

Identification of Transparent, Compact, Accurate and Reliable Linguistic Fuzzy Models

Andri Riid^a, Ennu Rüstern^b

^a*Laboratory of Proactive Technologies, Tallinn University of Technology, Ehitajate tee 5, 19086, Tallinn, Estonia, e-mail: andri@dcc.ttu.ee*

^b*Department of Computer Control, Tallinn University of Technology, Ehitajate tee 5, 19086, Tallinn, Estonia, e-mail: ennu.rystern@dcc.ttu.ee*

Abstract

Transparency, accuracy, compactness and reliability all appear to be vital (even though somewhat contradictory) requirements when it comes down to linguistic fuzzy modeling. This paper presents a methodology for simultaneous optimization of these criteria by chaining previously published various algorithms - a heuristic fully automated identification algorithm that is able to extract sufficiently accurate, yet reliable and transparent models from data and two algorithms for subsequent simplification of the model that are able to reduce the number of output parameters as well as the the number of fuzzy rules with only a marginal negative effect to the accuracy of the model.

Keywords: Fuzzy modeling, interpretability of fuzzy systems, complexity reduction

1. Introduction

The research on fuzzy systems (see e.g. [9, 10, 22, 7]) of last years has adequately pointed out the uniqueness and value of interpretability and has also provided means and tools for facilitation and exploitation of this property. It seems that a tentative consensus has been reached in what comprises interpretability. Aside from low-level interpretability requirements (normality, coverage, convexity and distinguishability of fuzzy partitions) that have progressively become a norm in fuzzy community, higher-level interpretability has become somewhat interchangeable with complexity (often termed as readability in interpretability context). For example, a recent work [3] considers a small number of fuzzy rules and compact (incomplete) rules for large

systems instrumental to interpretability and to reflect that, the proposed hierarchical fuzzy system for assessing interpretability in this paper combines different complexity measures to produce the interpretability index.

Aside from being a measure of evaluation, interpretability index can serve as the optimization criterion for evolutionary algorithms to improve interpretability of a fuzzy system and indeed, evolutionary algorithms have become increasingly popular in fuzzy optimization [10, 7, 13, 17, 2]. However, these algorithms work with a family of potential solutions, are therefore computationally expensive and require many (sometimes thousands) iterations to converge. This is often unacceptable for practical applications and computationally more affordable alternatives must be sought.

Interestingly enough, most latest interpretability-related developments ([3, 17, 2, 23]) have taken place in the context of classification where the task of a fuzzy rule-based classifier is just to assign a class label (the number of which is limited) to the sample presented to it. In modeling and control, however, the output is generally continuous imposing perhaps higher accuracy requirements and rule interpolation obtains a central place. In consequence, complexity/readability issue that is prominent in most interpretability studies becomes less important concern (note that because of the curse of dimensionality fuzzy modeling is rarely performed for large-scale systems), however, this is more than compensated by increased interpolation-driven interpretability (and other) concerns.

The latter is the main reason why in fuzzy modeling and control we prefer to handle interpretability in a wider context where interpretability is perceived as a measure of fuzzy system consistency [33] - an umbrella term that has been coined to embrace all aspects of fuzzy system applicability in modeling (not to be confused with rule consistency utilized e.g. in [3]) - and more specifically, a measure of internal consistency (that has its own aspects of transparency, linguistic integrity and complexity).

What really unites all aspects of internal consistency is that they can be generally validated without external information (e.g. validation data). Aside from purely academic research we, however, usually want to exploit interpretability for the problem at hand and therefore an internally totally consistent fuzzy system is generally not really useful if it is numerically grossly inaccurate or its rules cannot be relied on because they express information that cannot be confirmed otherwise (by available numerical data or expert opinion). These concerns - accuracy and reliability - are the most important aspects of external consistency and, incidentally, what we typically aim for

is a certain balance between internal and external consistency of the system (this is perhaps better known as interpretability-accuracy tradeoff).

In this paper our goal is to provide a new methodology that is able to handle adequately all aspects of system consistency (both internal and external) in fuzzy modeling at a moderate computational cost. For this we employ different algorithms.

The first step of the procedure is the identification of a transparent fuzzy model using the training data and a fully automatic algorithm (developed to perfection in [35] to cope with noisy environment) that has built-in mechanisms for transparency protection and reliability preservation.

