Fuzzy Models Synthesis with Kernel-Density-Based Clustering Algorithm

Szymon Łukasik, Piotr A. Kowalski, Małgorzata Charytanowicz and Piotr Kulczycki
Systems Research Institute, Polish Academy of Sciences
ul. Newelska 6, PL-01-447 Warsaw, Poland
{slukasik, pakowal, mchmat, kulczycki}@ibspan.waw.pl

Abstract

Data clustering constitutes at present a commonly used technique for extracting fuzzy system rules from experimental data. Detailed studies in the field have shown that using above-mentioned method results in significantly reduced structure of fuzzy identification system, maintaining at the same time its high modelling efficiency. In this paper a clustering algorithm, based on a kernel density gradient estimation procedure applied for fuzzy models synthesis, is presented. It consists of two stages: data elements relocation and their division into clusters. The method is automatic, unsupervised, and does not require any assumptions concerning the desired number of fuzzy rules. The results of experimental evaluation show that the algorithm under consideration achieves relatively high performance when compared to the standard techniques frequently applied in similar class of problems.

1. Introduction

Fuzzy modelling (or fuzzy identification), introduced by Takagi and Sugeno [1], is currently a very popular method of capturing a system's behaviour using its available measurable characteristics. This approach has been successfully applied in several multi disciplinary problems i.e. prediction tasks [2], controllers' design [3] or systems analysis [4]. Inference by means of fuzzy modelling is based on employing a set of fuzzy *if–then* rules, which reflects the input-output relationship of the modelled system. Usually typical Takagi-Sugeno rules are used, where the consequent part is described by non-fuzzy equations with the input variables, for example: *if error is high and error_derivative is low then control=α·error + β·error_derivative*.

The problem of extracting rules from data has been approached using different techniques, such as genetic algorithms [5], neuro-fuzzy methods [6] or criteria based on information theory [7]. Since one wishes to find the compact representation of a fuzzy relationship, clustering is also used often, either as a stand-alone procedure or as part of another method. The most classic approach is subtractive clustering developed by Chiu [8], although some other promising techniques, such as evolving clustering [9] or Gustafson-Kessel clustering [10] were also applied.

The aim of the paper is to present an alternative method of obtaining rules' prototypes by means of clustering based on nonparametric density gradient estimation. A detailed description of the clustering procedure can be found in work [11] and preliminary study on its application in fuzzy modelling was presented in [12].

The estimation is performed using kernel density estimators (KDE) [13], [14]. The kernel estimator of unknown density function f for the n-dimensional probabilistic variable U with the sample $u_1, u_2, \ldots, u_m$, kernel K and bandwidth (smoothing parameter) h, is defined as the following function:

$$\hat{f}(u) = \frac{1}{mh^n} \sum_{i=1}^{m} K\left(\frac{u - u_i}{h}\right) . \qquad (1)$$

For the kernel function K introduced in the above definition one can use either radial

$$K(u) = c\,\kappa(||u||) \qquad (2)$$

or product kernel

$$K(u) = c \prod_{j=1}^{n} \kappa(u_j) . \qquad (3)$$

In the latter case each one-dimensional kernel κ is associated with the individual bandwidth h_j with $j = 1, \ldots, n$, and consequently h^n in formula (1) is equal to $\prod_{j=1}^{n} h_j$. Furthermore the factor c normalises the integral of the kernel K to 1.

Applying radial kernel ensures a relatively higher estimation quality. The drawback of this approach

978-0-7695-3305-6/08 $25.00 © 2008 IEEE
DOI 10.1109/FSKD.2008.139

is the need to perform linear transformation of data with differently-scaled dimensions; furthermore the calculation of bandwidth has to be performed using least-squares cross-validation, not so suitable from the applicational point of view. Moreover, although estimation based on the product kernel suffers from slightly lower effectiveness, it profits from the use of the simple and automatic plug-in method of the selection of the bandwidths h_j, and also much easier integration and differentiation procedures. For above mentioned reasons in the further part of this paper the product kernel will be considered and applied. More detailed information about the practical issues of KDE methods and usage examples can be found in [14], [15].

