Fuzzy Model Identification Using Kernel-Density-Based Clustering

Szymon Łukasik
Piotr Andrzej Kowalski
Małgorzata Charytanowicz
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

Extracting fuzzy system rules from experimental data by means of clustering constitutes at present a commonly used technique. Intense studies in the field have shown that this method leads to a significant reduction in the system's complexity, maintaining at the same time its high modelling performance. In this paper a clustering algorithm, based on a kernel density gradient estimation procedure applied for fuzzy system identification, is presented. The method is unsupervised, fully automatic and does not need any requirement concerning the assumed number of data clusters. The results of experimental evaluation show that the algorithm under investigation is in most cases superior over the standard subtractive clustering technique frequently applied in similar types of problems.

Keywords: fuzzy inference, clustering, kernel density estimation.

1 Introduction

Fuzzy identification (or fuzzy modelling), introduced by Takagi and Sugeno [1], is currently a very popular method of capturing a system's behaviour using its available quantitative characteristics. Numerous applications of this approach include, among others, prediction problems [2], controllers' design [3] and systems analysis [4].

Inference based on fuzzy modelling relies 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 of the input variables, for example: *if temperature is high and air_circulation is low then control=*$\alpha\cdot$*temperature –* $\beta\cdot$*air_circulation*.

The problem of extracting rules from data is not trivial and has been investigated 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 minimum representation of a fuzzy relationship, clustering is also used very frequently, either as a stand-alone procedure or as part of another method. The most popular approach is subtractive clustering developed by Chiu [8], although researchers' efforts resulted in some other interesting techniques, such as evolving clustering [9] or Gath-Geva clustering [10].

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 method can be found in work [11]. The estimation is performed using kernel density estimators (KDE); more information is found in e.g. books [12, 13]. Such an estimator of unknown density function f for the n-dimensional probabilistic variable U with the sample $u_1, u_2, \ldots, u_m$, the kernel K and the bandwidth h, is defined as the following function:

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

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

$$K(x) = c\,\kappa(||x||) \tag{2}$$

or product kernel

$$K(x) = c\,\kappa(x_1) \cdot \kappa(x_2) \cdot \ldots \cdot \kappa(x_n) , \tag{3}$$

where each one-dimensional kernel κ is associated with the individual bandwidth $h_1, h_2, \ldots, h_n$, and consequently h^n in formula (1) is equal to $h_1 \cdot h_2 \cdot \ldots \cdot h_n$ and the factor c normalises the integral of the kernel K to 1.

The radial kernel experiences a relatively higher estimation effectiveness. The drawback of this approach is the need to perform data linear transformation in the case of differently-scaled dimensions; moreover,

the calculation of bandwidth has to be performed using least-squares cross-validation, not so suitable from the applicational point of view. On the other hand, 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_i, and also much easier integration and differentiation procedures. More detailed information about the practical issues of KDE methods and applications are found in books [13, 14]. In the further part of this paper the product kernel will be applied.

The clustering algorithm being considered here will use the gradient ∇f estimation. The idea of the proposed concept is based on the Fukunaga method [15], 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 commonly used subtractive clustering method (Subsection 2.1); it presents the identification of cluster centers as well as fuzzy rule construction from data prototypes (Subsection 2.2). In the next section a detailed description of the clustering algorithm based on KDE, worked out in this paper, is given. The subsequent section contains results of the computational experiments and comparisons. Finally, some concluding remarks on the clustering method under investigation, its effectiveness and directions for future work, are presented.

2 Subtractive Clustering for Fuzzy Model Synthesis

Among various cluster analysis procedures the most suitable for fuzzy identification are the ones, which do not need an arbitrarily assumed number of clusters. Subtractive clustering is a method with such property, commonly used for fuzzy modelling. It also constitutes a reference for other procedures' performance. Some short description of the method, based on work [8], will be given in the following. The KDE approach investigated in this paper refers to it, both in methodological and experimental aspects.

