Notes on Sugeno and Yasukawa's fuzzy modelling approach *

Domonkos Tikk† György Biró and László T. Kóczy
Dept. of Telecom. & Telematics, Budapest University of Technology and Economics
H-1117 Budapest, Pázmány sétány 1/d, Hungary
E-mails: {tikk,koczy}@ttt.bme.hu

Tamás D. Gedeon and Kok Wai Wong
School of Information Technology, Murdoch University
South St, Murdoch, 6150 W.A., Australia
E-mails:{tgedeon,kwong}@murdoch.edu.au

Abstract

This paper investigates the Sugeno's and Yasukawa's qualitative fuzzy modelling approach. We propose some easily implementable solution for the unclear details of the original paper. These details are crucial concerning the method's performance.

1. Introduction

This paper deals with Sugeno and Yasukawa's qualitative fuzzy modelling [5], briefly SY method. This method creates a fuzzy rule base (a set of fuzzy if-then rules) from sample input-output data, and assigns meaningful linguistic labels to the fuzzy sets in the rule base. This assignment is very important in fuzzy systems, because it makes the behaviour of system, which is modelled by the rule base, easily interpretable and transparent. In the original paper there are some details which are not quite clear, thus, require clarification or leave room for improvement. These problems concern the determination of trapeze membership function, the rule projection from sample data, the selection of important variables, and the parameter identification.

2. The SY method

The goal of the SY fuzzy modelling method is to create a transparent, viz. linguistic interpretable fuzzy rule based model from input-output sample data. The construction of the rule base is performed in two main steps: the *identification* and the build-up of the *qualitative model*. The former can be further divided into two tasks: the structure identification and parameter identification. Having an identified model at hand, linguistic labels can be assigned to the finalized fuzzy sets in the rules in the qualitative modelling phase. In this paper we focus solely on the identification step.

Table 1. Classification of the identification [5]

Structure identification I	a: input candidates b: input variables
Structure identification II	a: number of rules b: partition of the input space
Parameter identification	

In [5] the authors classified the structure identification task into two types. The type I structure identification consists of finding the input candidates of the system and finding its actual variables which affect the output. In general, the selection of the input candidates is not a systematic process, i.e., one has to take a heuristic method based on experience and/or common sense knowledge for this purpose. The type II structure identification covers the determination of the number of rules and the partition of the (usually) multidimensional input space. The identification task is summarized in Table 1. In this study we discuss the latter three structure identification methods (type Ib, type IIa,b) and the parameter identification step.

The given data set is the following; $x_1, \ldots, x_n$ are the input variables, y is the output variable. N sample data are given in the form of $(x_1^i, x_2^i, \ldots, x_n^i) \rightarrow y^i$, $i = 1, \ldots, N$.

2.1. Identification of input variables

The structure identification of type Ib concerns the selection of input variables which influence truly the out-

*This research was supported by the Australian Research Council, and the Hungarian Scientific Research Fund (OTKA) Grants No. D034614, T34212 and T34233.

†Corresponding author

put. This means that one has to choose a set of effective variables among a finite set of original variables. For this purpose one needs a criterion function to evaluate the various candidate sets of variables. This function assigns a value for a given set of variables and its task is to minimize or maximize it. In [5], they used the regularity criterion (RC) [3], which was performed between steps identification of type II and parameter identification. The outcome of the RC method depends on the identification of type II (see also Figure 1). The RC is a heuristic method which selects a set of inputs among the possible candidates.

In the first step, the sample data set is divided into two groups, A and B. The criterion function is

$$RC = \left[\sum_{i=1}^{k_A} \frac{(y_i^A - y_i^{AB})^2}{k_A} + \sum_{i=1}^{k_B} \frac{(y_i^B - y_i^{BA})^2}{k_B} \right] / 2 \tag{1}$$

where k_A and k_B are the numbers of data in groups A and B, respectively; y_i^A and y_i^B are the outputs of groups A and B, respectively; and finally y_i^{AB} (y_i^{BA}) is the model output for the group A (B) input estimated by the model identified using group B (A) data.

