Fourth International Conference on knowledge-Based Intelligent Engineering Systems & Allied Technologies, 30th Aug-1st Sept 2000, Brighton,UK

# Obtaining Interpretable Fuzzy Models from Fuzzy Clustering and Fuzzy Regression*

**Frank Höppner** **Frank Klawonn**

University of Applied Sciences, Emden
Department of Electrical Engineering and Computer Science
Constantiaplatz 4
D-26723 Emden, Germany
e-mail alias: frank.hoeppner@ieee.org

## Abstract

*In this paper we develop an objective function-based clustering algorithm to build fuzzy models of the Takagi-Sugeno (TS) type automatically from data. In contrast to most of the TS models that can be found in the literature, we decided to use very simple input-space partitions and a higher degree of consequence polynomials (quadratic). Only in this way transparency and interpretability can be guaranteed. We also show how to derive linguistic labels for the polynomials found by the algorithm.*

**Keywords:** fuzzy model, Takagi-Sugeno type, fuzzy clustering, linguistic approximation, fuzzy c-means, fuzzy c-regression models

## 1 Introduction

Fuzzy models are widely used in many areas like expert systems, pattern recognition, or system modelling. When using rules to describe an input-output relation instead of classical function approximation techniques, this is mainly because of the transparency of the resulting fuzzy model. Fuzzy models are especially useful when facing inter-disciplinary problems, where the presentation of a purely mathematical description to the domain expert is not desirable. Unfortunately, this transparency aspect is neglected in many approaches that can be found in the literature. Aiming at a high approximation quality, some authors tend to allow any transformation of the fuzzy sets, ending up with a fuzzy model that is not interpretable any longer.

*This work was supported by the Deutsche Forschungsgemeinschaft (DFG) under grant no. Kl 648/1-1.

In this paper we deal with fuzzy models of the Takagi-Sugeno type [8], consisting of rules like

$$R: \text{if } x \text{ is } \mu \text{ then } f(x) \approx \sum_{j=0}^{q} a_j x^j$$

where $\mu$ denotes the fuzzy set in the premise and $a \in \mathbb{R}^{q+1}$ the coefficient tuple of the consequence polynomial. With $q = 0$ we obtain rules similar to the Mamdani type rules that read like if $x$ is low then $f(x)$ is high. With Mamdani rules, both linguistic terms low and high represent fuzzy sets, here high denotes $a_0 \in \mathbb{R}$.

If we choose triangular functions for the premise fuzzy sets $\mu$ and constant terms for the conclusions ($q = 0$), one can easily see that the resulting model becomes a piecewise linear function. (An objective function-based algorithm to derive an optimized model automatically from data for this case can be found in [5].) To increase approximation quality we can either increase the complexity of the premise fuzzy sets (others than triangular memberships) or we increase the degree of polynomials in the conclusion. It can be shown [7, 1] that a given function can be reconstructed only by selecting appropriate premise fuzzy sets. However, if we build a linguistic rule base the user does not "see" the premise fuzzy sets but only the linguistic terms. Let us consider two single-input single-output functions $f$ and $g$ as shown in figure 1. If we approximate the different functions $f$ and $g$ in this way we would get very different premise fuzzy sets for, let us say the linguistic term x is zero in both rule sets. But the user will identify both fuzzy sets since the same linguistic term has been used. Fine-tuning premise fuzzy sets in a fuzzy model that uses linguistic terms is therefore not transparent to the reader.

0-7803-6400-7/00/$10.00 ©2000 IEEE

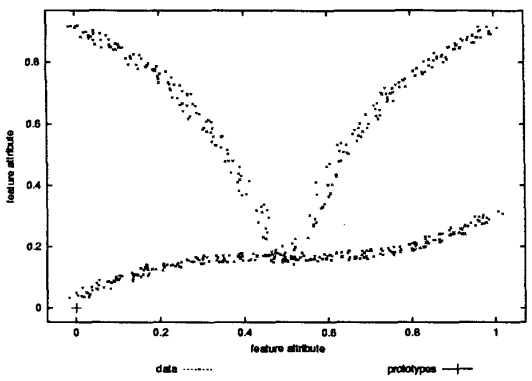


Figure 1: Functions $f$ (lower) and $g$ (upper).

Therefore we choose the other possibility, increase the polynomial degree in the conclusion. This requires new types of linguistic terms, describing the "shape of a function", for instance

if x is nearly zero
then f(x) is absolutely constant zero
then g(x) has a steep local minimum near zero

This corresponds much more to the notion of intuitive fuzzy rules than the previously discussed approach, because we use the same fuzzy set in the premise and reflect the different behaviour in the consequence. To build these rules automatically from data, we make use of fuzzy clustering techniques, which will be introduced in brevity in the next section, see [6] for a thorough overview. In section 3 we combine two clustering methods [2, 4] to obtain a new algorithm that yields a certain type of TS models as output. In section 4 we discuss the linguistic transformation of second-degree polynomials into descriptive linguistic terms. We give an example of an automatically generated linguistic rule base in section 5.

