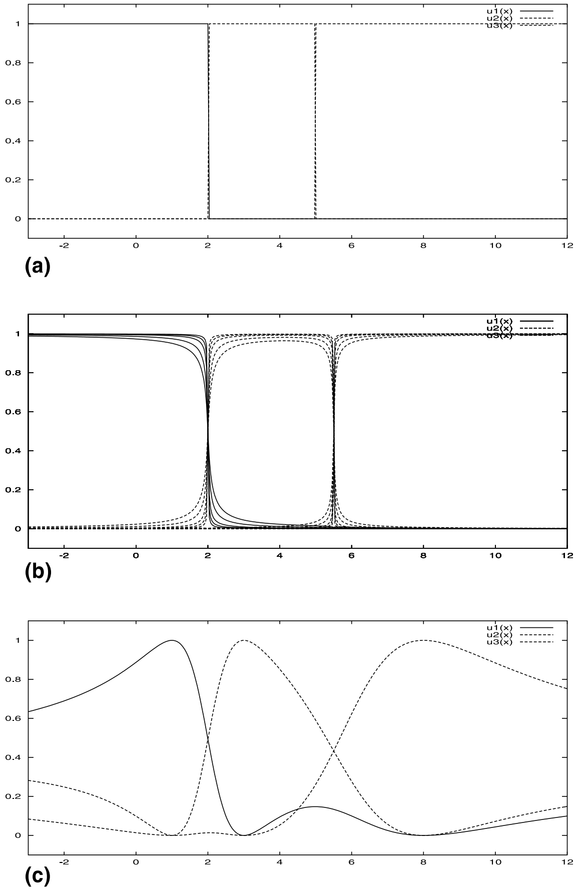

Fig. 1. Different kinds of membership functions: (a) indicator functions of crisp partition; (b) intuitively fuzzified partitions of (a); (c) FCM membership functions ($m = 2.0$).

ture about different aspects of interpretability in fuzzy systems, see for instance [2].

In this paper, we investigate alternative approaches to influence the fuzziness of the final partition. We consider a "reward" term for membership degrees near 0 and 1 in order to force a more crisp assignment in Section 3. If we choose an (in some sense) maximal reward, we arrive at fuzzy membership functions which are identical to those that we would obtain by using a (scaled) distance to the Voronoi cell that represents the cluster instead of the Euclidean distance to the clusters center, as we will see in Section 4. Furthermore, the membership functions – as a whole – can be interpreted as a *fuzzified minimum* *function* [7], for which we give an estimation of the error we make when substituting a crisp minimum function by its fuzzy version (Section 5).

2. Objective function-based fuzzy clustering

In this section, we briefly review the fuzzy *c*-means [1] and related algorithms, for a thorough overview of objective function-based fuzzy clustering see [9], for instance. Let us denote the membership degree of data object $x_j \in X,\ j \in \{1,\dots,n\}$, to cluster $p_i \in P,\ i \in \{1,\dots,c\}$, by $u_{i,j} \in [0,1]$. Denoting the distance of a data object x_j to a cluster determined by the prototype p_i by $d(x_j,p_i)$, we minimize the objective function

$$J_m(P,U;X) = \sum_{j=1}^{n}\sum_{i=1}^{c} u_{i,j}^m d^2(x_j,p_i), \tag{1}$$

where the so-called "fuzzifier" m is chosen in advance and influences the fuzziness of the final partition (crisp as $m \to 1$ and totally fuzzy as $m \to \infty$; common values for m are within 1.5 and 4, 2 is most frequently used). The objective function is minimized iteratively subject to the constraints

$$\forall_{1 \leqslant j \leqslant n} : \sum_{i=1}^{c} u_{i,j} = 1, \quad \forall_{1 \leqslant i \leqslant c} : \sum_{j=1}^{n} u_{i,j} > 0. \tag{2}$$

In every iteration step, minimization with respect to $u_{i,j}$ and p_i is done separately. The necessary conditions for a minimum yield update equations for both half-steps. Independent of the choice of the distance function and the prototypes, the membership update equation is

$$u_{i,j} = \frac{1}{\sum_{k=1}^{c}\left(\frac{d^2(x_j,p_i)}{d^2(x_j,p_k)}\right)^{\frac{1}{m-1}}}. \tag{3}$$

In the most simple case of FCM, where the prototypes – to be interpreted as cluster centers – are vectors of the same dimension as the data vectors and the distance function is the Euclidean distance d_{E}, we obtain

$$p_i = \frac{\sum_{j=1}^{n} u_{i,j}^m x_j}{\sum_{j=1}^{n} u_{i,j}^m}. \tag{4}$$

Fig. 2(a) shows an example for an FCM clustering with $c = 7$. The membership degrees are indicated by contour lines, the maximum over all membership degrees is depicted.