2.1 Cluster Estimation

The algorithm for m-elements data set $u_1, u_2, \ldots, u_m$ starts with calculating data density measures:

$$D_i = \sum_{j=1}^{m} e^{-\frac{||u_i - u_j||^2}{(r_a/2)^2}} \quad \text{for } i = 1, 2, \ldots, m \ , \tag{4}$$

where $r_a > 0$ is a so-called cluster radius coefficient. A data element with the highest density value $D_{(c_1)}$ is treated as a first cluster center $u_{(c_1)}$ and density measures of points lying within radius r_b (usually assumed as $1.5r_a$) are reduced according to the following formula:

$$D_i := D_i - D_{(c_1)} e^{-\frac{||u_i - u_{(c_1)}||^2}{(r_b/2)^2}} \ . \tag{5}$$

The procedure of selecting new clusters and reducing density measures is repeated iteratively until

$$D_{(c_k)} > A D_{(c_1)} \tag{6}$$

or

$$D_{(c_k)} \leq R D_{(c_1)} \ , \tag{7}$$

where $A > 0, R > 0$ and $A > R$. If the first condition is true, the clustering ends with the obtained cluster centers $u_{(c_1)}, u_{(c_2)}, \ldots, u_{(c_k)}$. In the case when inequality (7) is true, one has to check whether the following holds:

$$\frac{d_{min}}{r_a} + \frac{D_{(c_k)}}{D_{(c_1)}} \geq 1 \ , \tag{8}$$

where d_{min} is the smallest distance between the candidate for a new cluster center $u_{(c_k)}$ and the set of previously obtained cluster centers $u_{(c_1)}, u_{(c_2)}, \ldots, u_{(c_{k-1})}$. If dependence (8) is true, the clustering procedure continues with obtained $u_{(c_k)}$; if not, then $D_{(c_k)} = 0$ and the next cluster center candidate with the biggest density is tested by conditions (6) and (7).

It should be noted that the assumed value of the parameter r_a strongly affects the number of clusters obtained by the algorithm. Generally, a large r_a yields lower number of clusters, however, a smaller r_a results in the opposite phenomenon. It allows a full control of the algorithm resolution, but on the other hand there are no general guidelines as to how this parameter should be chosen.

2.2 Generation of Rules

Assume that n-dimensional space, previously partitioned into C clusters, consists of a n_x-dimensional inputs space and a n_y-dimensional outputs space of the modelled system under consideration. As a result, each cluster center is defined as $u_{(c_k)} = \{x_{(c_k)1}, \ldots, x_{(c_k)n_x}, y_{(c_k)1}, \cdots, y_{(c_k)n_y}\}$ and can be treated as a general fuzzy rule describing the system's behaviour. A degree to which this rule is fulfilled, for given input vector x, is defined by

$$\mu_{(c_k)} = e^{-\alpha||x - x_{(c_k)}||^2} , \tag{9}$$

where $\alpha = \frac{4}{r_a^2}$. The output vector y can be calculated as

$$y = \frac{\sum_{k=1}^{C} \mu_{(c_k)} y_{(c_k)}}{\sum_{k=1}^{C} \mu_{(c_k)}} . \tag{10}$$

To build a fuzzy inference system (FIS), based on the information obtained from the clustering process presented above, one has to define a set of C fuzzy rules, taking into account each of the system's inputs and outputs, i.e. every cluster should be mapped onto each dimension of input-output space: *if* x_1 *is* $A_{(c_i)1}$ *and* x_2 *is* $A_{(c_i)2}$ *and ... and* x_{n_x} *is* $A_{(c_i)n_x}$ *then* y_1 *is* $B_{(c_i)1}$ *and* y_2 *is* $B_{(c_i)2}$ *and ... and* y_{n_y} *is* $B_{(c_i)n_y}$ for $i = 1, 2, \ldots, C$. Thus, membership functions for input variables are defined in the following manner:

$$A_{(c_i)j}(x_j) = e^{-\alpha(x_j - x_{(c_i)j})} \tag{11}$$

and their consequents as a set of singletons

$$B_{(c_i)j} = y_{(c_i)j} . \tag{12}$$

It was shown in paper [8] though that significantly better accuracy of fuzzy modelling can be achieved if I-order Takagi-Sugeno rules are used, i.e. when each output cluster in formula (10) is a linear function of input variables:

$$y_{(c_i)} = G_{(c_i)} \cdot x + w_{(c_i)} . \tag{13}$$

The procedure for calculation of optimal consequent parameters, i.e. matrices $G_{(c_i)}$ and $w_{(c_i)}$, does not result in significant additional computational overhead, as they can be easily obtained – when the set of training data $\{x_1, x_2, \ldots, x_m, y_1, y_2, \ldots, y_m\}$ is given, then equation (10), with substitution from formula (13), represents a simple linear least-squares estimation problem [1].