# 2 Objective Function-Based Fuzzy Clustering

Subdividing a set $X = \{x_1, ..., x_n\} \subset \mathbb{R}^p$, $p \in \mathbb{N}$, of data objects into groups of similar data objects is called clustering. Given a 'model' $P$ for clusters we can define a distance $d : \mathbb{R}^p \times P \to \mathbb{R}_0^+$ between data objects and cluster models or 'prototypes', yielding $d(x, k) = 0$ if data object $x$ matches model $k$ perfectly. For example, the popular fuzzy c-means (FCM) algorithm [2] uses points as elements of $P_{\mathrm{FCM}} := \mathbb{R}^p$ for prototypes and the Euclidean distance measure as $d$. The relationship between data objects and clusters $\{k_1, ..., k_c\}$ is modelled by means of a 'membership matrix' $U$, where each $u_{i,j}$ denotes the degree of belongingness of data object $x_j$ to cluster $k_i$. Having fixed the number of prototypes (and thus the dimension of $U$) one can perform clustering by means of minimizing the objective function

$$J = \sum_{j=1}^{n} \sum_{i=1}^{c} u_{i,j}^m d_{i,j}^2 \qquad (1)$$

under the 'probabilistic constraint'

$$\forall j \in \mathbb{N}_{\leq n} : \quad \sum_{i=1}^{c} u_{i,j} = 1 \qquad (2)$$

which is necessary to avoid the trivial solution $U \equiv 0$. The so called 'fuzzifier' $m \in \mathbb{R}_{>1}$ influences the 'fuzziness' of the final partition, with $m \to \infty$ membership degrees become totally fuzzy $u_{i,j} \mapsto \frac{1}{c}$, with $m \to 1$ they become more crisp $u_{i,j} \mapsto \{0, 1\}$.

The objective function (1) is minimized by alternating optimization (AO), that is, $J$ is first minimized with respect to membership degrees $u_{i,j}$ (considering prototypes to be constant) and then with respect to prototypes $k_i$ (considering membership degrees to be constant). In case of FCM there are closed-form solutions for both minimization steps. In this paper, we also utilize the fuzzy c-regression models (FCRM) algorithm [4], 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, $P_{\mathrm{FCRM}} := \mathbb{R}^{q+1}$ where $q$ is the degree of the polynomials. The distance of a data object, consisting of input value $x$ and output value $y$, to the polynomial $h$ is defined as $|y - h(x)|$. Due to the limited space, we refer to the literature for detailed descriptions of the algorithms.

# 3 Combining FCM and FCRM

If we consider fuzzy models as shown in the introductory example, we expect the function to *behave near zero as it has been described.* Again, many systems in the literature allow overlapping premise fuzzy sets for higher-order TS models. In this 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 sets have a large support of 1 and thus there is only one rule applicable at the same time. 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 even 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 $1 < m \leq 1.5$)[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 [4]. If we combine both algorithms, we obtain a fuzzy clustering/regression algorithm where each cluster can be interpreted as a rule in a TS model.

Since both algorithms are objective function-based, their combination is straightforward. The new *fuzzy model* (FM) algorithm uses $P = P_{\text{FCM}} \times P_{\text{FCRM}}$ and $d = d_{\text{FCM}} + d_{\text{FCRM}}$, where $P_{\text{FCM}} = \mathbb{R}^r$ and $r$ denotes the dimensionality of the input space only. In case of real-valued functions the data objects are pairs $(x, y) \in \mathbb{R}^2$ and we use $P_{\text{FCM}} = \mathbb{R}$ for a partition of the $x$-values. Since there are no dependencies between the parameters of the FCM and FCRM prototypes, the same prototype update equations hold for the combined algorithm. Nevertheless, FCM centres and FCRM 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].)

Figure 2 shows the result of the FM algorithm when approximating $f$ and $g$ using $c = 3$. Although it is not shown which polynomial belong to which function one can guess it easily from the figure. Of course, polynomials of degree 2 cannot match the original functions (locally) perfect, but they catch the behaviour or shape of the function very well.

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

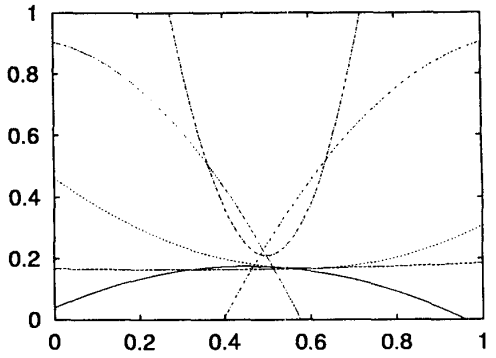


Figure 2: FM algorithm applied to $f$ and $g$.

# 4 Linguistic Model Description

We assume real-valued functions in this section. The FM algorithm yields $c$ clusters, each consisting of a centre $v_i \in \mathbb{R}$ and a polynomial $h_i(x) = \sum_{k=0}^{2} a_k x^k$. From the data objects and the clusters we construct histograms

$$\begin{aligned} H_{X_i}(x) &= \max\{u_{i,j} \mid x = x_j, j = 1..n\} \\ H_{Y_i}(y) &= \max\{u_{i,j} \mid y = y_j, j = 1..n\} \\ H_{h'(X_i)}(d) &= \max\{u_{i,j} \mid d = h'(x_j), j = 1..n\} \end{aligned}$$

and generate approximating convex fuzzy sets as described in [6]. The resulting fuzzy sets $\mu_{X_i}$, $\mu_{Y_i}$ and $\mu_{h'(X_i)}$ denote the support of the rule, the output values of the rule, and its derivative which is approximated with the help of the polynomial $h$.