The class of systems under consideration here are the fuzzy singleton (or 0-th order Takagi-Sugeno) systems. What makes these systems special is that they have all the attractive properties of linguistic (Mamdani) systems, whereas numerically they are very easy to manipulate (their inference function is analytical and inexpensive) and interpolation in such systems is very intuitive.

The assessment of complexity/readability of rules is carried out in subsequent manipulation of the identified model by two further algorithms and is twofold. First, the issue of abundance of output singletons, characteristic to 0-th order TS systems and the direct result of the application of the modeling algorithm in previous step, is addressed using a recently developed reduction algorithm [36]. This heavily reduces the number of output parameters and makes evident otherwise hidden redundancy of fuzzy rules that can be removed by yet another recent method [34, 37].

Numerous examples (including the applications of gas furnace and acidogenic state modeling) positively confirm that what we have here is an efficient tool for minimizing the gap between accuracy (from one side) and the properties of transparency, reliability and complexity from another side.

2. Preliminaries

Consider a multi-input single-output fuzzy system, consisting of R rules:

$$\begin{aligned} &\text{IF } x_1 \text{ is } A_{1r} \text{ AND } x_2 \text{ is } A_{2r} \text{ AND } \dots \\ &\quad \dots \text{ AND } x_N \text{ is } A_{Nr} \text{ THEN } y \text{ is } b_r \\ &\quad \text{OR } \dots, \end{aligned} \tag{1}$$

where A_{ir} denote the linguistic labels of the i -th ($i = 1, \dots, N$) input variable (into which these variables have been partitioned) associated with the r -th

($r = 1, \dots, R$) rule, and b_r is the scalar (fuzzy singleton), associated with the r -th rule.

Each A_{ir} has its representation in the numerical domain - the membership function μ_{ir} (MF). In a normal fuzzy system the number of MFs per i -th variable (S_i) is relatively small - in any way, this number is rarely equal to R as the notation style in (1) implies - moreover, it is often desired that all possible unique combinations of input MFs are represented ($R = \prod_{i=1}^N S_i$). MFs of the system are thus shared between the rules and a separate $R \times N$ dimensional matrix that accommodates the identifiers $m_{ri} \in \{1, 2, \dots, S_i\}$ maps the existing MFs μ_i^s to the rule slots. The number of independent output singletons (T) in fuzzy singleton (0-th order Takagi-Sugeno systems), on the other hand, is generally equal to R (and thus matches the notation style in (1)).

In current approach MFs μ_i^s are defined by

$$\mu_i^s(x_i) = \begin{cases} \frac{x_i - a_i^{s-1}}{a_i^s - a_i^{s-1}}, & a_i^{s-1} < x_i < a_i^s \\ \frac{a_i^{s+1} - x_i}{a_i^{s+1} - a_i^s}, & a_i^s < x_i < a_i^{s+1} \\ 0, & a_i^{s+1} \leq x_i \leq a_i^{s+1} \end{cases}, \quad (2)$$

by what

$$\sum_{s=1}^{S_i} \mu_i^s(x_i(k)) = 1. \quad (3)$$

The latter has become known as Ruspini [38], strong [13] or *standard* partition and is often exploited for its simplicity and for built-in low-level interpretability requirements (coverage, normality, convexity, distinguishability).

The inference function that corresponds to (1) and computes the matching $y(k)$ to the input vector $[x_1(k), \dots, x_i(k), \dots, x_N(k)]$ is given by

$$y(k) = \frac{\sum_{r=1}^R \tau_r(k) b_r}{\sum_{r=1}^R \tau_r(k)}, \quad (4)$$

where $\tau_r(k)$ is the activation degree of the r -th rule.