3 Kernel Density Estimation Clustering

3.1 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 $u_1, u_2, \ldots, u_m$. 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:

$$u_i^{(k+1)} = u_i^{(k)} + b\frac{\nabla\hat{f}(u_i^{(k)})}{\hat{f}(u_i^{(k)})} \quad \text{for } i = 1, 2, \ldots, m \tag{14}$$

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

$$u_i^{(0)} = u_i \quad \text{for } i = 1, 2, \ldots, m\ . \tag{15}$$

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

$$b_i = \frac{h_i^2}{3}\ . \tag{16}$$

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

$$\frac{||D^{(k)} - D^{(k-1)}||}{D^{(0)}} \leq 0.001\ , \tag{17}$$

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 (17) is true. First, a sample consisting of elements' distances $d_{ij}^{(k)}$ for $i = 1, 2, \ldots, m$ and $j = i, i+1, \ldots, m$ is constructed. 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 found. 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 are found in paper [11].

3.2 Application for Fuzzy System Modelling

Now assume that the data set under consideration represents input-output values. When one obtains the mapping of elements to C clusters, by use of the earlier-presented algorithm, it is easy to calculate the centers of gravity for clusters. Those centers can then serve as a basis for the rule generation process as shown in Section 2.2. The parameter r_a is used here only in equation (9) and allows the designer to control the generalisation ability of a resulting fuzzy inference system. As a "rule of thumb", $r_a = \frac{1}{C}$ can be proposed.

It is important to note that the 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 – increasing it 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 benchmarks presented already in the literature.

4 Experimental Results

The algorithm was tested on three data sets representing, in order, problems of function approximation, gas furnace system input-output modelling, and synthesis of fuzzy PI controllers. Moreover, comparison with the up-to-date subtractive clustering method was also performed. Both algorithms were used in their MATLAB® implementations.

4.1 Function Approximation

Consider the classic problem [8] of fuzzy modelling of the function

$$y = \frac{sin(x)}{x} \ . \tag{18}$$

The training set consists of 50 data pairs for $x \in [-10, 10]$ generated with uniform distribution. A testing set was created in the analogical way.

The modelling error was defined as root mean square deviation (RMS) between the testing set and data obtained from the model. The results of the methods' comparison is presented in Tab. 1. It can be seen that the

Table 1: Comparison of clustering methods for fuzzy function modelling.

Method	Radius r_a	Rules Obtained	RMS
Subtractive clustering	0.5	3	0.1834
KDE-based clustering	0.33	3	0.1678

introduced KDE-based clustering technique achieves better accuracy of fuzzy modelling than subtractive clustering, providing at the same time a simple structure of the fuzzy inference system, consisting of only 3 rules.

4.2 Gas Furnace Fuzzy Model Identification

The gas furnace data set, originally presented in book [16], is a benchmark frequently used for research on fuzzy and neural modelling. The set consists of 296 elements $(x, y) \in \mathcal{R}^2$. The input signal represents the flow rate of the methane in a gas furnace, while the output of the model corresponds to the CO^2 concentration in the gas mixture flowing out of the furnace, under a steady air supply. Both signals are sampled every 9 seconds. The aim of fuzzy modelling is in this case predicting y at fixed k iteration on the basis of available knowledge of system behaviour in prior moments of time. Most studies use the values $y(k-1)$ and $u(k-4)$ for such purpose [17]. Here the same approach was applied.

A training set consists of the values $y(k-1), u(k-4), y(k)$ with $k = 5, 6, ..., 150$. For evaluation of the obtained models the rest of the data set was used. Subtractive clustering was tested with $r_a = 0.1, 0.2, ..., 1.0$ and KDE-based clustering with bandwidth varying from 50% to 150% of the optimal value. Here the best two-clustered configurations of both compared methods are presented in Tab. 2. Note that in the case of subtractive clustering the presented configuration is also the best one in the terms of RMS, however, for the KDE-based method the reported result can be further improved by changing the bandwidth value.