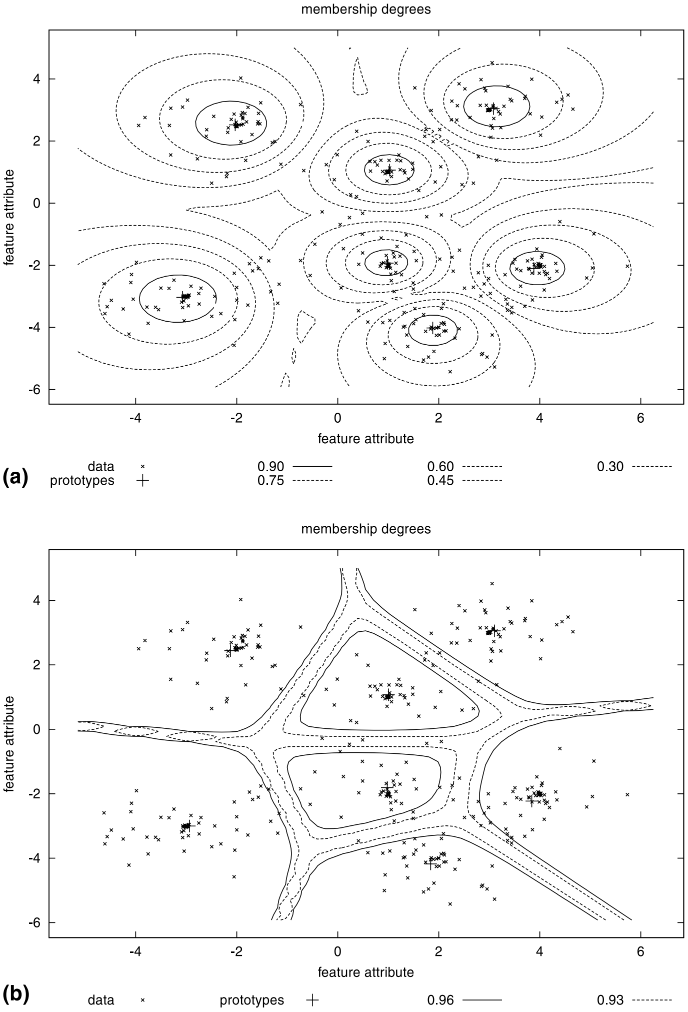

Fig. 2. Effect of modification on the resulting partition: (a) FCM partition; (b) Voronoi-like partition.

The Gustafson–Kessel algorithm (GK) [5] is an extension of FCM, where a cluster prototype contains in addition to the cluster center p_i a symmetric, positive definite matrix A_i. The distance is defined by

$$d^2(x_j, (p_i, A_i)) = (x_j - p_i)^{\mathrm{T}} A_i (x_j - p_i).$$

In order to avoid the trivial solution $A_i \equiv 0$, it is required that $\det(A_i) = 1$ holds. The resulting update equation for the matrix A_i turns out to be the (fuzzy) covariance matrix of the corresponding cluster, normalized w.r.t. the constraint $\det(A_i) = 1$ (for details see [5]).

In this paper, we also utilize the fuzzy c-regression models (FCRMs) algorithm [6], which uses polynomials as cluster prototypes. With real functions $\mathbb{R} \to \mathbb{R}$ the cluster models are characterized by the coefficients of the polynomial, that is, the prototypes are elements of $\mathbb{R}^{q+1}$, where q is the degree of the polynomials. The Euclidean distance d_{E} of FCM is replaced by the residual error $|y - h(x)|$ of a data object (x, y) (consisting of input value x and output value y) to the polynomial h. For simplicity, we consider extended data objects $\hat{x}$ which have an additional component $\hat{x}_0 \equiv 1$. Then, the distance function can be written as

$$d^2((x_j, y_j), p_i) = \left(y_j - p_i^{\mathrm{T}} \hat{x}_j\right)^2.$$

For multiple inputs $\hat{x}_j$ has to be extended further, for instance for $x_j = (a, b)$ we have $\hat{x}_j = (1, a, b, ab, a^2, b^2)$ such that all coefficients of the polynomial can be represented by an element of p_i. The coefficients p_i are obtained in the same fashion as the cluster centers of FCM before, we only have to replace the prototype update equation according to the modified distance function [6]

$$p_i = \left(\sum_{j=1}^{n} u_{i,j}^m (\hat{x}_j \hat{x}_j^{\mathrm{T}})\right)^{-1} \left(\sum_{j=1}^{n} u_{i,j}^m y_j \hat{x}_j\right). \tag{5}$$

3. Rewarding crisp memberships in fuzzy clustering

Some properties of the membership functions defined by (3) are undesired – at least in some application areas, as we have seen in Section 1. Let us consider the question how to reward more crisp membership degrees. We would like to avoid those small peaks of high membership degrees (cf. Fig. 1(c)) and are interested in broad areas of (nearly) crisp membership degrees and only narrow regions where the membership degree changes from 0 to 1 or vice versa (cf. Fig. 1(b)). Let us choose a couple of parameters $a_j \in \mathbb{R}_{\geqslant 0}$, $1 \leqslant j \leqslant n$, and consider the following modified objective function:

$$J = \sum_{j=1}^{n} \sum_{i=1}^{c} u_{i,j}^2 d^2(x_j, p_i) - \sum_{i=1}^{n} a_j \sum_{j=1}^{c} \left(u_{i,j} - \frac{1}{2}\right)^2. \tag{6}$$

The first term is identical to the standard objective function for fuzzy clustering with $m = 2$. Let us therefore examine the second term. If a data object x_j is clearly assigned to one prototype p_i, then we have $u_{i,j} = 1$ and $u_{k,j} = 0$ for all other $k \neq i$. For all these cases, the second term evaluates to $-a_j/4$. If the membership degrees become more fuzzy, the second term increases. Since we seek to minimize (6), this modification rewards crisp membership degrees.

