Focus

Soft Computing 6 (2002) 6–13 © Springer-Verlag 2002
DOI 10.1007/s005000100122

Learning indistinguishability from data

F. Höppner, F. Klawonn, P. Eklund

Abstract In this paper we revisit the idea of interpreting fuzzy sets as representations of vague values. In this context a fuzzy set is induced by a crisp value and the membership degree of an element is understood as the similarity degree between this element and the crisp value that determines the fuzzy set. Similarity is assumed to be a notion of distance. This means that fuzzy sets are induced by crisp values and an appropriate distance function. This distance function can be described in terms of scaling the ordinary distance between real numbers. With this interpretation in mind, the task of designing a fuzzy system corresponds to determining suitable crisp values and appropriate scaling functions for the distance. When we want to generate a fuzzy model from data, the parameters have to be fitted to the data. This leads to an optimisation problem that is very similar to the optimisation task to be solved in objective function based clustering. We borrow ideas from the alternating optimisation schemes applied in fuzzy clustering in order to develop a new technique to determine our set of parameters from data, supporting the interpretability of the fuzzy system.

Keywords Fuzzy systems, Equality relations, Function approximation, Alternating optimisation

1 Introduction

Fuzzy sets are often understood on a purely intuitive basis. The role of the membership degrees is nothing more than a weighting concept. As a consequence, learning from data in the setting of fuzzy systems becomes a mere parameter tuning task.

This is, of course, not always true, if we for instance think of a possibilistic interpretation of fuzzy sets. However, a data driven possibilistic framework usually remains in the general context of probability theory, although set valued random variables might be considered.

There are evidently at least two meaningful interpretations of "indistinguishability". Data objects might be inherently indistinguishable: due to a lack of attributes we might be unable to distinguish one from the other. There is not much that we can do about this rather than seeking more information about the objects. On the other hand, sometimes we do not want to distinguish between two objects, because they are so similar that we consider them together as a single case. Following this interpretation of intended indistinguishability, we perform a granularisation of the real line, which can be done, for example, by means of fuzzy sets. In this paper we revisit the interpretation of fuzzy sets on the basis of equality relations and establish learning techniques that are based on this interpretation.

Section 2 establishes the connection between equality relations and fuzzy sets. The underlying fundamental principle is a scaling of the ordinary distance between real numbers. Section 3 discusses fuzzy systems and their interpretation in the view of the previously introduced concepts. The development of an algorithm to automatically generate a fuzzy system from data in terms of the provided interpretation of fuzzy sets is explained in Section 4.

2 Equality relations

An equality relation (w.r.t. a t-norm $*$) on the set X is a fuzzy relation $E : X \times X \to [0,1]$ satisfying

(E1) $E(x,x) = 1, \quad$ (reflexivity)

(E2) $E(x,y) = E(y,x), \quad$ (symmetry)

(E3) $E(x,y) * E(y,z) \le E(x,z) \quad .$ (transitivity)

Sometimes E is also called a similarity relation [9, 13], indistinguishability operator [11], fuzzy equality (relation) [2, 6], fuzzy equivalence relation [10] or proximity relation [1], also depending on the chosen t-norm.

In this paper we concentrate on equality relations w.r.t. the Łukasiewicz t-norm defined by $\alpha * \beta = \max\{\alpha + \beta - 1, 0\}$. There is a duality between equality relations w.r.t. the Łukasiewicz t-norm and pseudo-metrics bounded by one: A pseudo-metric δ bounded by one induces an equality relation E by $E = 1 - \delta$ and vice versa.

There are various connections between fuzzy sets and equality relations starting from pioneering work like [11, 12]. Here we focus on the interpretation of fuzzy sets as vague points induced by crisp points and an underlying equality relation. The fuzzy set μ_{x_0} induced by the point $x_0 \in X$ in the presence of the equality relation E is defined as the (fuzzy) set of all elements that are (fuzzy) equal to x_0, i.e. $\mu_{x_0} = E(x, x_0)$. When X is an interval and the equality relation E is defined in terms of the standard

F. Höppner (✉), F. Klawonn
Department of Electrical Engineering and Computer Science, Emden University of Applied Sciences, Constantiaplatz 4, D-26723 Emden, Germany

P. Eklund
Department of Computing Science, Umeå University, SE-90187 Umeå, Sweden

metric on X by $E(x,y) = 1 - \min\{|x-y|, 1\}$, then μ_{x_0} is a triangular fuzzy set.

Scaling [4] is an important concept in this view of fuzzy sets. The idea behind scaling is to modify the standard metric by scaling factors, stretching the distance (and decreasing the associated equality degrees) in regions where it is important to distinguish well between values and contracting the distance (and increasing the associated equality degrees) in regions where the exact value is not very important in the considered context or application.

In this way, if the interval $X = [a,b]$ is the underlying domain, a scaling factor $c(x) \geq 0$ is associated to each element $x \in X$, indicating the importance of the exactness of values in the neighbourhood of x. The scaled distance between two points $x_1, x_2 \in X$ is then

$$\left| \int_{x_1}^{x_2} c(x)\mathrm{d}x \right| \quad .$$

This means that the scaling function $c(x)$ induces a transformation

$$t : [a,b] \to \left[0, \int_a^b c(x)\mathrm{d}x\right], \quad x \mapsto \int_a^x c(s)\mathrm{d}s$$

and the distance between two points $x_1, x_2 \in X$ is not measured in X but in the transformed (scaled) domain.