For evaluating (1) two models should be built from the data groups A and B at each evaluation stage. According to [5], the parameter identification should be done before calculating RC, but it is rather superfluous and makes the calculation very lengthy.

Although, this procedure is done off line, to reduce the necessary time a heuristic algorithm is used for determining the order of evaluation and the optimal set of variables which breaks down the number of evaluated nodes to $n(n+1)/2$ (see [5] for details).

Due to the lack of space, the investigation of the efficiency and reliability of the RC method is only briefly summarized here. We refer to papers [6, 7] for further details. The cited papers shows that RC heavily depends on the implementation details of the method such as, e.g., the trapeze approximation version and its parameters (see section 3.1), division of data into two groups in the starting step of the RC method (see also Fig. 1). [6] offers an alternative feature ranking method, which solves efficiently the problem.

2.2. Determination of the number of rules and the input partition

Usually in the design of a fuzzy system the rule antecedents and the partition of the input domain are determined. This (dense) rule base design methodology results in exponentiality in terms of the number of rules. To avoid this significant drawback the SY method proceeds oppositely: first the partition of the output space is determined, which is done by clustering the whole output data set by the fuzzy c-means clustering (FCMC) [1]. The optimal number of cluster are determined by means of the following criterion [2]:

$$S(C) = \sum_{k=1}^{N} \sum_{i=1}^{C} (\mu_{ik})^m \left(\|x_k - v_i\|^2 - \|v_i - \overline{x}\|^2 \right), \tag{2}$$

where N is the number of data to be clustered; C is the number of clusters, $C \geq 2$; x_k is the kth datum (usually vector), $\overline{x}$ is the average of data x_k; v_i is the the centre of the ith cluster (vector); μ_{ik} is the membership degree of the kth datum with respect to the ith cluster; m is the fuzzy exponent, $m > 1$.

As a result of the clustering, every output datum is associated with a membership degree in all the clusters B_i, $i = 1, \ldots, C$. From an output fuzzy clusters B_i we can induce a fuzzy cluster A_i in the multi-dimensional input space. This cluster can be projected onto the axes of the variables, hence defining the antecedent fuzzy sets in each input dimension. Starting from a cluster B_i, and assuming that we have two input variables x_1 and x_2, we usually obtain a rule like

$$\text{If } x_1 \text{ is } A_{i1} \text{ and } x_2 \text{ is } A_{i2} \text{ then } y \text{ is } B_i.$$

We remark that although this notation implies that the number of rules is identical with the number of output clusters, it can happen that this is not the case.

In the original paper two procedures are not specified clearly. However, we found them crucial with respect to the performance of the model and the selection of input variables with the RC method. On the one hand, in [5] the authors stated that, despite the input cluster B was convex, the corresponding input set A might not be also convex. Hence it needed to be approximated. For simplicity, they proposed to approximate the (nonconvex) input clusters with trapezoidal membership functions. On the other hand they remarked that more than one input cluster could belong to an output cluster. As a solution they suggested to "form carefully two convex fuzzy clusters" in the input space.

Although, these details seem to be not very important from the methodology aspect of the SY fuzzy modelling, they affect, especially when using the RC method for input identification, significantly the performance of the built model.

2.3. Parameter identification

Parameter identification step can be accomplished in two stages in the fuzzy model design. The authors in [5] proposed to repeat it in every input candidate evaluation step, but this is mostly superfluous and very time consuming. Performing it may be enough after the important input variables have been identified.

At this stage we have to measure the performance of the rough fuzzy model. For this purpose the following performance index (PI) is used:

$$PI = \sum_{i=1}^{N} (y^i - \hat{y}^i)^2 / N \qquad (3)$$

where $\hat{y}^i$ is the model output for the ith sample datum.

In the case of fuzzy model, the parameters are those of the membership functions. Having trapezoidal membership functions it means 4 parameter for each antecedent $p_1 \leq p_2 \leq p_3 \leq p_4$. In the parameter identification step they adjusted these four values in an iterative algorithm. 5% of the width of the actual input space was applied as adjusting value.