Since there are no additional occurrences of p_i in the second term, the prototype update step remains the same as with the corresponding fuzzy clustering algorithm (FCM, GK, FCRM, etc.).

Lemma 1. *The necessary condition for a minimum of* (6) *yields the following membership update equation*:

$$u_{i,j} = \frac{1}{\sum_{k=1}^{c} \frac{d^2(x_j,p_i)-a_j}{d^2(x_j,p_k)-a_j}}. \tag{7}$$

Proof. Let us consider (6) for a single data object x_j. We apply Lagrange multipliers λ to satisfy the constraint $\sum_{i=1}^{c} u_{i,j} = 1$ for x_j (cf. (2)). We have

$$F = \sum_{i=1}^{c} u_{i,j}^2 d^2(x_j,p_i) - \sum_{i=1}^{c} a_j \left(u_{i,j} - \frac{1}{2} \right)^2 + \lambda \left(\sum_{i=1}^{c} u_{i,j} - 1 \right).$$

Setting the gradient to zero yields

$$\frac{\partial F}{\partial \lambda} = \sum_{i=1}^{c} u_{i,j} - 1 = 0,$$

$$\frac{\partial F}{\partial u_{k,j}} = 2u_{k,j} d^2(x_j,p_k) - 2a_j \left(u_{k,j} - \frac{1}{2} \right) + \lambda.$$

Note that we have fixed $m = 2$ in (6) to obtain an analytical solution. From $\partial F/\partial u_{k,j}$ we obtain

$$u_{k,j} = \frac{-a_j - \lambda}{2d^2(x_j,p_k) - 2a_j}.$$

Using $\partial F/\partial \lambda$, we have

$$\sum_{i=1}^{c} \frac{-a_j - \lambda}{2d^2(x_j,p_i) - 2a_j} = 1 \iff \lambda = -\frac{1}{\sum_{i=1}^{c} 2d^2(x_j,p_i) - 2a_j} - a_j.$$

Substituting λ in the previous equation yields (7). □

Obviously, we immediately run into some problems when choosing $a_j > \|x_j - p_i\|$ for some $1 \leqslant i \leqslant c$. Then, the distance value $d_{i,j}^2 - a_j$ becomes negative and the same is true for the membership degrees $(d_{i,j} = d(x_j,p_i))$. Therefore, we have to require explicitly the constraint $0 \leqslant u_{i,j} \leqslant 1$. From the Kuhn–Tucker conditions we obtain a simple solution as long as only a single prototype has a distance smaller than a_j to x_j, in this case we obtain the minimum by setting $u_{i,j} = 1$. However, things are getting more complicated if multiple negative terms $d_{i,j} - a_j$ occur.

If we want to avoid the problem of negative memberships, we could also heuristically adapt the reward a_j such that $d^2_{i,j} - a_j$ always remains positive. The maximal reward we can give is then

$$\min d^2_{*,j} = \min\{d^2_{i,j} \mid i \in \{1, \ldots, c\}\} - \eta$$

and thus

$$u_{i,j} = \frac{1}{\sum_{k=1}^{c} \frac{d^2_{i,j} - \min d^2_{*,j}}{d^2_{k,j} - \min d^2_{*,j}}}. \tag{8}$$

Without an $\eta > 0$ we find always an i such that $d^2(x_j, p_i) - \min^2_{*,j} = 0$ and therefore $u_{i,j} = 1$. In other words, for $\eta = 0$ we obtain a crisp partition, the algorithm reduces in this case to (crisp) c-means (also known as ISODATA). The choice of η influences the fuzziness of the partition, similar to the fuzzifier m with FCM. Fig. 1(b) shows different partitions for η ranging from 0.01 to 0.2.

Surprisingly, besides the different shape of the membership functions, the resulting algorithm performs very similar to conventional FCM in terms of resulting cluster centers. The modified version seems slightly less sensitive to noise and outliers, as we will see in the next section. Fig. 2 compares the results of FCM and our modification for an example dataset. The maximum over all membership degrees is indicated by contour lines.

4. Memberships induced by Voronoi distance

With FCM the Euclidean distance between cluster centroids plays a central role in the definition of the membership functions. The idea is to "represent" each cluster by a single data instance – the prototype – and to use the distance between prototype and data objects as the distance between cluster and data object. Then, the relative distances (cf. (3)) define the degree of membership to a cluster, e.g., if the distance between x_j and p_1 is half the distance to p_2, the membership degree $u_{1,j}$ is twice as large as $u_{2,j}$. If we consider crisp membership degrees things are different, the membership degree does not depend on the ratio of distances, but the distances serve as threshold values. If the distance to p_1 is smaller than to p_2 – no matter how much smaller – we always have $u_{j,1} = 1$.

Let us consider (8) again and assume that p_i is closest to x_j. No matter if x_j is far away from p_i (but all other p_k are even further away) or x_j is very close to p_i, the numerator of the distance ratio is always constant η. Inside a region in which all data points are closest to p_i, the distance to cluster i is considered to be constant η. The membership degrees $u_{k,j}$ are therefore determined by the denominator, that is, mainly by $d_{\mathrm{E}}^2(x_j, p_k)$. Therefore, the membership degrees obtained by (8) are no longer defined by a ratio of distances, but the maximum reward ($\min d_{*,j}^2$) has the flavor of a threshold value.