The clustering algorithm being considered here will use the gradient ∇f estimation. The idea of the proposed concept is based on the Fukunaga method [16], but the rule extraction from clusters centers is similar to the one used in standard subtractive clustering.

This paper is organised as follows. The second section is devoted to the short description of the clustering algorithm based on KDE considered in this paper. It includes as well the explanation of fuzzy rules construction process based on obtained clusters centers. The subsequent section contains results of the computational experiments and performance comparisons with existing methods. Finally, some concluding remarks on the clustering technique under investigation, its effectiveness, applicational issues and directions for future work are presented.

2. Fuzzy identification with kernel density estimation clustering

2.1. Algorithm description

The algorithm is based on the natural assumption that each cluster can be represented by the local maximum of the kernel estimator of the probability density function $\hat{f}$, obtained for the considered n-dimensional data elements $u_1, u_2, \dots, u_m$. However, instead of the direct analysis of a density function, its gradient is used here. The algorithm consists of two stages: relocation of data elements and their division into clusters.

In the first phase of the algorithm each element is moved along a direction defined by the gradient $\nabla \hat{f}$, according to the following equation:

$$u_i^{(k+1)} = u_i^{(k)} + b\frac{\nabla \hat{f}(u_i^{(k)})}{\hat{f}(u_i^{(k)})} \qquad \text{for } i = 1, 2, \dots, m \quad (4)$$

taking the original data set as a procedure starting point, i.e.

$$u_i^{(0)} = u_i \qquad \text{for } i = 1, 2, \dots, m \; . \quad (5)$$

The parameter $b = [b_1, b_2, \dots, b_n]^{\mathrm{T}}$, with $b_j > 0$, defines the "speed" of data movement:

$$b_j = \frac{h_j^2}{3} \; . \quad (6)$$

The first phase of the KDE clustering algorithm ends if the following stop condition is fulfilled:

$$\frac{||D^{(k)} - D^{(k-1)}||}{D^{(0)}} \leq 0.001 \; , \quad (7)$$

where $D^{(k)} = \sum_{i=1,j=i}^{m} d_{ij}^{(k)}$ with $d_{ij}^{(k)} = ||u_i^{(k)} - u_j^{(k)}||^2$ constitutes a sum of distances in each of k algorithm's iterations.

The second stage of the KDE clustering algorithm starts when condition (7) is found to be true. First, a sample consisting of elements' distances $d_{ij}^{(k)}$ for $i = 1, 2, \dots, m$ and $j = i, i+1, \dots, m$ is constructed. For large data sets it is advised to use at this stage some data condensation algorithm (e.g. [17]). Then, the smallest argument d_{min} for which the KDE function, calculated for the sample, assumes its local minimum (excluding possible minimum in zero) should be identified. The value d_{min} serves as a cluster distinction parameter, i.e. two points u_i and u_j belong to the same cluster if $d_{ij}^{(k)} \leq d_{min}$. The last step of this algorithm consist of mapping points to proper clusters according to the above-formulated rule. Details of the clustering procedure can be found in paper [11].

2.2. Rules base construction

Now assume that the data set under consideration represents input $x_1, x_2, \dots, x_m$ and $y_1, y_2, \dots, y_m$ output values. As a result data dimension $n = n_x + n_y$, where n_x denotes a dimension of inputs space and n_y - dimension of outputs space. Using presented clustering procedure one obtains centers of gravity for C clusters i.e. $u_{c_j} = \{x_{c_j1}, \dots, x_{c_jn_x}, y_{c_j1}, \dots, y_{c_jn_y}\}$ with $j = 1, 2, \dots, C$. Those centers can then serve as a basis for the rule generation process as shown in [8].