The flowchart on Figure 1 shows the overall design of the identification steps of the SY modelling.

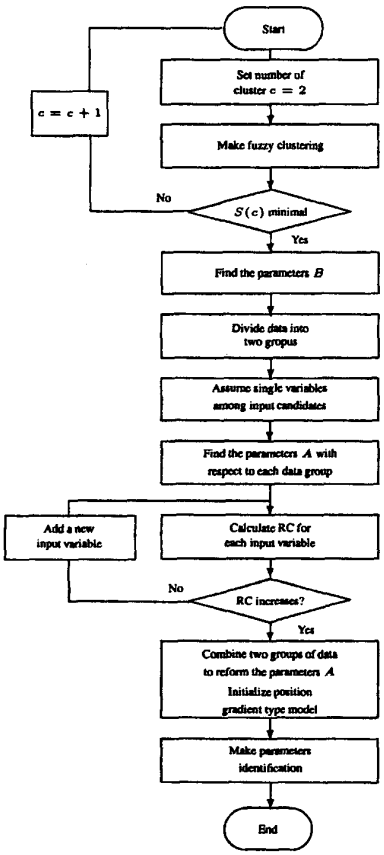

Figure 1. Flowchart of the identification steps (redrawn from [5])

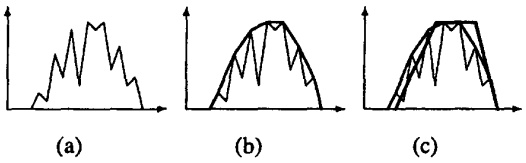

Figure 2. The two steps of trapezoidal membership function construction. (a) c-means clustered raw data; (b) the convex hull of the input cluster; (c) approximated trapezoidal fuzzy set (reproduced from [5])

3. Detailed discussion

3.1. Trapeze approximation

The trapeze approximation of the clustered raw data is proceeded in two steps. First, the convex hull of the original data set is determined, then the convex hull is approximated by a trapezoidal membership function. Figure 2 depicts the idea of the construction of trapezoidal membership function.

We propose a simple and fast trapeze approximation algorithm in three different versions. The three versions differ in the determined support length, or in other words, in the angle of the slopes of the trapeze. However, in ordinary situations the three versions generate identical or almost identical results. The difference is significant when the distribution of the data with high membership grades is large near the minimum/maximum of the cluster. In such cases the first version results in close-to-oblong shape trapezes with steep slopes, the third in trapezes with long and smooth slopes, while the second version generates an average solution of the former two. For brevity, these versions are termed steep-slope, smooth-slope and average-slope.

The common base algorithm is the following:

Trapeze approximation:

1. Determine μ_{min} and $\mu_{\max}$ the minimum/maximum of the membership degree of the data point in the given cluster, $x_{\max}$ the first point where the maximum $\mu_{\max}$ is attained, further $d_{\min}$ and $d_{\max}$ the minimum/maximum of all the data values in the given cluster domain.
2. Set the boundaries of investigated interval to

$$m = \frac{\mu_{\min}(r-1) + \mu_{\max}}{r};$$
$$M = \frac{\mu_{\max}(r-1) + \mu_{\min}}{r},$$

where $r > 1$ is a given parameter, usually $r \in [2,4]$.

3. Determine the parameters of the left slope, x_1 and x_2:

   (a) Let us initialize x_j as the last data point which has smaller membership degree than m and x_i be the next point of the convex hull:

   $$\mu(x_j) < m; \qquad \mu(x_{j+1}) = \mu(x_i) > m.$$

   (If there is no such point x in the convex hull which satisfies $\mu(x) < m$ then x_j and x_i is the first two leftmost point of the convex hull). Further the parameters of the left $x_1 = d_{\min}$ and $x_2 = d_{\min}$.

   (b) Let x_1^* and x_2^* be the location of the intersection made by the support and the core, respectively, with the line passing through the points $(x_j, \mu(x_j))$ and $(x_i, \mu(x_i))$ (see Figure 3).