$$\tau_r(k) = \prod_{i=1}^N \mu_{ir}(x_i(k)), \quad (5)$$

Using the notations

$$\Gamma = \begin{bmatrix} \tau_1(1) & \tau_2(1) & \dots & \tau_R(1) \\ \tau_1(2) & \tau_2(2) & \dots & \tau_R(2) \\ \dots & \dots & \dots & \dots \\ \tau_1(K) & \tau_2(K) & \dots & \tau_R(K) \end{bmatrix}, \quad (6)$$

$$\mathbf{b} = [b_1, b_2, \dots, b_R]^T, \quad (7)$$

and

$$\mathbf{y} = [y(1), y(2), \dots, y(K)]^T, \quad (8)$$

we can see that (4) can be expressed by

$$\mathbf{y} = \mathit{pinv}(\mathit{diag}(\Gamma \cdot \mathbf{e})) \cdot \Gamma \cdot \mathbf{b}, \quad (9)$$

where $\mathit{diag}()$ denotes the operation which transforms a column vector (its argument) into a diagonal matrix, \mathbf{e} is a $(R \times 1)$ vector of ones and $\mathit{pinv}()$ is the Moore-Penrose pseudoinverse [30] that is applied for matrix inversion throughout the paper. Note, however, that in (9), the inverted matrix is a diagonal one so its inversion can as well be obtained by replacing each element in the diagonal with its reciprocal and $\mathit{pinv}()$ is there just for the convenience of notation.

If Γ and \mathbf{y} are known, we can use the pseudoinverse to compute a least squares solution to (9) that lacks an exact solution in terms of \mathbf{b} .

$$\mathbf{b} = \mathit{pinv}(\Gamma) \cdot \mathit{diag}(\Gamma \cdot \mathbf{e}) \cdot \mathbf{y} \quad (10)$$

The latter is *de facto* standard identification method for the output parameters (singletons b_r) of 0-th order TS systems and perhaps the reason why the number of unique singletons in these systems is generally equal to R .

3. Internal and external consistency

In this section we describe in more detail the different aspects of system consistency, both internal (transparency, linguistic integrity and complexity) and external (reliability and accuracy).

Transparency is defined as a measure of conformity between the linguistic and inference layers of a fuzzy system - (1) and (4), respectively. Transparency of a linguistic fuzzy system is validated rule-by-rule whereas r -th rule is transparent if by

$$\exists k, \tau_r(k) = 1 \quad (11)$$

$$a_{ir} = x_i(k), (i = 1, \dots, N), y(k) = b_r, \quad (12)$$

where a_{ir} is the center of the input MF of the i -th input variable associated with the r -th rule. One can see such favorable situation in Figure 1, where the rule node (square at right) - the data point in what $\tau_r(k) = 1$, is also backed up numerically as the input-output relationship the rules of the system generate (solid line) goes through this point. On the other hand it is possible to construct numerically identical fuzzy systems that have rules that do not satisfy (12) and consequently have the rule node at a different location (the square at left in Figure 1) because transparency is not a default property of fuzzy systems. However, transparency of fuzzy systems can be easily maintained according to [31] that states that a fuzzy system such as (1) is transparent as long as we preserve (2) in all manipulations with the model parameters. For these systems, transparency is of binary character that requires no further evaluation. In current approach transparency preservation is a built-in feature in all presented algorithms.

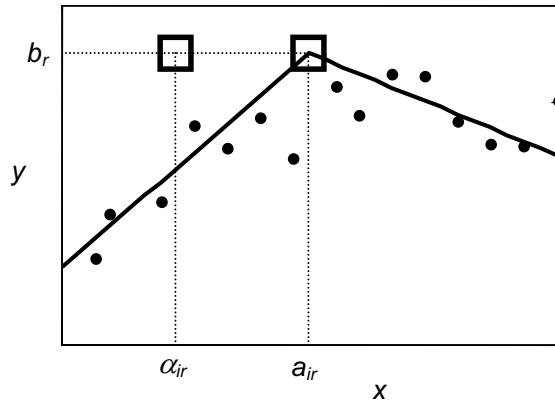


Figure 1: Transparent rule (its node is depicted by a square at right) matches the data it infers (solid line) whereas numerically identical non-transparent system may have a rule (square at left) interpretation of which would give us an untrue assumption about system behavior

If system transparency is taken care for, the question of *linguistic integrity* boils down to the proper labeling of MFs. Requirements such as “the ordering of linguistic labels sets should reflect the order of membership values of corresponding fuzzy sets” or “MFs carrying semantically negative labels should not appear in the positive side of the domain” are typical linguistic

integrity considerations that can be generally solved by revision and relabeling of fuzzy sets, which usually requires no other skills than common sense. This is a post-modeling procedure that is not specifically targeted here.