In fuzzy systems it is very popular to work with 'fuzzy partitions' of a real interval $[a,b]$ that use trapezoidal membership functions at the boundaries and triangular membership functions whose membership degrees add to one. Such fuzzy partitions are uniquely determined by points $a \leq x_1 < x_2 < \cdots < x_n \leq b$ where the trapezoidal membership functions are defined as

$$\mu_1(x) = \begin{cases} 1 & \text{if } a \leq x \leq x_1 \\ \frac{x_2 - x}{x_2 - x_1} & \text{if } x_1 < x < x_2 \\ 0 & \text{otherwise} \end{cases}$$

and

$$\mu_n(x) = \begin{cases} 1 & \text{if } x_n \leq x \leq b \\ \frac{x - x_{n-1}}{x_n - x_{n-1}} & \text{if } x_{n-1} < x < x_n \\ 0 & \text{otherwise} \quad . \end{cases}$$

The triangular membership functions are given by

$$\mu_i(x) = \begin{cases} \frac{x - x_{i-1}}{x_i - x_{i-1}} & \text{if } x_{i-1} \leq x \leq x_i \\ \frac{x_{i+1} - x}{x_{i+1} - x_i} & \text{if } x_i < x < x_{i+1} \\ 0 & \text{otherwise} \end{cases}$$
$$\text{for } i \in \{2, \ldots, n-1\} \quad . \tag{1}$$

When we choose the scaling function $c(x)$ as

$$c(x) = \begin{cases} 0 & \text{if } a < x < x_1 \ \text{ or } \ x_n < x < b \\ \frac{1}{x_i - x_{i-1}} & \text{if } x_{i-1} < x < x_i \ , \end{cases} \tag{2}$$

we obtain an equality relation and the fuzzy sets μ_i are exactly the fuzzy sets μ_{x_i} that are induced by the points x_i in the context of the equality relation derived from the scaling function $c(x)$.

3
Fuzzy systems in the view of equality relations

So far we have considered a single interval endowed with an equality relation so that single points induce fuzzy sets. In applications as for instance in fuzzy control we have to deal with various domains for input and output variables simultaneously. Especially the rules of Mamdani fuzzy controllers can be interpreted in the context of equality relations where each fuzzy set can be seen as induced by a single point in the presence of a suitable equality relation [5]. Nevertheless, in such a case we have to build the product space of the considered domains and must aggregate the equality relations to a joint one on the product space.

In principle, we could extend the concept of a scaling function to product space $c : \mathbb{R}^n \to [0, \infty)$. However, this would mean that we would have to define the distance between two points $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^n$ in the following way:

$$\inf\left\{ \left| \int_P c(s)\mathrm{d}s \right| \mid P \text{ is a path from } x \text{ to } y \right\} \quad .$$

Unless c is a potential function and the value of the integral is independent of the path, it would in general not be tractable to compute this distance.

When we consider equality relations on product spaces, the crucial notions are aggregation and independence.

As long as we assume some kind of independence of the equality relations, aggregation can be done in a straightforward way. It turns out that this seems to be the underlying assumption behind many fuzzy controllers. However, taking the concept of scaling seriously, the independence assumption seems not to be justified in typical control applications. Consider a controller using the error and the change of error as input variables. Usually it is not very important to consider the change of error, when the error is large, since then a strong control action has to be carried out anyway. This means that we might use a small scaling factor for the domain representing the change of error. However, when the error is almost zero, it is very important to know the value of the change of error almost exactly, in order to take the right control action. This would speak in favour for a large scaling factor for the domain representing the change of error. The scaling or the indistinguishability in these two domains does not seem to be independent.

A detailed discussion of the independence concept in the context of equality relations is outside the scope of this paper. Nevertheless we would like to point out some facts.

Independence can be defined in different ways. One possibility is to say a structure on a product space $X \times Y$ is formed by two independent structures on X and Y, if we can fix any element of X and always obtain the structure on Y and vice versa. In probabilistic terms this independence notion simply requires for two random variables $P(Z_1 = z_1 | Z_2 = z_2) = P(Z_1 = z_1)$. This would mean that Z_1 is independent of Z_2. In probability theory we immediately have that this implies that Z_2 is also independent of Z_1, i.e. $P(Z_2 = z_2 | Z_1 = z_1) = P(Z_2 = z_2)$.

The following example illustrates that the situation is different for equality relations. Consider the unit square and the metric defined by the transformation $t(x,y) = (x, (1-0.5x)y)$ (see Fig. 1), i.e. the distance between two points (x_1,y_1) and (x_2,y_2) is the distance between the transformed points

$$\|t(x_1,y_1) - t(x_2,y_2)\| \ .$$

We obviously have

$$\|t(x_1,y_1) - t(x_2,y_1)\| = |x_1 - x_2| \ ,$$

so the distance on X is independent of the element $y_1 \in Y$. However, in this case the distance on Y strongly depends on the choice of the element in X.

In the following we restrict our considerations to equality relations on product spaces that are obtained by applying an aggregation operation to scaling induced equality relations on the single domains. A general discussion on how equality relations can be aggregated can be found in [7]. For reasons of simplicity we only consider the aggregation operation minimum and product.

There are various approaches to fuzzy systems on the basis of equality relation. In the following we consider a very simple type of fuzzy system. The domain of each input variable is endowed with a piecewise constant scaling function of the form (2) and the corresponding reference points x_i are given. The rules assign to each combination of reference points of different input domains a crisp output value. In this way we avoid the problem of defuzzification. The specification of a fuzzy controller reduces in the context to the choice of suitable reference points and appropriate output values. The scaling functions are implicitly given by the reference points. In principle, we could choose the reference points and the scaling functions more or less independently. But if we assume that we try to minimise the number of reference points, we only have to specify a new reference point, when the previous reference point does not provide any information, i.e. when the membership degree of the corresponding fuzzy set reaches zero. In this view the reference points and the scaling functions should not be chosen independently.