Table 2: Comparison of clustering methods for gas furnace modelling

Method	Radius r_a	Rules Obtained	RMS
Subtractive clustering	1	2	0.6134
KDE-based clustering	0.5	2	0.6112

Figure 1 presents a comparison of real system response and data acquired from fuzzy models synthesised via investigated clustering meth-

ods. Although both reduced inference systems' outputs follow real data closely, the RMS is in the case of KDE-based clustering a little smaller.

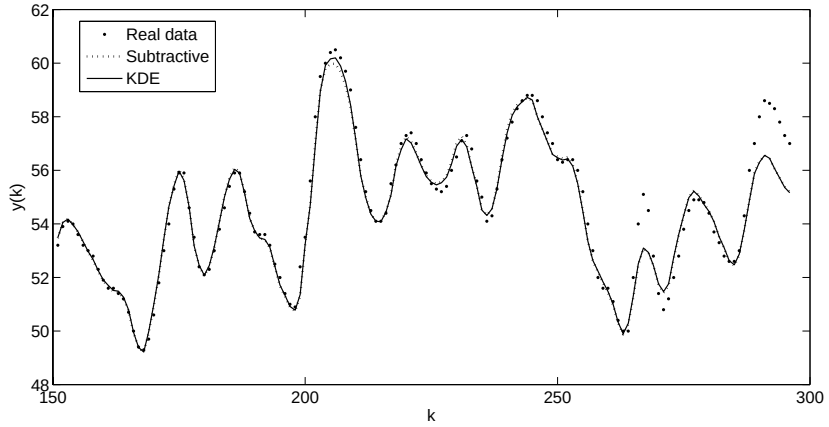

Figure 1: Results of Box-Jenkins' gas furnace data set modelling.

4.3 Fuzzy PI Controller Synthesis

In paper [18] subtractive clustering was effectively applied for the rules set reduction of the fuzzy logic PI feedback controller. It was presented there how input-output data obtained from a controller consisting of 49 rules can be used as training values for a cluster analysis algorithm which generates a less complex model of the considered system. Furthermore, the reduced controller maintains almost the same level of performance as the original one.

Assume that the following process transfer function is given:

$$G(s) = \frac{1}{(s+1)} \frac{1}{(0.2s+1)} \frac{1}{(3s+1)} \,. \tag{19}$$

Using the reference PI fuzzy logic controller, with rules base consisting of 49 elements, a set of training data was created. It consists of 926 $\{\Delta e_N, e_N, \Delta u_n\}$ triplets representing control system input-output signal values for the square wave reference x. The clustering algorithms were performed on a part of this set – a hundred points sampled at regular intervals. Then the resulting fuzzy models were evaluated using the whole training sample. The comparison of the algorithms' performance is shown in Tab. 3. The response characteristics of clustering-based fuzzy logic controllers were presented on Fig. 2.

Table 3: Performance of clustering methods for fuzzy PI reduction.

Method	Radius r_a	Rules	RMS
Initial controller	–	49	0.4824
Subtractive clustering	0.5	4	0.4742
KDE-based clustering	0.5	2	0.4643

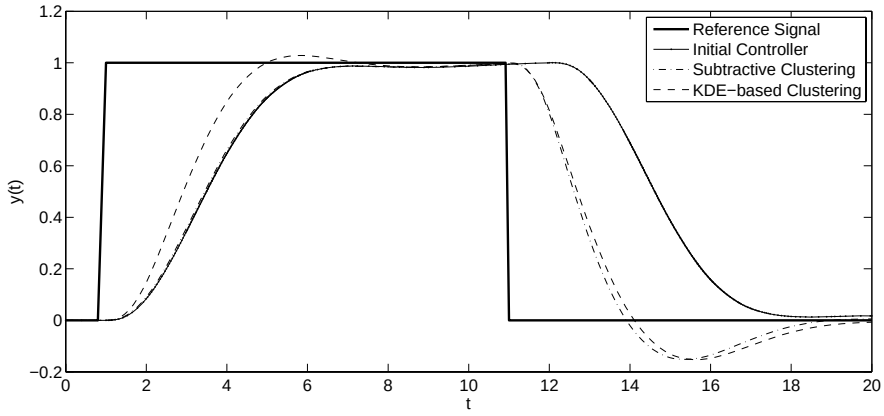

Figure 2: Comparison of fuzzy logic controllers' time responses.