Let us consider a crisp partition, which is induced by cluster centroids. The resulting partition is usually referred to as the Voronoi diagram. The Euclidean distance of a data object x_j to the hyperplane that separates the clusters p_i and p_s is given by $|(x_j - h_s)^{\mathrm{T}} n_s|$, where h_s is a point on the hyperplane, e.g., $h_s = (p_s + p_i)/2$, and n_s is the normal vector $n_s = \beta_s \cdot (p_i - p_s)$ with $\beta_s = 1/(\|p_i - p_s\|)$ for $s \neq i$. How can we define the distance of a data object x_j to the Voronoi cell of cluster i rather than to a separating hyperplane? If we do not take absolute values, we obtain *directed* distances $(x_j - h_s)^{\mathrm{T}} n_s$, which become positive if x_j lies on the same side as the cluster center and negative if x_j lies on the opposite side. Taking the absolute value of the minimum over all the directed distances yields the distance to the *border of the cell* (see also [7] for the case of rectangles in shell clustering). If x_j lies within the Voronoi cell of cluster i, then the distance to the *cell* is zero. We can formalize this special case easily by setting $\beta_s = 1$ and defining:

$$d_V(x_j, p_i) = \left| \min_{1 \leqslant s \leqslant c} (x - h_s)^{\mathrm{T}} n_s \right|.$$

In Fig. 3, x_j is closest to the separating line between p_1 and p_2, therefore this distance serves as the distance to the Voronoi cell of p_1. The graph of d_V for the four clusters of Fig. 3 is shown in Fig. 4.

If we do not scale the normal vectors n_s to unit length, but assume $\beta_s = 1$ for all s, we preserve the shape of d_V (position of hyperplanes does not change), only the gradient of the different hyperplanes varies. The following lemma

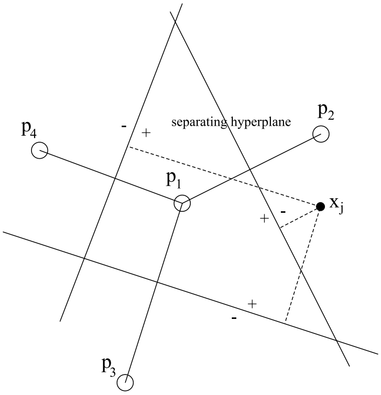

Fig. 3. Voronoi cell of centroid p_1.

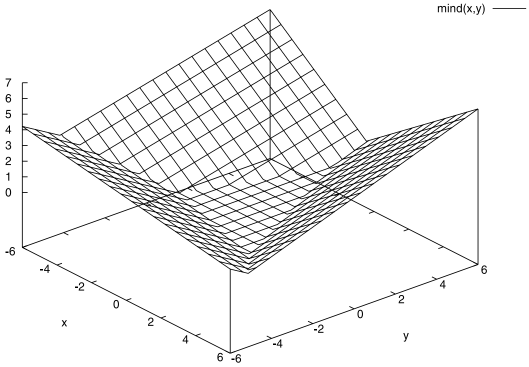

Fig. 4. Distance to Voronoi cell.

establishes a connection between the scaled Voronoi distance and the approach discussed in the previous section.

Lemma 2. *Given a Voronoi diagram induced by a set of distinct points* p_i, $1 \leqslant i \leqslant c$, *and a point* x. *Using* $\beta_s = 1$ *for all* $1 \leqslant s \leqslant c$, *the (scaled) distance between* x *and the Voronoi cell of point* p_i *is given by*

$$d_V(x,p_i) = \frac{1}{2}\left(d_{\mathrm{E}}^2(x,p_i) - \min_{1\leqslant s\leqslant c} d_{\mathrm{E}}^2(x,p_s)\right). \tag{9}$$

Proof. Some simple transformations yield the following chain of equalities:

$$\begin{aligned}
d_V(x,p_i) &= \left|\min_{1\leqslant s\leqslant c}\left(x - \frac{p_s+p_i}{2}\right)^{\mathrm{T}}(p_i - p_s)\right| \\
&= \left|\min_{1\leqslant s\leqslant c} x^{\mathrm{T}}(p_i - p_s) - \frac{1}{2}(p_i^{\mathrm{T}}p_i - p_s^{\mathrm{T}}p_s)\right| \\
&= \frac{1}{2}\left|\min_{1\leqslant s\leqslant c} x^{\mathrm{T}}x - 2x^{\mathrm{T}}p_s + p_s^{\mathrm{T}}p_s + (x^{\mathrm{T}}x - 2x^{\mathrm{T}}p_i + p_i^{\mathrm{T}}p_i)\right| \\
&= \frac{1}{2}\left|\min_{1\leqslant s\leqslant c} \|x-p_s\|^2 - \|x-p_i\|^2\right| \\
&\overset{(\star)}{=} \frac{1}{2}\left(\|x-p_i\|^2 - \min_{1\leqslant s\leqslant c}\|x-p_s\|^2\right).
\end{aligned}$$

In the above equation ($\star$) we have used the trivial fact that any $d_{\mathrm{E}}(x,p_i)$ is greater than or equal to $\min_{1\leqslant s\leqslant c} d_{\mathrm{E}}(x,p_s)$. □

Thus, the lemma tells us, by using a maximum reward the resulting membership values are identical to those that we would obtain by using standard

FCM membership functions and a (scaled) Voronoi cell distance instead of Euclidean centroid distance.