From a list of landmark values we build a fixed dictionary of fuzzy sets for linguistic approximation. The dictionary contains single values (triangular membership functions) as well as intervals (trapezoidal membership functions). A list of values $\{-1, 0, 1\}$ with associated labels negative one, zero, one, for instance, would result in the singletons $-1$, 0, 1 and the intervals (trapezoidals) $[-1, 0]$, $[0, 1]$ and $[-1, 1]$. The landmark values specify the granularity to be used in the rules. Since the vocabulary is fixed, the user can be sure that the same linguistic term describes approximately the same fuzzy set even in different rule bases.

The landmark values that have been chosen to approximate $\mu_{X_i}$ are used to generate the linguistic term for the premise fuzzy set, those of $\mu_{Y_i}$

are used to describe the range of output values of this rule. The linguistic terms that characterise the shape of the local function are derived from $\mu_{h'(X_i)}$ as follows: Let us denote the leftmost and rightmost landmark value of the approximation of the fuzzy set $\mu_{h'(X_i)}$ by $d_i^l$ and $d_i^r$, resp. If $d_i^l = d_i^r$ the fuzzy set $\mu_{h'(X_i)}$ is characterized by a singleton, the derivative is nearly constant and the function therefore nearly linear. Furthermore, if this singleton corresponds to the zero singleton, the linear function is almost constant. Otherwise, i.e. $d_i^l \neq d_i^r$, we have a non-linear shape of the function. If the zero singleton is included in the trapzoidal fuzzy set the derivative has a zero passage and the function therefore a local extremum. Some linguistic labels are summarized in table 1.

| shape | condition |
|---|---|
| constant | $d_i^l = d_i^r = 0$ |
| linear | $d_i^l = d_i^r$ |
| quadratic | $d_i^l \neq d_i^r$ |
| increasing | $d_i^l > 0 \wedge d_i^r > 0$ |
| decreasing | $d_i^l < 0 \wedge d_i^r < 0$ |
| extremum | $d_i^l \neq d_i^r \wedge \text{sign}(d_i^l) \neq \text{sign}(d_i^r)$ |

Table 1: Development of linguistic terms.

Linguistic hedges, like absolutely zero or more or less zero are used to further specify the shape of a fuzzy set.

# 5 Example

Using a fixed vocabulary with singletons at $\frac{i}{10}$, $i = 0..10$, for the input and output space the described system generated the following rules for the functions $f$

| if x is | then f(x) is |
|---|---|
| close to [0,0.3] | gently increasing to 0.2 |
| more or less [0.4,0.6] | is absolutely constant at 0.2 |
| about [0.7,1.0] | gently increasing to 0.3 |

and $g$

| if x is | then f(x) is |
|---|---|
| about [0,0.3] | moderately convex decreasing to 0.6 |
| about [0.4,0.6] | has a steep local minimum at 0.2 |
| about [0.7,1] | moderately convex increasing to 0.8 |

# 6 Conclusions

We have combined two fuzzy clustering algorithms to get a new fuzzy clustering algorithm that constructs a Takagi-Sugeno type fuzzy model automatically from data. We have shown how to build readable, transparent rule bases automatically from these models. Thus the proposed system yields a qualitative functional description of the input-output relation within a data set. This kind of rules are useful in many applications, for instance with short term weather forecast or material science.

For more information and implementation see `http://www.et-inf.fho-emden.de/~dmlab`.

# References


[1] P. Bauer, E. Klement, A. Leikermoser, and B. Moser. *Fuzzy Systems in Computer Science*, chapter Interpolation and approximation of real input-output functions using fuzzy rule bases, pages 245–254. Vieweg, Braunschweig, 1994.

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

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

[4] R. J. Hathaway and J. C. Bezdek. Switching regression models and fuzzy clustering. *IEEE Transactions on Fuzzy Systems*, 1(3):195–204, Aug. 1993.

[5] F. Höppner. Piecewise linear function approximation by alternating optimization. In *Proc. of the 8th Int. Conf. on Information Processing and Management of Uncertainty in Knowledge Based Systems (IPMU)*, Madrid, Spain, July 2000. To appear.

[6] F. Höppner, F. Klawonn, R. Kruse, and T. Runkler. *Fuzzy Cluster Analysis.* John Wiley & Sons, Chichester, England, 1999. `http://fuzzy.cs.uni-magdeburg.de/clusterbook`.

[7] J. Lee and S. Chae. Analysis on function duplicating capabilities of fuzzy controllers. *Fuzzy Sets and Systems*, 56:127–143, 1993.

[8] T. Takagi and M. Sugeno. Fuzzy identification of systems and its application to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, 15:116–132, 1985.