Again, KDE clustering was found to be superior considering obtained clustering results. It offers the smaller RMS modelling error, quicker settling time (although with a little overshoot) and less complex structure of the inference system at the same time. It is worth mentioning as well that the reduced fuzzy logic controller performs in this case better than the original controller.

5 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. Furthermore, the high computational burden of associated calculations can be effectively reduced using parallel

processing, as shown in paper [19].

The proposed method was tested on various fuzzy modelling cases and was shown to achieve a better level of performance as an up-to-date subtractive clustering algorithm. 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] Takagi T., Sugeno M. (1985). Fuzzy Identification of Systems and Its Applications to Modeling and Control. *IEEE Transactions on Systems, Man and Cybernetics*, 15, 116–132.

[2] Olej V., Krupka J. (2005). Prediction of Gross Domestic Product Development by Takagi-Sugeno Fuzzy Inference Systems. *IEEE Proceedings of International Conference on Intelligent Systems Design and Applications*, Wroclaw, 186–191.

[3] Kadmiry B., Driankov D. (2004). A Fuzzy Gain-Scheduler for the Attitude Control of an Unmanned Helicopter. *IEEE Transactions on Fuzzy Systems*, 12, 502–515.

[4] Trabelsi A. et al. (2004). Identification of Nonlinear Multivariable Systems by Adaptive Fuzzy Takagi-Sugeno Model. *International Journal of Computational Cognition*, 2, 137-153.

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

[6] Jang J-S. R., (1993). ANFIS: Adaptive-Network-Based Fuzzy Inference System, *IEEE Transactions on Systems, Man and Cybernetics*, 23, 665–684.

[7] Yen J., Wang L. (1998). Application of Statistical Information Criteria for Optimal Fuzzy Model Construction. *IEEE Transactions on Fuzzy Systems*, 6, 362–372.

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

[9] Song Q., Kasabov N. (2001). ECM – A Novel On-line, Evolving Clustering Method and Its Applications. *Proceedings of the Fifth Biannual Conference on Artificial Neural Networks and Expert Systems*, Dunedin, 87–92.

[10] Abonyi J. et al. (2002). Modified Gath-Geva fuzzy clustering for identification of Takagi-Sugeno fuzzy models. *IEEE Transactions on Systems, Man and Cybernetics B*, 32, 612–621.

[11] Kulczycki P., Charytanowicz M. (2008). A Complete Gradient Clustering Algorithm by Kernel Estimators. *In press.*

[12] Wand M.P., Jones M.C. (1995). Kernel Smoothing. Chapman and Hall, London.

[13] Kulczycki P. (2005). Estymatory jadrowe w analizie systemowej. WNT, Warsaw.

[14] Kulczycki P. (2008). Kernel Estimators in Industrial Applications. In: Soft Computing Applications in Industry, Prasad B. (ed.), Springer-Verlag, Berlin. *In press.*

[15] Fukunaga K., Hostetler L.D. (1975). The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition. *IEEE Transactions on Information Theory*, 21, 32–40.

[16] Box G.E.P., Jenkins G.M. (1970). Time Series Analysis: Forecasting and Control. Holden-Day, San Francisco.

[17] Farag W., Tawfik A. (2000). On Fuzzy Model Identification and the Gas Furnace Data. *Proceedings of the International Conference on Intelligent Systems and Control*, Honolulu, paper: 317-083.

[18] Chopra S. et al. (2006). Analysis of Fuzzy PI and PD Type Controllers Using Subtractive Clustering. *International Journal of Computational Cognition*, 4, 30–34.

[19] Łukasik S. (2007). Parallel Computing of Kernel Density Estimates with MPI, *Lecture Notes in Computer Science*, 4489, 726–734.