By replacing the Euclidean distance with the Voronoi distance during membership calculation, we obtain different membership functions which are much closer to those of the original c-means (cf. Fig. 2(b)). In this sense we can speak of a new c-means fuzzification.

Note that with FCM squared Euclidean distances are used to determine the membership degrees, but if we use the maximum reward/Voronoi distance we use Euclidean distances to the Voronoi cell, which are not squared. Therefore, the modification might be less sensitive to noise and outliers.

5. Interpretation as fuzzified minimum function

In the previous sections, we have seen how the introduction of a reward term leads us to a fuzzy partition which is more closely related to the results of the crisp c-means (or a Voronoi partition) than the standard FCM partition. The c-means algorithm minimizes the objective function

$$\sum_{j=1}^{n} \min_{1 \leqslant i \leqslant c} \|x_j - p_i\|^2.$$

The crisp minimum function can be reformulated as

$$\min_{1 \leqslant i \leqslant c} \|x_j - p_i\|^2 = \sum_{i=1}^{c} u_{i,j} \|x_j - p_i\|^2 \tag{10}$$

using crisp membership degrees $u_{i,j}$ defined by $u_{i,j} = 1 \iff i = \mathrm{argmin}_i \|x_j - p_i\|^2$ (0 otherwise). If the partition of the discussed algorithm can be interpreted as a fuzzified Voronoi diagram, is it also possible to interpret the term $\sum_{i=1}^{c} u_{i,j}^2 d_V(x_j, p_i)$ as a fuzzified minimum function? We have faced the problem of a fuzzified minimum function before in [7]. There, we considered the terms $d_i = b_i - \min_{1 \leqslant s \leqslant k} b_s$ in a minimum term $\min(b_1, b_2, \ldots, b_k)$ as the "distance of argument i to the minimum" and used the standard FCM membership degrees to assign a "degree of minimality" to each argument b_i (a minimality degree is within $[0, 1]$ and high values indicate that b_i is close to the minimum of $b_1, \ldots, b_k$). Note that this leads to the same equations as we have discussed in the previous sections.

Regarding the approximation quality, we state the following theorem:

Theorem 1 (Fuzzified minimum function). *Let* $f : \mathbb{R}_{\geqslant 0} \to \mathbb{R}_{\geqslant 0}$ *be a strictly increasing function with* $f(x) \geqslant x$, *let* $\eta \in \mathbb{R}_{\geqslant 0}$. *Then for all* $d = (d_1, \ldots, d_k) \in \mathbb{R}^k$, $D_s = (f(d_s - \min\{d_1, \ldots, d_k\}) + \eta)^q$, $q \geqslant 1$, *the following inequality holds*:

$$\left|\sum_{s=1}^{k} u_s d_s - \min\{d_1, d_2, \ldots, d_k\}\right| < \eta^q r + \eta(k-r-1) \leqslant \eta(k-1),$$

where $u_s = 1/(\sum_{i=1}^{k} \frac{D_s}{D_i})$ *and* r *is the number of indices* s *for which* d_s *has at least a distance of* $1-\eta$ *from the minimum*:

$$r = |\{s | 1 \leqslant s \leqslant k, d_s - \min\{d_1, d_2, \ldots, d_k\} > 1 - \eta\}|.$$

Proof. We have the following equality:

$$\begin{aligned}\sum_{s=1}^{k} \frac{d_s}{D_s \sum_{i=1}^{k} \frac{1}{D_i}} &= \sum_{s=1}^{k} \frac{d_s}{D_s \frac{\sum_{i=1}^{k} \prod_{t=1,t\neq 1}^{k} D_t}{\prod_{i=1}^{k} D_i}} = \sum_{s=1}^{k} \frac{d_s \prod_{i=1}^{k} D_i}{D_s \sum_{i=1}^{k} \prod_{t=1,t\neq 1}^{k} D_t} \\ &= \sum_{s=1}^{k} \frac{d_s \prod_{i=1,i\neq s}^{k} D_i}{\sum_{i=1}^{k} \prod_{t=1,t\neq 1}^{k} D_t} = \frac{\sum_{s=1}^{k} d_s \prod_{i=1,i\neq s}^{k} D_i}{\sum_{i=1}^{k} \prod_{t=1,t\neq 1}^{k} D_t}. \qquad (11)\end{aligned}$$

Using the abbreviations $M = \min\{d_1, d_2, \ldots, d_k\}$ we estimate the approximation error as follows:

$$\begin{aligned}&\left|\frac{\sum_{s=1}^{k} d_s \prod_{i=1,i\neq s}^{k} D_i}{\sum_{s=1}^{k} \prod_{i=1,i\neq s}^{k} D_i} - M\right| \\ &= \left|\frac{\left(\sum_{s=1}^{k} d_s \prod_{i=1,i\neq s}^{k} D_i\right) - M\left(\sum_{s=1}^{k} \prod_{i=1,i\neq s}^{k} D_i\right)}{\sum_{s=1}^{k} \prod_{i=1,i\neq s}^{k} D_i}\right| \\ &= \left|\frac{\sum_{s=1}^{k} (d_s - M) \prod_{i=1,i\neq s}^{k} D_i}{\sum_{s=1}^{k} \prod_{i=1,i\neq s}^{k} D_i}\right| \overset{\star^1}{=} \left|\frac{\sum_{s=1}^{k} (d_s - M) \prod_{i=1,i\neq s}^{k} D_i}{\sum_{s=1}^{k} \prod_{i=1,i\neq s}^{k} D_i}\right| \\ &\overset{\star^2}{\leqslant} \left|\frac{\eta^q \sum_{s=2}^{k} (d_s - M) \prod_{i=2,i\neq s}^{k} D_i}{\sum_{s=1}^{k} \prod_{i=1,i\neq s}^{k} D_i}\right| \overset{\star^3}{<} \left|\frac{\eta^q \sum_{s=2}^{k} (d_s - M) \prod_{i=2,i\neq s}^{k} D_i}{\prod_{i=2}^{k} D_i}\right| \\ &= \left|\eta^q \sum_{s=2}^{k} \frac{(d_s - M)}{D_s}\right| \leqslant \eta^q \sum_{s=2}^{k} \left|\frac{(d_s - M)}{D_s}\right| \\ &\overset{\star^4}{<} \eta^q \sum_{s=2}^{k} \left|\frac{(d_s - M)}{(d_s - M)(d_s - M + \eta)^{q-1}}\right| \\ &= \eta^q \sum_{s=2}^{k} \left|\frac{1}{(d_s - M + \eta)^{q-1}}\right| \overset{\star^5}{\leqslant} \eta^q \sum_{s=2}^{k} \left|\frac{1}{\eta^{q-1}}\right| = \eta(k-1).\end{aligned}$$

Remarks

$\bigstar^1$ Without loss of generality we have assume that d_1 is the minimum and have $(d_1 - M) = 0$.

$\bigstar^2$ From $d_1 = M$ we can conclude $D_1 = (f(d_1 - d_1) + \eta)^q \leqslant \eta^q$.

$\bigstar^3$ We have dropped all summands in the denominator $\sum_{s=1}^{k} \prod_{i=1,i\neq s}^{k} D_i$ that contain D_1. All summands are positive.

$\bigstar^4$ We drop one η in the denominator $D_s = (d_s - M + \eta)(d_s - M + \eta)^{q-1}$ which makes the term smaller.

$\bigstar^5$ Here we assume the worst case that all d_s are minimal and thus $d_s - M = 0$. (However, if this would actually be the case, we can see from the equality $\bigstar^1$ that the approximation error is zero.) We also obtain an equality if $q = 1$.

If some $d_s, s \in \{2, 3, \ldots, k\}$, have reached a distance $d_s - M \geqslant 1 - \eta$ from the minimum, the estimation can be improved. [1] If we continue from the result after $\bigstar^3$ we have $d_s - M < f(d_s - M) + \eta < (f(d_s - M) + \eta)^q = D_s$ and thus may substitute $(d_s - M)$ by D_s. This leads us to an error below $\eta^q(k-1)$.

To summarize both estimations, if there are r values that have a distance of at least $d_s > 1 - \eta + M$, we have an error smaller than $\eta(k - r - 1) + \eta^q r$. □

Although we deal only with non-negative distances in the context of clustering, note that the fuzzified minimum function does also work with negative terms. If there are negative arguments, the minimum will also be negative, and subtracting the (negative) minimum from all other arguments yields a set of non-negative arguments. Also note that the fuzzified minimum is once differentiable for $q > 1$. Fig. 5 shows an example where we take the pointwise minimum of three functions. The resulting fuzzified minimum is displayed for two different values of $\eta = 0.1/0.2$ (solid lines) using $q = 1.5$. According to the theorem, the error is bounded by 0.06/0.18 if the minimum is clearly separated from the other values and 0.2/0.4 in general.

6. Combining clustering and regression

If we consider a fuzzy model using amongst others a rule "if x is approximately zero, then $y = 2x + 1$", we expect the resulting model to *behave near zero as it has been described*. Again, many systems in the literature allow massively overlapping premise fuzzy sets for higher-order TS models. In this

[1] This additional condition has not been mentioned in [7].

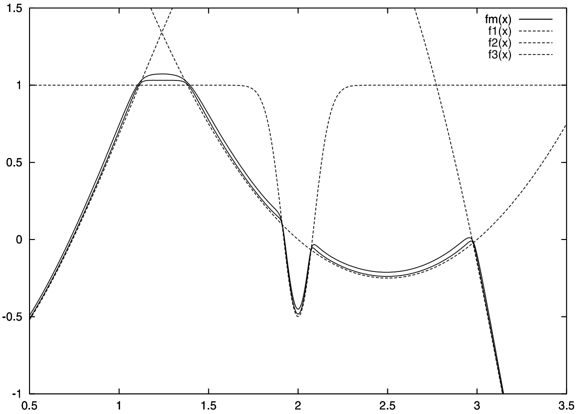

Fig. 5. Minimum of three functions.

case, the resulting function *does not behave at all* like one might expect from the conclusion of the rule, but is composed out of polynomials of many different rules. The fuzzy model will behave as desired only if the premise fuzzy set "approximately zero" has a large support of 1 near zero and thus there is only one rule applicable. This leads us to trapezoidal or even crisp premise fuzzy sets. Note that in case of crisp membership functions we have the classical case of piecewise polynomial function approximation. Since the support of the local polynomials is not fixed in advance, this is a non-trivial problem in the classical case, too. With crisp premise memberships and linear functions in the conclusion we again have the piecewise linear case, we therefore consider polynomials of degree 2 in this paper – but the algorithm can also be used for higher polynomial degrees.