   (c) If $x_1^* > x_1$ then $x_1 := x_1^*$.

   (d) If $x_2^* < x_2$ or $i = j + 1$ then $x_2 := x_2^*$.

   (e) If $x_{i+1} \leq x_{\max}$ then let $i := i + 1$ and go to step (3b), otherwise continue.

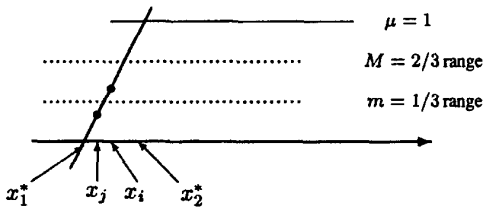

Figure 3. Determination of the location x_1^* and x_2^*. Here $r = 3$.

4. Determine the parameters of the left slope, x_3 and x_4, analogously as in the previous step.

5. Order the parameters according to $x_1 \leq x_2 \leq x_3 \leq x_4$.

In step (2), the reason of narrowing the range is twofold. From below it is important to exclude the data points with very low membership grades. Notice that the clustering algorithm assigns (almost) every data a positive membership grade in each cluster, however, usually a data has significant membership degree in at most two clusters. From above it is also reasonable to disregard the points with high membership grades, because, on one hand, they most probably belong to core of the membership function, and they do not play role in the determination of the slopes; on the other hand, the technique used for the determination of the slope is sensitive to high membership grade, as it is described in the next paragraph.

The difference between the three versions appears in step (3c). Observe that in such a situation where the two leftmost points have high membership grade, then the left slope is very fuzzy. So in this case the minimum of the support is default value: $d_{\min}$ (see Figure 4). If this phenomenon is present in both ends of the support of a trapeze, then the final membership function is positive on the whole domain. However, it is unlikely that the original cluster dominates the whole dimension.

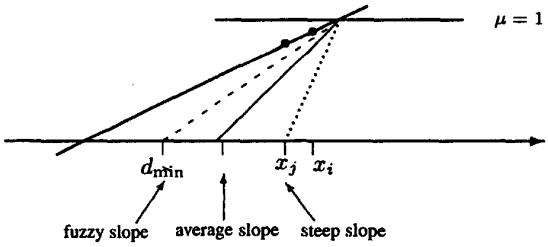

Figure 4. Too smooth left slope (thick line). The default smooth-slope version then takes $d_{\min}$ as x_1 (dash line). The steep-slope version takes x_j as x_1 (dotted line), while the average-slope version takes $(d_{\min} + x_j)/2$ (thin line).

To alleviate this drawback we propose two possible solutions: steep-slope version and average-slope version. The steep-slope version generates short support by giving x_j as the left end of the support, while the average-slope version determines the arithmetic mean of $d_{\min}$ and x_j as the leftmost point (see Figure 4).

In order to provide a simple schema we did not include exception check and handling in the above algorithm. Nevertheless, those are important parts of the implementation process.

We remark that the SY method uses parameter identification for fine tuning of the rules, when the parameters of the membership functions are shifted. Hence, starting from a wide membership function with smooth slopes one can end up with a much narrow one with steep slope, if the testing data set sustains this modification. From this point of view it is better to start with a wide membership function, which may make more flexible the parameter identification procedure. We shall return to this issue in subsection 3.3.

3.2. Determining the number of the rules

In this subsection we present an algorithm which determines the number of rules belonging to one output

cluster by defining the input clusters in the multidimensional input space. This is an important detail of the original algorithm in our view, which was not given the required attention in [5]. This question has significant effect on the total number of rules especially, when the modelled function is not strictly monotone, but the same output is assigned to several region of the input space.

Algorithm for determining the antecedents

Inputs: training data set, membership degree vector of the fuzzy c-means clustered training data set,

output: the actual rules.