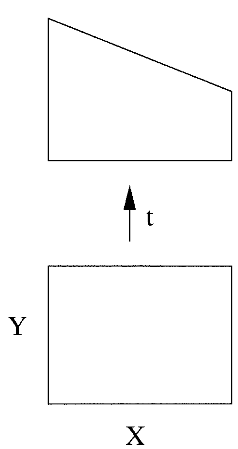

Fig. 1. A transformation

4 Equality relations induced by data

Now that we have clarified the interpretation of fuzzy sets in terms of scaling and indistinguishability, we can try to design learning techniques for fuzzy systems that are based on these ideas.

Fuzzy clustering (for an overview see for example [3]) is very much in the spirit of our concepts. Clusters are usually represented by single points and more sophisticated algorithms can even incorporate a scaled distance adapted to the data. However, the membership degrees are derived in a different way from the distance function and the scaling is always an individual scaling for each cluster.

We will introduce a clustering-like alternating optimisation technique that is devised to overcome these problems and is more in the spirit of the proposed interpretation of fuzzy sets.

Let us consider a two-dimensional fuzzy system (two input variables) which defines a function $\hat{f} : X \times Y \to Z$ by means of

- n fuzzy singletons $\mu_i : X \to [0,1]$ with core x_i,
- m fuzzy singletons $\nu_j : Y \to [0,1]$ with core y_j and
- $n \cdot m$ output values $z_{i,j}$

and thus $n \cdot m$ rules of the type

if x is about x_i and y is about y_j then z is $z_{i,j}$

Then, the output value $\hat{f}$ is given by

$$\hat{f}(x,y) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} \top(\mu_i(x), \nu_j(y)) \cdot z_{i,j}}{\sum_{i=1}^{n}\sum_{j=1}^{m} \top(\mu_i(x), \nu_j(y))}$$

where $\top$ is a t-norm. The parameters of the fuzzy system are the fuzzy set core values $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, $\mathbf{y} = (y_1, y_2, \ldots, y_m)$ and the output values $\mathbf{z} = (z_{1,1}, z_{1,2}, \ldots, z_{1,m}, z_{2,1}, \ldots, z_{n,m})$. We assume that the input space is bounded and fix x_1/x_n to the minimum/maximum value (same for y_1/y_m). Then the triangular fuzzy sets μ_i are given by (analogously for ν_j):

$$\mu_1(x) = \begin{cases} \frac{x_2 - x}{x_2 - x_1} & \text{if } x_1 \le x < x_2 \\ 0 & \text{otherwise} \end{cases}$$

$$\mu_n(x) = \begin{cases} \frac{x_n - x}{x_n - x_{n-1}} & \text{if } x_{n-1} \le x < x_n \\ 0 & \text{otherwise} \end{cases}$$

This means that we do not admit trapezoidal membership functions at the boundaries of the interval. The fuzzy sets μ_i for $1 < i < n$ are defined as in (1).

In this section we consider the automatic adaption of fuzzy systems of this type to a given data set, where $\top$ is either the minimum or the product. Given a set of samples $S \subset X \times Y \times Z$ drawn from a function $f : X \times Y \to Z$, that is $\forall\,(x,y,z) \in S : f(x,y) = z \pm \varepsilon$, the minimisation of the approximation error of a fuzzy system $\hat{f}$

$$e(\mathbf{x},\mathbf{y},\mathbf{z};S) = \sum_{(x,y,z)\in S} (\hat{f}(x,y) - z)^2 \qquad (3)$$

is a nonlinear optimisation task. In the following, we propose an alternating optimisation method that minimises a locally scaled error function (3). The method works for arbitrary dimensions DIM $\in \mathbb{N}$ and is not restricted to the two-dimensional case, however, for the sake of simplicity we present the method with DIM $= 2$.

4.1
Partitioning the input space

Due to the restrictions on our fuzzy sets μ_i and ν_j, we have a natural partitioning of the input space $X \times Y$ into rectangular areas (or hyperboxes in arbitrary dimensions). Figure 2 illustrates this in case of a fuzzy system with $n = 4$ and $m = 5$. Within each rectangle $R_{i,j} = \{(x,y) \in X \times Y | x_i \le x < x_{i+1} \wedge y_j \le y < y_{j+1}\}$ the output value $\hat{f}(x,y)$, $(x,y) \in R_{i,j}$, is fixed by the adjacent 2^{DIM} rules only, because the other rules have a zero membership degree in this area. In Fig. 2 the four rules for the shaded rectangle $R_{2,3}$ are

if x is about x_2 and y is about y_3 then z is $z_{2,3}$
if x is about x_2 and y is about y_4 then z is $z_{2,4}$
if x is about x_3 and y is about y_3 then z is $z_{3,3}$
if x is about x_3 and y is about y_4 then z is $z_{3,4}$

The participating fuzzy sets are drawn with thick lines in Fig. 2.

Within each rectangle $R_{k,l}$ the function $\hat{f}$ depends on eight numbers (in general $2^{\mathrm{DIM}+1}$ numbers): x_k, x_{k+1}, y_l, y_{l+1}, $z_{k,l}$, $z_{k,l+1}$, $z_{k+1,l}$ and $z_{k+1,l+1}$.