Complexity of the system is a more universal concept. Considering interpretability, complexity plays assisting role as the systems with less rules and rules with lesser components can be interpreted with less effort (this has been confirmed by a recent web poll [4]). For fuzzy systems, it is understood that the number of variables (N), the number of MFs (S_i) and the number of rules (R) should be moderate (all these serve as measures of complexity). Obviously, computational cost is directly influenced by complexity. In the end, the problem is about how to make the system as simple as possible without jeopardizing its functionality. Two simplification algorithms have been employed to obtain fine balance between accuracy and complexity in this paper.

The primary measure of *accuracy* is the approximation error (the difference between actual output of the system \mathbf{y} and the desired output $\hat{\mathbf{y}}$), usually computed as root-mean-squared error (RMSE)

$$\epsilon = \|\mathbf{y} - \hat{\mathbf{y}}\|/K, \quad (13)$$

but, particularly when model is identified from scarce (and possibly noisy data), additional difficulties arise as the modeling algorithm has made generalizations on the basis of existing samples. These situations where there is not enough material (data) or immaterial (knowledge) evidence to cover the input space universally arise quite frequently, not only because it would be too time consuming to collect exhaustive evidence in large scale applications but also because of potential inconsistencies that certain antecedent combinations may present (an antecedent “IF sun is bright AND rain is heavy” could be one such example).

Reliability of the model depends on the distribution of the training data as well on how the identification algorithm treats the parameters of the model. Neural network inspired fuzzy modeling methods [18] generally rely on global learning techniques driven by numerical approximation error and tend to obtain the missing rules by drawing conclusions through the extrapolation of existing data samples often resulting in fuzzy rules that are unrealistic or simply untrue for the given application, interpretation of which would lead to invalid conclusions (Figure 2). It is known that (10) in its pure form has also such properties and is therefore only very carefully used in current approach.

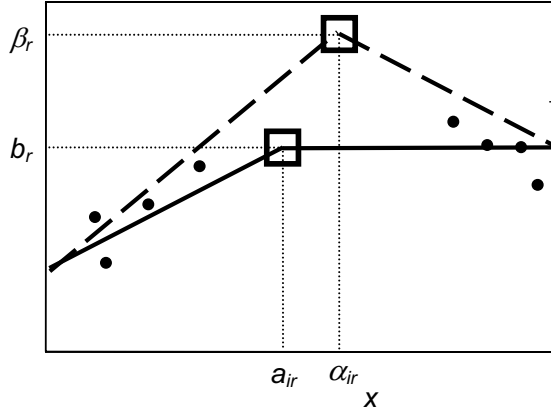


Figure 2: Some algorithms that identify rules from scarce data (dots) can occasionally give us better approximation of training data (rule on the right vs. the rule on the left) but they also may generate rules that are implausible for the given application

Our treatment of reliability is twofold. First, the rules with little evidence are filtered out according to $\max_k(\tau(r)) < \tau_{min}$, where τ_{min} is the threshold value. Secondly, instead of (10) output singletons are identified by a simple method of Nozaki [27] that also provides the measure for system reliability (singletons of 0-th order TS system are compared to the ones computed by Nozaki's method that are considered ideal from reliability viewpoint).

$$\rho = \|\mathbf{b} - \text{pinv}(\text{diag}(\Gamma^T \cdot \mathbf{e})) \cdot \Gamma^T \cdot \mathbf{y}\|/R, \quad (14)$$

The latter measure is used consistently alongside (13) in further sections of this paper to evaluate identified models.

4. The identification algorithm

Arguably, good learning schemes should be able to place optimal lone rules so that they cover the extremes or bumps of the approximand and then fill in between with extra rule patches if the rule budget allows [20]. The method by Nakoula et al. [25] that serves as the basic building block of the proposed algorithm, is principally an implementation of this strategy and places the rules iteratively at the locations in input space responsible for maximum local error. The algorithm consists of the following steps:

- Initialization. For each input variable x_i , two MFs are placed at the extremes of its domain (x_i^{min}, x_i^{max}) . This is followed by rule generation

phase where 2^N rules containing all possible unique input MF combinations are created (minus these that do not satisfy $\max_k(\tau_r(k)) > \tau_{min}$) and b_r in these rules (1) are given the values of output readings $y(k)$ that correspond to the sample $\mathbf{z}_k = [\mathbf{x}_k, y(k)]$ that provides the maximum value of (5) for the given rule.