Therefore each cluster represents a fuzzy rule with membership function which, for given input vector x, is defined by:

$$\mu_{c_j}(x) = e^{-\frac{4}{r_a^2}||x - x_{c_j}||^2} \; . \quad (8)$$

The parameter $r_a > 0$ used in (8) allows the designer to control the generalisation ability of a resulting fuzzy inference system. One can adjust it to his needs, it was observed however that in most cases satisfactory results were obtained for $r_a = \frac{1}{C}$.

The output of fuzzy modelling system is defined consequently as:

$$y = \frac{\sum_{j=1}^{C} \mu_{c_j}(x) y_{c_j}}{\sum_{j=1}^{C} \mu_{c_j}(x)} \,. \tag{9}$$

Furthermore, as shown in paper [8] significantly better accuracy of fuzzy modelling can be achieved if one allows each output cluster in formula (9) to be a linear function of input variables. Given the set of training data it adds only a relatively easy-to-solve simple linear least-squares estimation problem to the process of rules set construction. Obtained consequents with associated membership functions (8) and C rules base constitute a complete fuzzy modelling system.

To conclude, presented method, when applying any standard bandwidth selection procedure, offers automatic definition of the number of clusters directly resulting model complexity. However one can adjust it to one's needs by altering bandwidth value – its increasing implies stronger density smoothing and smaller number of clusters obtained, with the opposite result when this value is decreased.

The next section is devoted to the experimental verification of the method investigated in this paper using some standard benchmark examples presented already in the literature.

3. Results of experimental evaluation

The algorithm was tested on several data sets representing various engineering problems where fuzzy modelling was perceived as an appropriate solving tool. Here, the examples of function approximation and synthesis of fuzzy PD controller will be presented. Moreover, the results of comparison with popular subtractive clustering and ANFIS methods will be given as well. All algorithms were used in their MATLAB® implementations.

3.1. Function approximation

At first, the matter of interest was a problem of fuzzy modelling of the following function

$$y = \frac{20}{x} sin\left(\frac{x}{20}\right) \tag{10}$$

presented already in [10]. The training set consisted of 100 data pairs $\{x, y\}$ for $x \in [1, 200]$ generated with uniform distribution and rounded to the nearest integer. Additionally, normal distributed random noise with zero mean and 0.02 variance was added to this data set.

The modelling error was defined as root mean square error (RMSE) between the noise-free testing curve $y(x)$, where $x = 1, 2, ..., 200$, and data obtained from the models synthesised using tested fuzzy identification techniques. For each method a set of parameter's values were tested. Subtractive clustering was invoked with $r_a = \{0.1, 0.2, ..., 1.0\}$, KDE clustering with bandwidth varying from optimal in range $h_j = \{0.5\, h_{j_{opt}}, 0.6\, h_{j_{opt}}, ..., 1.4\, h_{j_{opt}}\}$ and ANFIS with initial rules number r from 2 to 11. Experiments were repeated 10 times, with different random number generator's seed. Average rules number and root mean square errors (with its standard deviation) for compared methods are reported in Tab. 1. Best algorithms' configurations (in terms of obtained average RMSE) are shown in Tab. 2.

Table 1. Average performance of fuzzy modelling methods for function (10) approximation.

Method	$\overline{\mathbf{Rules}}$	$\overline{\mathbf{RMSE}}$	$\overline{\sigma}_{\mathbf{RMSE}}$
Subtractive	5.42	0.152	0.135
ANFIS	6.50	0.058	0.009
KDE-based	3.78	0.058	0.012

Table 2. Average performance of best fuzzy modelling methods' configurations.