We define

$$\hat{g}_{k,l}(x,y) := \hat{f}|_{R_{k,l}}(x,y) = \frac{\sum_{i=k}^{k+1}\sum_{j=l}^{l+1} \top(\mu_i(x),\nu_j(y)) \cdot z_{i,j}}{\sum_{i=k}^{k+1}\sum_{j=l}^{l+1} \top(\mu_i(x),\nu_j(y))} \tag{4}$$

Note that within the restriction $\hat{f}|_{R_{k,l}}$ the membership degrees μ_i and ν_j are linear functions and there is no need for considering multiple cases as in the piecewise definition of μ_i and ν_j.

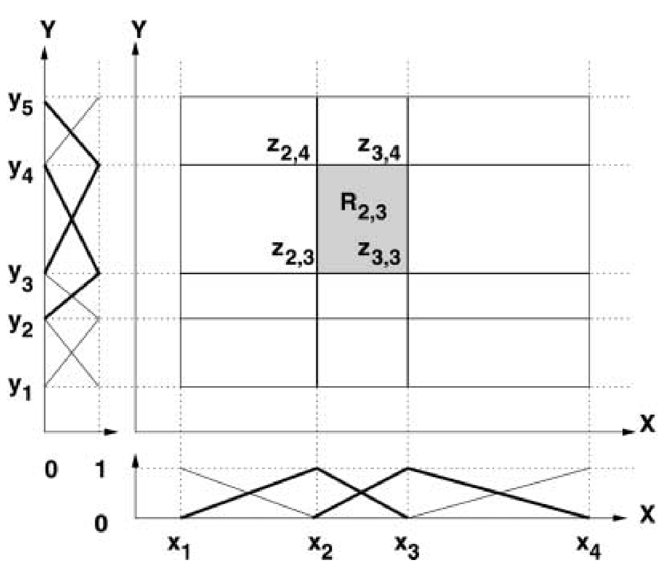

Fig. 2. The **x** and **y** vectors partition the input space $X \times Y$ into rectangular regions. The **z** vector specifies the value of $\hat{f}$ at the vertices of the rectangle

For every point (x,y) in the $X \times Y$ plane we define

$$\delta_{i,j}(x,y) := \begin{cases} 1, & \text{if } (x,y) \in R_{i,j} \\ 0, & \text{if } (x,y) \notin R_{i,j} \end{cases} \tag{5}$$

Note that for any $(x,y) \in X \times Y$ there is only one pair (i,j) such that $\delta_{i,j}(x,y)$ equals 1. Therefore, we can reformulate the function $\hat{f}$ as

$$\hat{f}(x,y) = \sum_{i=1}^{n}\sum_{j=1}^{m} \delta_{i,j}(x,y)\hat{g}_{i,j}(x,y) \tag{6}$$

and the error as

$$\begin{aligned} e(\mathbf{x},\mathbf{y},\mathbf{z};S) &= \sum_{(x,y,z)\in S} (\hat{f}(x,y) - z)^2 \\ &\overset{(6)}{=} \sum_{(x,y,z)\in S} \left(\sum_{i=1}^{n}\sum_{j=1}^{m} \delta_{i,j}(x,y)\hat{g}_{i,j}(x,y) - z\right)^2 \\ &= \sum_{(x,y,z)\in S} \sum_{i=1}^{n}\sum_{j=1}^{m} \delta_{i,j}(x,y)\left(\hat{g}_{i,j}(x,y) - z\right)^2 \end{aligned} \tag{7}$$

In the following two sections we examine the local definition of $\hat{f}$ (that is $\hat{g}_{i,j}$) in dependency of $\top$.

4.2
Using the $\top_{\mathrm{prod}}$-norm

In this section we consider the t-norm

$$\top : X \times Y \to Z, \quad (x,y) \mapsto x \cdot y \tag{8}$$

In this case the denominator of $\hat{g}_{k,l}$ in (4) conveniently equals value 1, which simplifies the definition of $\hat{g}_{k,l}$ considerably. We have

$$\begin{aligned} &\hat{g}_{k,l}(x,y) \\ &\overset{(4)}{=} \mu_k(x)\cdot\nu_l(y)\cdot z_{k,l} + \mu_k(x)\cdot\nu_{l+1}(y)\cdot z_{k,l+1} \\ &\quad + \mu_{k+1}(x)\cdot\nu_l(y)\cdot z_{k+1,l} + \mu_{k+1}(x)\cdot\nu_{l+1}(y)\cdot z_{k+1,l+1} \\ &\overset{(1)}{=} ((x_{k+1}-x)(y_{l+1}-y)z_{k,l} \\ &\quad + (x_{k+1}-x)(y-y_l)z_{k,l+1} \\ &\quad + (x-x_k)(y_{l+1}-y)z_{k+1,l} + (x-x_k) \\ &\quad \cdot (y-y_l)z_{k+1,l+1})/((x_{k+1}-x_k)(y_{l+1}-y_l)) \end{aligned} \tag{9}$$

Figure 3 shows an example of $\hat{g}_{2,3}$, which defines $\hat{f}$ within area $R_{2,3}$ (cf. Fig. 2) with $z_{2,3} = 8$, $z_{2,4} = 0$, $z_{3,3} = -2$ and $z_{3,4} = 10$.