- At l -th iteration the absolute value of approximation error $\epsilon(l)$ is computed over the training data set and new rule node, the sample $\mathbf{z}_k(l)$ responsible for $\max(\epsilon(l))$ is identified. The input coordinates of the rule node $[x_1(k), \dots, x_i(k), \dots, x_N(k)]$ are used as the centers (a_i^s in (2)) of MFs added in this step (one per each input variable) and the MFs in the immediate neighborhood of added MFs are updated to preserve (2). The existing rulebase is then revised - all consistent rules that can be formulated on the basis of updated partition are added to the rule base (unless $\max_k(\tau_r(k)) < \tau_{min}$ for the given (r -th) rule). This is followed by another iteration until we feel like calling it a day for one reason or another (approximation error is low enough, there are enough rules and MFs already or there is no further improvement).

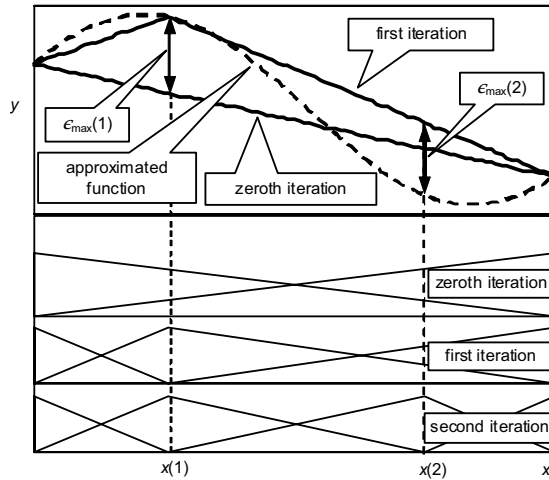


Figure 3: First few iterations of Nakoula's algorithm.

For illustration, an example of approximation of a single-input-single-output function is depicted in Figure 3. As the final result in Figure 4

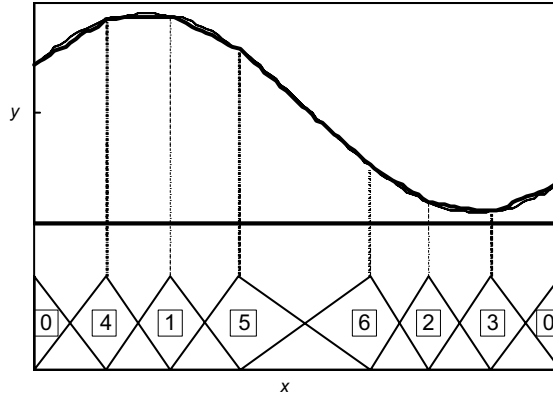


Figure 4: Final approximation result with 8 fuzzy rules. Numbers on MFs indicate the order in what they were generated.

demonstrates, what we have here is a very simple yet clever and also computationally cheap method that can produce a reasonable approximation just in a few iterations.

However, the fundamental shortcoming of the method is that it does not cope well with noisy data, as an example in Figure 5 bluntly demonstrates. The main issue with the algorithm is that it tends to learn noise rather than the signal (i.e. it favors the samples with highest noise ratio as rule nodes). The outliers (erroneous samples 2, 6 and 8) are the worst offenders as those are concentrated on in the first place. This can be evidenced from the resulting input partition, i.e. the high concentration of input MFs at certain locations. Even if there are no obvious outliers in the data set, the samples with higher noise ratio are still among the first to be picked (e.g. samples no 1, 4, 12) and the resulting approximation is therefore grossly non-smooth. Besides, more iterations are required to obtain an approximation of any quality than in the noise-free case.

One part of the solution comes from replacing consequent parameter identification routine in Nakoula's original approach with the method of Nozaki et al. [27]

$$\mathbf{b} = \text{pinv}(\text{diag}(\Gamma^T \cdot \mathbf{e})) \cdot \Gamma^T \cdot \mathbf{y}. \quad (15)$$

Γ in (15) is exponentiated elementwise with α so that each element in Γ becomes $\tau_r^\alpha(k)$ and the value of α influences model accuracy in terms of root-mean-squared error (RMSE) - it is reported in [27] that $\alpha = 10$ provides best results in ideal environment and that it should be smaller if data is bad. Note also

that if $\alpha = 1$, (15) is the local least squares method [1] and the larger it is, the more will (15) resemble original Nakoula's method in terms of performance. The basic important characteristic of Nozaki's method is that consequent parameters for a given rule are computed as the weighted average of relevant (relevancy is expressed by rule activation degree $\tau_r(k)$) output samples that gives the algorithm interpolating rather than extrapolating character.