1. Select a cluster i, where $i = 1, \ldots, C$, and divide the training data set into two groups: group A consists of datum with membership degree not smaller than 0.5 in cluster i, while group B includes the remaining data set:

$$A = \{x_j | \mu_i(x_j) \geq 0.5,\ j = 1, \ldots, N\}$$
$$B = \{x_j | \mu_i(x_j) < 0.5,\ j = 1, \ldots, N\}$$

2. Create a rule base R of $2 \cdot |A|$ different rules, by forming separate rules from each of the group A data end the first $|A|$ group B data. If $|A| > |B|$ then we form N rules. At this point the rules are crisp in the sense that the antecedents as well as the consequent is a crisp value.

3. Evaluate the performance of this rule base by checking this model on the group B data set. We determine the rule base performance by counting the data with incorrectly high membership grade as:

$$p_R = |B'|/|B|$$

where $B' = \{\mathbf{x}_j \in B | W_R(\mathbf{x}_j) \geq t_1, j = 1, \ldots, N\}$. Here function W_R ($W_R : \mathbb{R}^n \to [0, 1]$) determines the firing weight of the datum $\mathbf{x}_j$ by the rule base R, and the t_1 is a given threshold, usually 0.5.

4. Temporarily merge the ith and the jth rules by using the "and" operation for the corresponding antecedents and the consequents. Denote the obtained rule base by R'. Calculate $p_{R'}$. If

$$p_{R'} > p_R + t_2$$

then the performance of the new rule base is significantly worse, therefore undo the fusion of the two rules; otherwise merge them permanently: $R := R'$. Here t_2 is another threshold parameter, usually ≤ 0.5.

5. If all the rule pairs is checked then stop, otherwise check the next pair and go to step 4.

This algorithm proceeds for all the output clusters consecutively. The group A data set serves as the basis of rule forming procedure. The group B data set serves as local training data set, and it also balances the impact of group A data in rule base forming. It is advantageous to use B as training set, because its cardinality is in general less than $N/2$, if the number of clusters exceeds 2.

When we merge two rules by the "and" operation, this includes the creation of trapezoidal membership function on the basis of the added data.

When we end up with only one rule, we have to go through step 4 $2|A|-1$ times, which is proportional with $O(N)$ for each cluster. In general, step 4 is executed in worst case

$$2r|A| - r(r-1)/2 < N^2 - N(N-1)/2 = O(N^2) \quad (4)$$

where r is the cardinality of the final rule base. For details [7]. Not that this is only the number of iteration of step 4. The asymptotical number of operation in step 4 depends on the implementation of W_R. We should point out that the calculation of W_R involves the firing of the rule bases which necessitates normally exponential time for dense rule bases [4]. Due to the construction of the rule base, however, in this case the time is proportional with $n|R|$, viz. the time is polynomial.

The role of the thresholds t_1 and t_2 is to control the number of the rules. The lower the thresholds are the more rule is generated. From computational complexity reason it is not useful to choose very low threshold, because it may increase the number of rules drastically. Beside this by setting the thresholds t_1 and t_2 low, the obtained model lies very much on the training set, the rules tends to be more crisp than fuzzy, and the generalization capability of the model declines. The reasonable choice for t_1 and t_2 is $0.2 \sim 0.5$.

3.3. The parameter identification procedure

We propose a modification of the original parameter identification procedure, which works with temporally changing adjusting value f depending on the actual performance value. We set the starting adjusting value in the kth input as

$$f = \mathrm{dom}(k)/4p_{\mathrm{steps}} \cdot 2^{p_{\mathrm{start}}+m}$$

where $\mathrm{dom}(k)$ is the domain of the kth input, i.e., the difference between the smallest and the largest input in the given dimension; p_{steps} a predefined constant (default: 3); p_{start} is to set the starting precision (default: -1), and m is an iteration counter which increases if

$$PI_{\mathrm{speed}} = \frac{PI_{\mathrm{last}} - PI_{\mathrm{actual}}}{PI_{\mathrm{last}}} < 0.1 \quad (5)$$

that is, if the amelioration of the performance index is less than 10%. The starting value of m is zero. The parameter identification is organized in a double loop. In the inner loop the four parameters of the trapeze membership function of all the antecedents are sequentially adjusted with the same actual adjusting value until no improvement can be achieved or the number of inner iteration attains a certain limit. Then m is increased if (5) holds, and the whole process restarts again. The stopping criterion of the outer loop can be either the crossing of a certain time limit or when the PI_{speed} gets smaller than a certain threshold.