The denominator $A_{k,l} := (x_{k+1}-x_k)(y_{l+1}-y_l)$ in (9) is the size of the area of rectangle $R_{k,l}$. Instead of minimising error (7) we locally scale the error within rectangle $R_{i,j}$ by $A_{i,j}$:

$$\begin{aligned} &\sum_{(x,y,z)\in S} \delta_{i,j}(x,y)\left(|\hat{g}_{i,j}(x,y) - z| \cdot A_{i,j}\right)^2 \\ &= A_{i,j}^2 \cdot \sum_{(x,y,z)\in S} \delta_{i,j}(x,y)\left(\hat{g}_{i,j}(x,y) - z\right)^2 \end{aligned}$$

which leads us to a modified error measure

$$e'(\mathbf{x},\mathbf{y},\mathbf{z};S) = \sum_{(x,y,z)\in S} \sum_{i=1}^{n} \sum_{j=1}^{m} \delta_{i,j}(x,y) d_{i,j}(x,y,z) \qquad (10)$$

where $d_{i,j} : X \times Y \times Z \to \mathbb{R}$ is given by

$$\begin{aligned} & d_{i,j}(x,y,z) \\ & \quad = (\hat{g}_{i,j}(x,y) - z)^2 A_{i,j}^2 \\ & \quad \overset{(9)}{=} ((x_{i+1}-x)(y_{j+1}-y)z_{i,j} + (x_{i+1}-x)(y-y_j)z_{i,j+1} \\ & \qquad + (x-x_i)(y_{j+1}-y)z_{i+1,j} + (x-x_i)(y-y_j)z_{i+1,j+1} \\ & \qquad - z(x_{i+1}-x_i)(y_{j+1}-y_j))^2 \end{aligned}$$

For a fuzzy system specified by $(\mathbf{x},\mathbf{y},\mathbf{z})$ to be a (local) minimiser of the scaled error function (10), we have a zero crossing in the first derivatives

$$\nabla_{\mathbf{x}} e' = 0, \quad \nabla_{\mathbf{y}} e' = 0, \quad \nabla_{\mathbf{z}} e' = 0 \ .$$

Due to the multiplication with the area of the rectangle, the zero gradient vectors yield a system of linear equations in each case. We minimise the error e' by alternatingly minimising with respect to $\mathbf{z}$ assuming $\mathbf{x}$ and $\mathbf{y}$ to be constant, then with respect to $\mathbf{x}$ assuming $\mathbf{y}$ and $\mathbf{z}$ to be constant, and then with respect to $\mathbf{y}$ assuming $\mathbf{x}$ and $\mathbf{z}$ to be constant. The algorithm is depicted in Fig. 4.

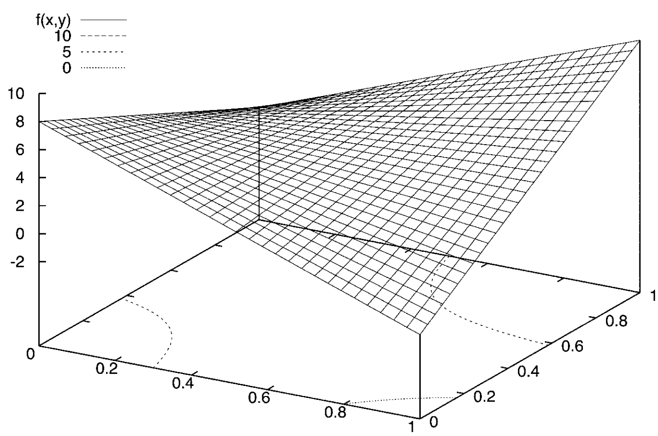

Fig. 3. Function $\hat{f}|_{R_{2,3}}$ with $\top(a,b) = a \cdot b$

Figure 5a shows the function $f(x,y) = \sin(x) \cdot x/(y+2)$ together with 200 samples on its surface, $X = Y = [0, 2\pi]$. This data set has been used to estimate a fuzzy systems with $n = m = 4$ as described above and the result after 12 iteration steps is shown in Fig. 5b. The original function f is approximated very well. Of course, since we minimise e' we cannot guarantee that the conventional least-squares error e is also minimised, however, in this example we recognised a decrease in the sum of squared error with each step.

The local error scaling seems to have no dramatic effect in Fig. 5, because the size of the rectangular regions does not differ that much. Figure 6 shows another example $f(x,y) = \exp(-(s-\frac{3}{2})^2) + \mathrm{atan}(-5 \cdot (t-5))$ where we can expect a greater variety in the area size. Note that the uniform initialisation is really poor in this example (still $n = m = 4$). Initially, we have $y_2 \approx 2.1$ and $y_3 \approx 4.2$ and the best solution is $y_2 \approx 4.4$ and $y_3 \approx 5.3$. Thus, the algorithm has to "replace" y_3 by y_2. As we can see from Fig. 6b, the algorithm has done very well after 20 iterations. The final result is remarkable, because in terms of the error function e there is a strong local minimum near the initial solution (adjust $y_3 \approx 5$ but leave y_2 half way between y_1 and y_3). During the iterations, the algorithm shortened the distance $|y_3 - y_2|$ which leads to smaller, long-stretched rectangles. The errors within these rectangles are not weighted that much so that these patches become "more flexible". It seems that in this example, the local error scaling helped to find the best solution.

```
initialise x and y uniformly
update z by means of linear equation system ∇_z e' = 0
repeat
  update x by means of linear equation system ∇_x e' = 0
  update y by means of linear equation system ∇_y e' = 0
  update z by means of linear equation system ∇_z e' = 0
until maximum number of iterations reached or
      error change drops below threshold
```