The idea behind the second part of the solution is quite simple. First, we define a resolution vector $\mathbf{res} = [res_1, res_2, \dots, res_N]$ that specifies the resolution for each input variable, measured as a percentage of its domain.

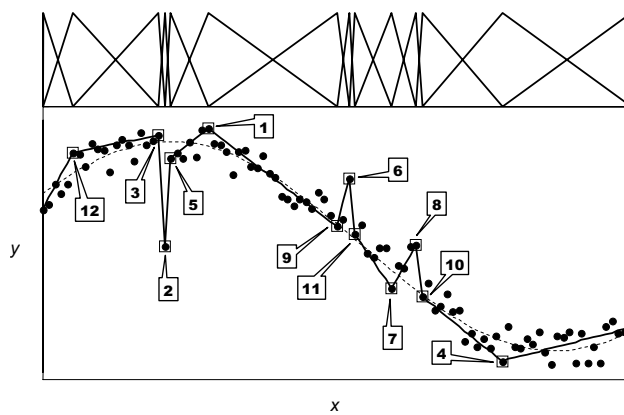


Figure 5: Nakoula's algorithm fails when data is noisy. Numbers indicate the order in which the rule nodes (highlighted samples) are picked

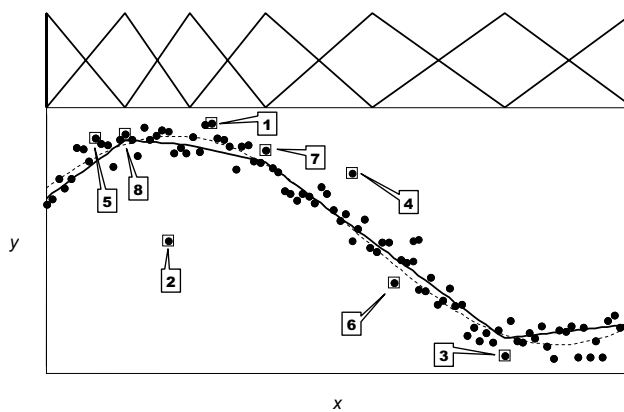


Figure 6: Approximation of noisy data with the proposed algorithm

Each iteration progresses normally as with Nakoula’s approach with consequent parameters computed with (15). At the end of each iteration, however, the data samples that fall within the hypercube around the last rule node with dimensions $res_1 \cdot (x_1^{max} - x_1^{min}) \times res_2 \cdot (x_2^{max} - x_2^{min}) \times \dots \times res_N \cdot (x_N^{max} - x_N^{min})$ are removed from the training data set. The specified resolution measures apply to input axes too, i.e. if the distance between the i -th coordinate of the rule node $x_i(k)$ and the center a_i^s of an already existing MF is smaller than $res_i \cdot (x_i^{max} - x_i^{min})$, the center of the already existing MF is updated so that $a_i^s = (a_i^s + x_i(k))/2$ is the arithmetic mean of those two (a new MF is not added to the partition of i -th input variable).

If we compare Figure 5 with Figure 6 we can see that the proposed modifications to the original method has several advantages: identified model has less MFs and consequently a lower number of rules as well as a lower RMSE.

The parameters **res** (typically uniform resolution measure is applied to all axes if $N > 1$) and τ_{min} act as modeling parameters that are specified manually by the user. Indirectly, **res** and τ_{min} allow us to determine the number of MFs per variable (S_i) and overall number of rules R , respectively, and heavily affect the overall course of learning. We will later see that especially **res** is crucial to convergence.

Due to the characteristics of the output singleton computation procedure (15) the model at this point has R generally unique singletons. In the next section we introduce a procedure that allows us to reduce this number considerably without sacrificing accuracy and reliability of the model.