We used the order p_1, p_4, p_2, p_3 for adjusting the parameters, i.e., first the support's and then the core's parameters are modified. This may affect the final performance index: wider the starting support of an antecedent more space is available for finding the appropriate core length. As an experimental observation, we noticed that the support's length is inclinable to shrink, and vice-versa the core's length is rather subject of widening. Therefore, in the case of steep-slope trapeze approximation, there is not much possibility of widening the core, while this is the opposite in the case of smooth-slope version. This may be the reason of the results of our comparative investigation on the trapeze approximation versions versus performance index. The results are summarized in Tables 2 and 3, where PI_1 and PI_2 denotes the performance index before and after parameter identification.

Table 2. Performance indices with the use of different trapeze approximation versions on synthetical sample data set ($y = (1 + x_1^{-2} + x_2^{-1.5})^2$)

	Smooth	Average	Steep
PI_1	0.5580	0.4110	0.3870
PI_2	0.0322	0.0355	0.0497
Improvement	17.33	11.57	7.79

Table 3. Performance indices with the use of different trapeze approximation versions on chemical plant sample data set

	Smooth	Average	Steep
PI_1	$4.64 \cdot 10^5$	$7.58 \cdot 10^4$	$1.03 \cdot 10^5$
PI_2	$2.60 \cdot 10^4$	$1.06 \cdot 10^4$	$7.15 \cdot 10^4$
Improvement	17.85	7.15	1.44

We can state that smoother the starting membership function the bigger the achieved improvement can be. However, it does not mean automatically that the smooth-slope version provides the best result. For example, in the chemical plant case the average-slope version produces the best result both before and after parameter identification. Based on these two examples the best choice seems to be the average-slope version, because it gives good results according to PI, before and after parameter identification as well. (We should remark that for Table 2 results we forced the $\{1, 2\}$ true input set due to the fact that the RC determined different inputs for the various trapeze versions.)

4. Conclusion

In this paper we cleared some vague, unexplicit details of the Sugeno's and Yasukawa's qualitative fuzzy modelling method. We proposed algorithms for trapeze approximation, for the determination of the number of rules (belonging to one output cluster) and parameter identification. We also pointed out that the reliability and stability of the RC method, which is applied to determine the true inputs among input candidate variables, is poor, and leaves room for improvement. Therefore we proposed an alternative technique for ranking the input variables in [6].

References

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

[2] Y. Fukuyama and M. Sugeno. A new method of choosing the number of clusters for fuzzy c-means method. In *Proc. of the 5th Fuzzy System Symposium*, pages 247–250, 1989. (in Japanese).

[3] J. Ihara. Group method of data handling towards a modeling of complex system IV. *Systems and Control*, 24:158–168, 1980. (in Japanese).

[4] L. T. Kóczy. Computational complexity of various fuzzy inference algorithms. *Annales Univ. Sci. Budapest, Sect. Comp.*, 12:151–158, 1991.

[5] M. Sugeno and T. Yasukawa. A fuzzy logic based approach to qualitative modelling. *IEEE Trans. on Fuzzy Systems*, 1(1):7–31, 1993.

[6] D. Tikk and T. D. Gedeon. Feature ranking based on interclass separability for fuzzy control application. In *Proc. of the Proceedings of Int. Conf. on Artificial Intelligence in Science and Technology (AISAT'2000)*, pages 29–32, Hobart, Tasmania, Australia, 2000.

[7] D. Tikk, L. T. Kóczy, G. Biró, and T. D. Gedeon. Improvements and critique on Sugeno and Yasukawa's qualitative modelling. Research Working Paper RWP-01-2001, School of Information Technology, Murdoch University, Perth, W.A., Australia, 2001. 22 p.