Fig. 4. Algorithm for estimating a fuzzy system

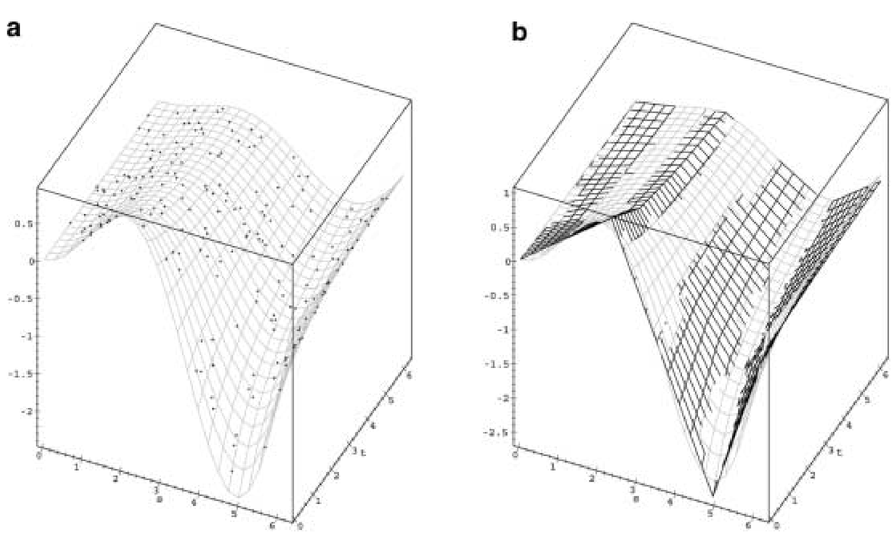

Fig. 5. The original function $f(x,y) = \sin(x) \cdot x/(y+2)$ is drawn in both images in light gray, the learned fuzzy system (after 12 iteration steps) is drawn in black

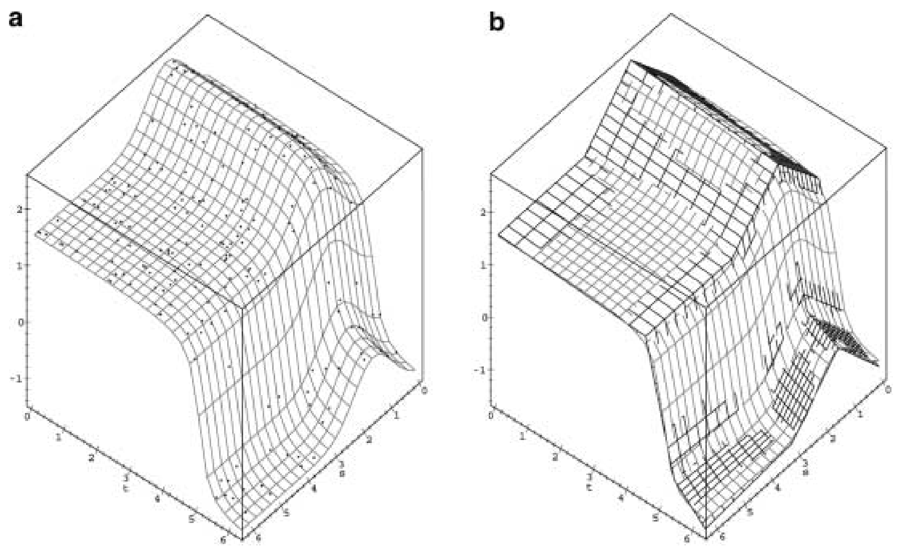

Fig. 6. The original function $f(x,y) = \exp(-(s-\frac{3}{2})^2 + \operatorname{atan}(-5\cdot\ (t-5))$ is drawn in both images in light gray, the learned fuzzy system (after 20 iteration steps) is drawn in black

4.3 Using the $\top_{\text{min}}$-norm

In this section we consider the t-norm

$$\top : X \times Y \to Z, \quad (x,y) \mapsto \min(x,y) \tag{11}$$

Due to the definition of our membership functions (1), all μ_i and v_j within $R_{k,l}$ are linear functions. When aggregating two membership functions using the $\top_{\text{min}}$-norm, we still have piecewise linear functions. The $X \times Y$ plane is once more subdivided, every rectangle consists of four triangular subregions, as shown in Fig. 7. In each of the subregions the term $\top(\mu_i(x), v_j(y))$ is a linear function, $i \in \{k, k+1\}$ and $j \in \{l, l+1\}$. However, the denominator of $\hat{g}_{k,l}$ does not evaluate to 1 but lies in the interval [1, 2], as shown in Fig. 8. But at least, within the triangular subregions the denominator is also a linear function. Thus, for each subregion we have $\hat{g}_{k,l}(x,y) = E(x,y)/F(x,y)$, where E and F are linear functions.

Figure 9 shows an example of $\hat{g}_{2,3}$, which defines $\hat{f}$ within area $R_{2,3}$ (cf. Figs. 3 and 7) with $z_{2,3} = 8$, $z_{2,4} = 0$, $z_{3,3} = -2$ and $z_{3,4} = 10$. At a quick glance $\hat{g}$ seems to be linear within the triangular subregions, but not perfectly: The linear function in each subregion is divided by a term that is 1 along the edge of the triangle that is shared with the rectangle $R_{k,l}$ and 2 at the centre of the rectangle (cf. Fig. 8).

Proceeding in a similar way as in Sect. 4.2, we minimise a locally scaled error function e' instead of (3). This time, the scaling factor has been chosen to be $A_{k,l} \cdot F(x,y)$, where F denotes the denominator of $\hat{g}_{k,l}$. Besides the multiplication by the size of the region $R_{k,l}$, we additionally multiply by a factor between 1 near the border and 2 in the centre. This means, that the resulting fuzzy systems will approximate the data especially well in the centre of the regions $R_{k,l}$ whereas it will tolerate larger errors at the border of $R_{k,l}$.