Thus, the goal is to partition the input space such that the resulting fuzzy membership functions have a large support of 1. This can be done by means of fuzzy clustering, for example the fuzzy c-means algorithm (using a fuzzifier $1 < m \leqslant 1.5$) [2] as proposed in [8]. An even better solution is obtained, when we apply our modified algorithm that rewards crisp membership values for the partition of the input space (as we have discussed with Fig. 1).

For each cluster in this partition, we use a polynomial of degree 2 to locally approximate the input–output relationship in this cluster. This can be done by means of switching regression models [6]. If we combine both algorithms, we obtain a fuzzy clustering/regression algorithm where each cluster can be in-

[2] By means of a fuzzifier near 1 we obtain more crisp and convex membership degrees.

```
choose number of clusters $c$;
choose termination threshold $\varepsilon > 0$;
choose $\eta > 0$;
initialize prototypes $p_i$, $q_i$;
repeat
   update memberships using (8) and distances (12);
   update prototypes $p_i$ using (4);
   update prototypes $q_i$ using (5);
until change in memberships drops below $\varepsilon$;
```

Fig. 6. The FM algorithm.

terpreted as a rule in a TS model. Since both algorithms (clustering and regression) are objective function-based, their combination is straightforward. The new *fuzzy model* (FM) algorithm uses the sum of both distance functions (FCM and FCRM) in the modified clustering algorithm:

$$d^2((x_j, y_j), (p_i, q_i)) = \underbrace{\|x_j - p_i\|^2}_{\text{FCM distance}} + \underbrace{\left(y_j - q_i^{\mathrm{T}}\hat{x}_j\right)^2}_{\text{FCRM distance}}. \tag{12}$$

The FCM distances are taken with respect to the input value x_j and cluster center p_i, while the FCRM distances are taken with respect to the given output value y_j and the value of the polynomial at $\widehat{x_j}$ with coefficients q_i. The algorithm is sketched in Fig. 6.

Since there are no dependencies between the parameters of the modified clustering and regression prototypes (p_i and q_i), the same prototype update equations hold for the combined algorithm. Nevertheless, cluster centers and polynomials influence each other indirectly by means of the membership degrees, which depend on the distance to both models. (A different way to combine FCM and linear FCRM can be found in [3].)

Of course, instead of using FCM to partition the input space other fuzzy clustering algorithms may be used, for instance the GK algorithm. To do this, we have to replace the FCM term in (12) and the corresponding prototype update in Fig. 6. For the examples in Figs. 7(a) and 8(a) we have chosen the GK algorithm. The figures show two functions [3] from which we generated noisy sample data. The data were distributed evenly over the input space and were not concentrated in clusters. The best resulting approximations (from a set of different cluster initializations) are shown in Figs. 7(b) ($\eta = 0.05$) and 8(b) ($\eta = 0.1$). At the bottom of the plots you can observe the contour lines of the membership functions and thus the obtained partition of the input space, which has been adopted pretty good to

[3] Fig. 7(a): $f(x,y) = \mathrm{atan}(3x + 4y) + 3\exp(-(3x-4)^2 - (2y-2)^2 + (3x-4)\cdot(2y-2))$, Fig. 8(a): $f(x,y) = \mathrm{atan}(x)\cos(y^2)$.

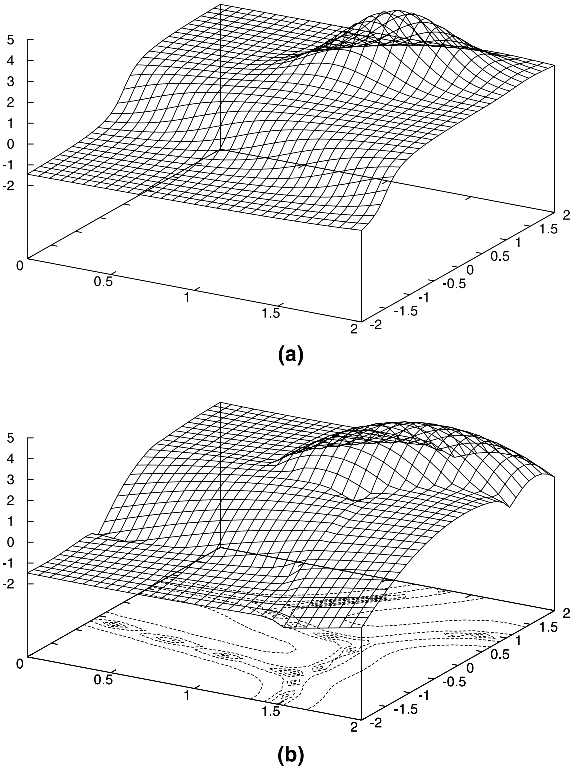

Fig. 7. Fuzzy model using five GK clusters to partition the input space: (a) original function; (b) approximation.

the peculiarities of the respective function. This is remarkable because the data distribution itself does not provide any hints for the optimal location of the cluster centers (data evenly distributed), only the output values help in adjusting them. When clustering is used to learn a fuzzy system it is quite often assumed that the data points clump together where a new local model should be inserted. However, in real-world data it is likely that regions of high data density simply indicate the operating points of the systems and not necessarily good centers for local models. Obviously, there is no such assumption in this approach.