Method	Config.	$\overline{\mathbf{Rules}}$	$\overline{\mathbf{RMSE}}$
Subtractive	$r_a = 0.5$	3.0	0.038
ANFIS	$r = 3$	3.0	0.043
KDE-based	$h_j = 0.9\, h_{j_{opt}}$	4.2	0.050

It can be seen that introduced KDE-based clustering technique is the best one in terms of average performance and stability to the choice of the algorithm's parameters. One should notice however that when this choice is done conveniently, other methods can be found fractionally superior.

3.2. Fuzzy PD controller synthesis

In paper [18] subtractive clustering was effectively applied for the rules set reduction of the fuzzy logic PI (FPIC) and PD (FPDC) feedback controllers. Input-output data obtained from a controller consisting of 49 rules were used as training values for a cluster analysis algorithm which generated a less complex model of the considered system. Furthermore, the reduced controller was shown to maintain almost the same level of performance as the original one.

The same approach will be under consideration here, with difficult to control marginally stable process, described by following transfer function:

$$G(s) = \frac{1}{s(s+1)} \; . \tag{11}$$

Using the reference, properly tuned fuzzy logic PD controller, with rules base consisting of 49 elements, a set of training data was created. It consists of 525 $\{\Delta e_N, e_N, u_n\}$ triplets representing control system input-output signal values for the square wave reference r. The fuzzy modelling algorithms were performed on a part of this set – 105 points sampled at regular intervals. Then the resulting models were evaluated with the same reference signal. The test was conducted for introduced KDE-based clustering technique (with default settings), subtractive clustering and ANFIS. For both referenced methods two configurations with approximately the same number of rules were under consideration. Results obtained during methods' experimental evaluation, taking into account RMSE, settling time $t_{s,5\%}$ and overshoot OV, are presented in Tab. 3. The response characteristics of synthesised fuzzy logic controllers are shown on Fig. 1.

Table 3. Performance of methods for fuzzy PD controller rules base reduction.

FPDC	Rules	RMSE	$t_{s,5\%}$[s]	OV[%]
Initial	49	0.371	7.168	0.0
Subtractive	3	0.134	8.000	0.0
	9	0.137	9.204	0.0
ANFIS	4	0.121	7.205	10.1
	9	0.147	7.000	36.0
KDE-based	5	0.114	5.924	7.7

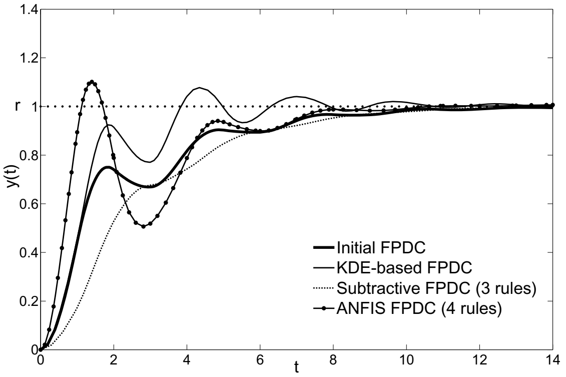

Figure 1. Comparison of selected fuzzy logic controllers' time responses.

Reducing initial complex structure of fuzzy PD controller brings positive effects in all considered examples. Reduced controllers offer smaller root mean square modelling error, quicker settling time, with the application of compact set of fuzzy rules. Here, considering obtained results, investigated KDE clustering technique was found to be superior - it offers the smallest RMSE, the quickest settling time (with a little overshoot though) and a small, 5 fuzzy rules base. It is worth to mention that the same encouraging remark can be made for the outcome of experiments with fuzzy PI controllers.

4. Conclusion and directions for further work

The aim of this paper was to present a novel approach to fuzzy rules synthesis via cluster analysis. The concept is based on the nonparametric kernel density estimation. The procedure consists of two phases: points relocation and their assignment into appropriate clusters. The method under investigation allows to determine a set of rule prototypes, without arbitrary assumption concerning their number and also a need to select any user-defined parameters. Moreover, the high computational burden of calculations needed can be easily and effectively reduced, with parallel processing, as shown in paper [19].