This may cause some undesired effect on the resulting fuzzy system. From the examples seen so far we can already conclude that the triples $(x_i, y_j, z_{i,j})$ – from which the fuzzy rules are created – are in general not very good approximations of the function. In order to approximate the data inside the patches $R_{k,l}$ the $z_{i,j}$ values are almost always slightly above or below $f(x_i, y_j)$. (Usually there are

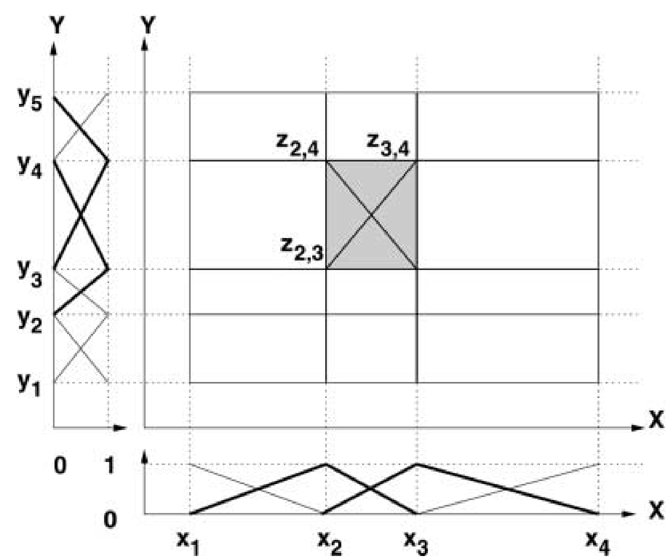

Fig. 7. Further subdivision of $R_{k,l}$ in case of $\top(a,b) = \min(a,b)$

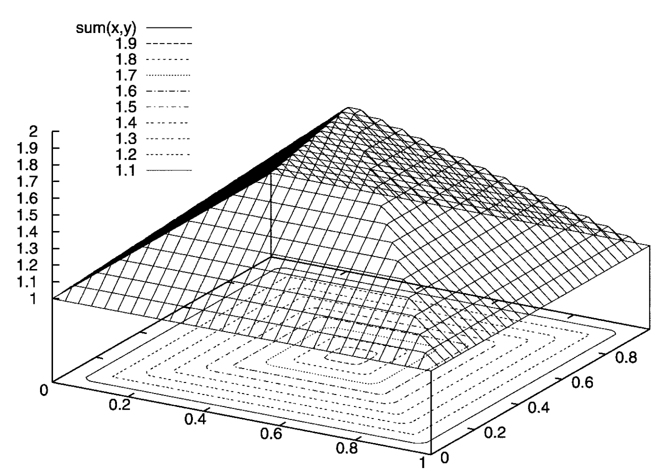

Fig. 8. Denominator of $\hat{g}_{k,l}$ with $\top(a,b) = \min(a,b)$

more data vectors inside a patch $R_{k,l}$ than near the border. Therefore it is better to tolerate larger errors near the border to minimise the total sum of squared errors.) If we use a scaling factor that emphasises the interior of the patches and does not care that much about the border, we expect that this effect becomes even more evident.

We therefore consider a second variant to optimise the fuzzy system, where we assume that the output $\hat{f}$ is a piecewise linear function – that is, we assume that the denominator $F(x,y)$ of $\hat{g}_{k,l}$ is always 1. This is not true, of course, but we have already seen in Fig. 9 that the effect of the denominator influences the curvature of $\hat{f}$ not that much. Figure 10 shows the results for both cases. The original function is the same as in Fig. 5. The figures confirm our conjecture, the deviations from the original values at the knot points have become smaller.

For these reasons, we slightly prefer the second variant (Fig. 10b), although the total sum of squared errors e does not differ significantly. Compared to the results of Fig. 5, where we used $\top_{\text{prod}}$, the total error is slightly higher when using $\top_{\min}$ (at least in our examples). Using $\top_{\min}$ instead of $\top_{\text{prod}}$ increases the computational cost by a factor of almost 4, since each of the regions $R_{k,l}$ is subdivided in four subregions (factor 2^{DIM} in arbitrary dimensions).