5. Reduction of output singletons

If the number of unique output MFs is smaller than the number of rules ($T < R$), it follows that just as input MFs, these MFs must be shared among rules. Let \mathbf{b}' be a $T \times 1$ vector of output singletons. The information about which output MFs belongs to which rule can be expressed by a $R \times T$ mapping matrix M (that can be considered as a crisp version of the fuzzy relational matrix introduced by Pedrycz [28]), in which each row is an unity vector (in normal 0-th order TS systems M is an $R \times R$ identity matrix that is appropriately neglected from (9)).

For example, given a 0-th order TS system with Γ that is a $K \times 6$ matrix and

$$\mathbf{b}' = [b_1, b_2, b_3, b_4]^T \tag{16}$$

then

$$M = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (17)$$

maps b_1 to the second rule, b_2 to the third and sixth rule, b_3 to the first and fifth rule and b_4 to the fourth rule.

It follows then that we can replace \mathbf{b} with $M \cdot \mathbf{b}'$ in (9) so that the latter becomes

$$\mathbf{y} = \text{pinv}(\text{diag}(\Gamma \cdot \mathbf{e})) \cdot \Gamma \cdot M \cdot \mathbf{b}' \quad (18)$$

Assume we have a fully defined 0-th order Takagi-Sugeno system (4). The algorithm that reduces the vocabulary of the output variable consists of three steps. In first step the initial definition of \mathbf{b}' is found by clustering the elements in \mathbf{b} using e.g. subtractive clustering [12] that determines the number of clusters automatically based on a pre-specified cluster radius (k-means clustering [16] is used for the same problem in [14] and c-means clustering [6] is suggested in [5]). These cluster centers serve only as the prototypes of final parameters. In the next step, the mapping matrix is found (initially a $R \times T$ zero matrix). For the r -th rule the j -th cluster center that is closest to given b_r is found and the element in j -th column and r -th row in M is assigned the value of one. In third step output singletons are identified by

$$\mathbf{b}' = \text{pinv}(\Gamma \cdot M) \cdot \text{diag}(\Gamma \cdot \mathbf{e}) \cdot \mathbf{y}, \quad (19)$$

which completes the algorithm.

To estimate the information loss, we propose a measure (root mean squared error of sorts)

$$J_q = \|\mathbf{b} - M \cdot \mathbf{b}'\|/R, \quad (20)$$

To sum it up, let us consider a nonlinear function

$$y = e^{-x_1} + e^{-x_2}, \quad x_1, x_2 \in [0, 5] \quad (21)$$

and the 16-rule 0-th order TS model of this function that has been replicated from [11]. We also regenerate the 1000-sample training data set (for being randomly distributed in input space it is not exactly the same set of data as in [11]). We then reduce the number of output singletons of the original

model from 16 to 8 (7 ± 2 is often considered an optimal number for S_i for the reasons rooted in human psychology [24]), according to the procedure described above. The results, the mean square errors ($MSE = RMSE^2$), reliability measures (ρ) of both models - the original one and the reduced one - and the measure of information loss (J_q) for the latter are given in Table 1. Note that if $T < R$, \mathbf{b} in (14) must be replaced by $M \cdot \mathbf{b}'$ to evaluate properly.

Table 1: Reduction of the model of the nonlinear function (21)

	Chen [11]	reduced model
$RMSE$	0.0929	0.0931
ρ	0.0411	0.0427
J_q	-	0.0276

We can conclude that reduction of the number of unique singletons in the original model of (21) is a win-win situation as it has only a very minor negative effect to the accuracy of the model or its reliability.

6. Redundancy detection and removal

In systems where the number of rules is relatively high and the number of unique MFs is small, potential for inherent redundancy is quite high and can be removed with the algorithm described below. Note that for the class of linguistic systems we are considering in this paper, this reduction scheme is error-free, i.e. without any performance loss.

The algorithm, which, in principle (although not in implementation) is rather similar to the one that can be found from [3], is based on three lemmas (the proofs and implementation details of which can be found from [37]).