The proposed method was tested on various fuzzy modelling cases and was shown to achieve similar or even better level of performance as other popular modelling techniques. Further research in the subject could concern the customisation of the rule construction process (e.g. by automatic preselection of the generalisation factor r_a) to the specific features of the kernel density estimation.

References

[1] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 15, pp. 116–132, 1985.

[2] V. Olej and J. Krupka, "Prediction of gross domestic product development by Takagi-Sugeno fuzzy inference systems," in *IEEE Proceedings of International Conference on Intelligent Systems Design and Applications, Wroclaw*, 2005, pp. 186–191.

[3] B. Kadmiry and D. Driankov, "A fuzzy gain-scheduler for the attitude control of an unmanned helicopter," *IEEE Transactions on Fuzzy Systems*, vol. 12, pp. 502–515, 2004.

[4] A. Trabelsi, F. Lafont, M. Kamoun, and G. Enea, "Identification of nonlinear multivariable systems by adaptive fuzzy Takagi-Sugeno model," *International Journal of Computational Cognition*, vol. 2, pp. 137–153, 2004.

[5] S. E. Papadakis and I. B. Theocharis, "A GA-based fuzzy modeling approach for generating TSK models," *Fuzzy Sets and Systems*, vol. 131, pp. 121–152, 2002.

[6] J.-S. R. Jang, "ANFIS: Adaptive-network-based fuzzy inference system," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 23, pp. 665–684, 1993.

[7] J. Yen and L. Wang, "Application of statistical information criteria for optimal fuzzy model construction," *IEEE Transactions on Fuzzy Systems*, vol. 6, pp. 362–372, 1998.

[8] S. L. Chiu, "Fuzzy model identification based on cluster estimation," *Journal of Intelligent and Fuzzy Systems*, vol. 2, pp. 267–278, 1994.

[9] Q. Song and N. Kasabov, "ECM – a novel on-line, evolving clustering method and its applications," in *Proceedings of the Fifth Biannual Conference on Artificial Neural Networks and Expert Systems, Dunedin*, 2001, pp. 87–92.

[10] K. M. Alexiev and O. I. Georgieva, "Improved fuzzy clustering for identification of Takagi-Sugeno model," in *Proceedings of the Second IEEE International Conference on Intelligent Systems*, 2004, pp. 213–218.

[11] P. Kulczycki and M. Charytanowicz, "A complete gradient clustering algorithm by kernel estimators," *In press*, 2008.

[12] S. Lukasik, P. Kowalski, M. Charytanowicz, and P. Kulczycki, "Fuzzy model identification using kernel-density-based clustering," in *Proceedings of the Sixth International Workshop on Intuitionistic Fuzzy Sets and Generalized Nets, Warsaw*, 2007, in print.

[13] M. P. Wand and M. C. Jones, *Kernel Smoothing*. London: Chapman and Hall, 1995.

[14] P. Kulczycki, *Estymatory jadrowe w analizie systemowej*. Warsaw: Wydawnictwa Naukowo-Techniczne, 2005, in Polish.

[15] ——, "Kernel estimators in industrial applications," in *Soft Computing Applications in Industry*, B. Prasad, Ed. Springer-Verlag, 2008, pp. 69–93.

[16] K. Fukunaga and L. D. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *IEEE Transactions on Information Theory*, vol. 21, pp. 32–40, 1975.

[17] S. K. Pal and P. Mitra, *Pattern Recognition Algorithms for Data Mining*. London: Chapman and Hall, 2004.

[18] S. Chopra, R. Mitra, and V. Kumar, "Analysis of fuzzy PI and PD type controllers using subtractive clustering," *International Journal of Computational Cognition*, vol. 4, pp. 30–34, 2006.

[19] S. Lukasik, "Parallel computing of kernel density estimates with MPI," *Lecture Notes in Computer Science*, vol. 4489, pp. 726–734, 2007.