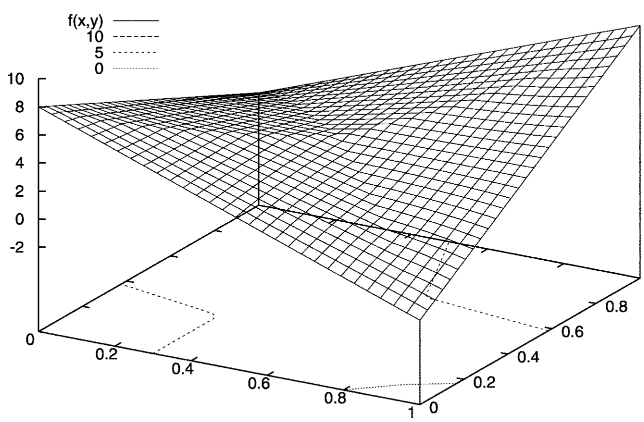

Fig. 9. Function $\hat{f}|_{R_{2,3}}$ with $\top(a,b) = \min(a,b)$

5 Conclusions

The most prominent reason for using fuzzy systems in function approximation is interpretability. However, it is of course not true, that an approximator is easily understandable, just because it uses linguistic rules. To be intuitive, the number of rules as well as the number of linguistic terms should be small, the membership degrees should add up to one and have an overlap of $\frac{1}{2}$, etc. Such a set of linguistic rules is often called a fuzzy partition. When interpreting the linguistic terms in a fuzzy system as vague values induced by an equality relation, we have a well-defined semantics of the system and easily obtain rules that fulfil the requirements for intuitive understanding. Optimisation of a fuzzy system then corresponds to selecting appropriate parameters of our equality relation, that is the scaling function. Whatever scaling function we might obtain, the resulting system remains interpretable. This is a strong point for this approach, because most of the methods usually used for parameter tuning violate the semantic constraints or have to consider them explicitly during optimisation [8]. In this paper, we have presented an algorithm for the optimisation of a fuzzy system that learns a piecewise constant scaling function, which corresponds to a fuzzy system with triangular memberships. Besides good approximation results (we have considered the case of a minimum and product t-norm for aggregation), the resulting fuzzy system preserves semantics and thus interpretability.

It is common practice to select the linguistic terms as a fuzzy partition for each input variable separately (and so does our proposed algorithm). We have illustrated, however, that this corresponds to an implicit independence assumption of the scaling functions for each variable, which is not necessarily fulfilled in applications. Due to the fact that multivariate scaling functions are computationally not tractable, it remains an open question how we can model the dependency between scaling functions efficiently, which is further complicated by the fact that this notion of independence is usually not symmetric. Since it is know from other fields, that independence assumptions

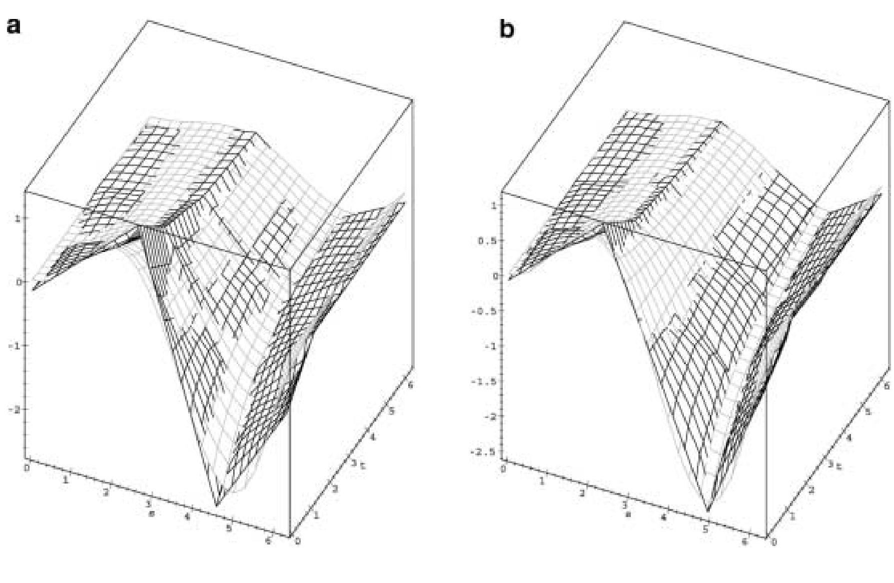

Fig. 10. The original function is the same as in Fig. 5 and is drawn in light gray, the black functions correspond to the learned fuzzy system after 12 iteration steps

can lead to powerful methods even if these assumptions do not hold in most applications (consider the naive Bayes classifier for example), another open question is whether the additional complexity will be justified by a significant improvement of the performance – without loosing too much in terms of intuitive understandability.

References

1. **Dubois D, Prade H** (1994) Similarity-based approximate reasoning. In: Zurada JM, Marks RJ, Robinson CJ (eds), Computational Intelligence Imitating Life, pp. 69–80. IEEE Press, New York
2. **Höhle U, Stout LN** (1991) Foundations of fuzzy sets, Fuzzy Sets Syst **40**, 257–296
3. **Höppner F, Klawonn F, Kruse R, Runkler T** (1999) Fuzzy Cluster Analysis. Wiley, Chichester
4. **Klawonn F** (1994) Fuzzy sets and vague environments, Fuzzy Sets Syst **66**, 207–221
5. **Klawonn F, Gebhardt J, Kruse R** (1995) Fuzzy control on the basis of equality relations – with an example from idle speed control, IEEE Trans Fuzzy Syst **3**, 336–350
6. **Klawonn F, Kruse R** (1993) Equality relations as a basis for fuzzy control, Fuzzy Sets Syst **54**, 147–156
7. **Klawonn F, Novák V** (1996) The relation between inference and interpolation in the framework of fuzzy systems, Fuzzy Sets Syst **81**, 331–354
8. **Oliveira JVD** (1999) Semantic constraints for membership function optimization, IEEE Trans Syst Man Cybernetics (Part A) **29**, 128–197
9. **Ruspini EH** (1991) On the semantics of fuzzy logic, Int J Approx Reasoning **5**, 45–88
10. **Thiele H, Schmechel N** (1995) The mutual defineability of fuzzy equivalence relations and fuzzy partitions. In: Proc Int Joint Conf of the 4th IEEE Int Conf on Fuzzy Systems and the 2nd Int Fuzzy Engineering Symposium, Yokohama, pp. 1383–1390
11. **Trillas E, Valverde L** (1984) An inquiry into indistinguishability operators. In: Skala HJ, Termini S, Trillas E (eds), Aspects of Vagueness, pp. 231–256. Reidel, Dordrecht
12. **Valverde L** (1985) On the structure of *F*-indistinguishability operators, Fuzzy Sets Syst **17**, 313–328
13. **Zadeh LA** (1971) Similarity relations and fuzzy orderings, Infor Sci **3**, 177–200