The large white regions at the bottom of the figures indicate those parts of the input space where the memberships are dominated by a single cluster, that is, we have a membership degree very close to 1 for a single cluster. Thus we can be sure that the fuzzy model indeed reflects the regression model that is associated with this cluster.

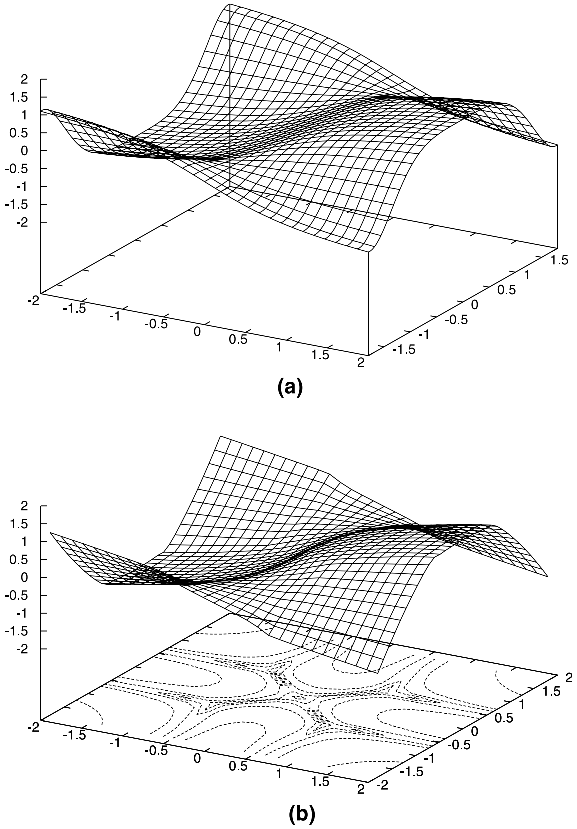

Fig. 8. Fuzzy model using eight GK clusters to partition the input space: (a) original function; (b) approximation.

7. Conclusions

In this paper, we have presented a modification of FCM, which is more closely related to the original (non-fuzzy) c-means algorithm. This can be desirable for certain applications, for example if we want to attach linguistic labels to the membership functions. We have proposed a modification of the objective function that is minimized by FCM to reward nearly crisp memberships. If we (heuristically) select a (in some sense) "maximum reward", we have shown that the membership functions correspond to membership functions that would be obtained by using the distance between the Voronoi cell and a data object.

The obtained membership functions can also be interpreted as a fuzzified minimum function. In retro-perspective, we can consider the modification as a substitution of the crisp minimum function of *c*-means by a fuzzified variant.

Our modified version is suitable, when fuzzy clustering is applied to partition the input space of a (sampled) function, in order to construct local models/ approximations of the function. In this case it is desired that on the one hand each local model should be valid on a large area as possible and on the other hand that a continuous switching between the model is carried out on the boundaries between the regions where the models are valid. With our approach both these requirements can be satisfied.

References

[1] J.C. Bezdek, A convergence theorem for the fuzzy ISODATA clustering algorithms, IEEE Trans. Pattern Anal. Machine Intell. 2 (1) (1980) 1–8.

[2] J. Casillas, O. Cordón, F. Herrera, and L. Magdalena (Eds.), Trade-off between accuracy and interpretability in fuzzy rule-based modelling. Studies in fuzziness and soft computing, Physica (to appear).

[3] J.-Q. Chen, Y.-G. Xi, Z.-J. Zhang, A clustering algorithm for fuzzy model identification, Fuzzy Sets and Systems 98 (1998) 319–329.

[4] R.N. Davé, R. Krishnapuram, Robust clustering methods: a unified view, IEEE Trans. Fuzzy Systems 5 (2) (1997) 270–293.

[5] D.E. Gustafson, W.C. Kessel, Fuzzy clustering with a fuzzy covariance matrix, in: Proceedings of the IEEE Conference on Decision and Control, 1979, pp. 761–766.

[6] R.J. Hathaway, J.C. Bezdek, Switching regression models and fuzzy clustering, IEEE Trans. Fuzzy Systems 1 (3) (1993) 195–204.

[7] F. Höppner, Fuzzy shell clustering algorithms in image processing: fuzzy *c*-rectangular and 2-rectangular shells, IEEE Trans. Fuzzy Systems 5 (4) (1997) 599–613.

[8] F. Höppner, F. Klawonn, Obtaining interpretable fuzzy models from fuzzy clustering and fuzzy regression, in: Proceedings of the 4th International Conference on Knowledge-Based Intelligent Engineering Systems & Allied Technologies, Brighton, UK, August 2000, pp. 162–165.

[9] F. Höppner, F. Klawonn, R. Kruse, T.A. Runkler, Fuzzy Cluster Analysis, Wiley, Chichester, England, UK, 1999.

[10] T.A. Runkler, J.C. Bezdek, Alternating cluster estimation: a new tool for clustering and function approximation, IEEE Trans. Fuzzy Systems 7 (4) (1999) 377–393.