Lemma 1 (rule compression scenario A): Consider a subset of fuzzy rules consisting of S_i rules that share the same output MF B_ξ so that

$$\begin{aligned}
 &\text{IF } x_1 \text{ is } A_1^{s_1} \text{ AND } \dots \text{ AND } x_i \text{ is } A_i^s \dots \\
 &\dots \text{ AND } x_N \text{ is } A_N^{s_N} \text{ THEN } y \text{ is } B_\xi \\
 &\quad s = 1, \dots, S_i
 \end{aligned} \tag{22}$$

It can be shown that (22) is equivalent to a rule

$$\begin{aligned} \text{IF } x_1 \text{ is } A_1^{s_1} \text{ AND } \dots \text{ AND } x_{i-1} \text{ is } A_{i-1}^{s_{i-1}} \text{ AND } x_{i+1} \text{ is } A_{i+1}^{s_{i+1}} \dots \\ \dots \text{ AND } x_N \text{ is } A_N^{s_N} \text{ THEN } y \text{ is } B_\xi \end{aligned} \quad (23)$$

Lemma 2 (rule compression scenario B): If a subset of fuzzy rules consisting of $S_i - 1$ rules share the same output MF

$$\begin{aligned} \text{IF } x_1 \text{ is } A_1^{s_1} \text{ AND } \dots \text{ AND } x_i \text{ is } A_i^s \dots \\ \dots \text{ AND } x_N \text{ is } A_N^{s_N} \text{ THEN } y \text{ is } B_\xi \\ s = 1, \dots, S_i, s \neq t \end{aligned} \quad (24)$$

then this group of rules can be replaced by a following single rule.

$$\begin{aligned} \text{IF } x_1 \text{ is } A_1^{s_1} \text{ AND } \dots \text{ AND } x_i \text{ is NOT } A_i^t \dots \\ \dots \text{ AND } x_N \text{ is } A_N^{s_N} \text{ THEN } y \text{ is } B_\xi \end{aligned} \quad (25)$$

Lemma 3: Redundant MFs. Consider a pair of fuzzy rules that share the same output MF B_ξ

$$\begin{aligned} \text{IF } x_1 \text{ is } A_1^{s_1} \text{ AND } \dots \text{ AND } x_i \text{ is } A_i^s \dots \text{ AND } x_N \text{ is } A_N^{s_N} \\ \text{THEN } y \text{ is } B_\xi \\ \text{IF } x_1 \text{ is } A_1^{s_1} \text{ AND } \dots \text{ AND } x_i \text{ is } A_i^{s+1} \dots \text{ AND } x_N \text{ is } A_N^{s_N} \\ \text{THEN } y \text{ is } B_\xi \end{aligned} \quad (26)$$

and assume that there are $\prod_{j=1, j \neq i}^N S_j$ similar pairs (having A_i^s in the first and A_i^{s+1} in the second rule) that share the output MFs B_ξ within the pair ($\xi \in [1, \dots, T]$). In this case the MFs μ_i^s and μ_i^{s+1} can be merged into $\mu_i^{s \cup s+1} = \mu_i^s + \mu_i^{s+1}$ by the means of summation, consequently each rule pair (26) will reduce to

$$\begin{aligned} \text{IF } x_1 \text{ is } A_1^{s_1} \text{ AND } \dots \text{ AND } x_i \text{ is } A_i^{s \cup s+1} \dots \\ \dots \text{ AND } x_N \text{ is } A_N^{s_N} \text{ THEN } y \text{ is } B_\xi \end{aligned} \quad (27)$$

Note that the merge of two triangles of (2) by sum would result in a trapezoid MF and the updated partition would still satisfy (3).

All these redundancy removal scenarios are exploited in the simplification of the fuzzy trajectory management unit (TMU) of truck backer-upper control system from [32] that originally uses 28 rules that specify the optimal truck angle Φ_r in respect to its coordinates x and y (e.g. "IF x is mf3 AND y is mf3 THEN Φ is 90°"). Application of the algorithm reveals that

the original controller is heavily redundant as the number of its rules can be reduced to 11 without any loss in control quality that means almost 60% reduction in size (see Figure 7). Incidentally, the biggest contribution to size reduction comes from detection and merging redundant MFs (13 rules), rule compression scenario A removes 2 and scenario B further 2 rules and the final rule base looks like:

$$\begin{aligned}
& \text{IF } x \text{ is mf1 AND } y \text{ is mf1 THEN } \Phi = 225^\circ \\
& \text{IF } x \text{ is mf1 AND } y \text{ is mf2 THEN } \Phi = 180^\circ \\
& \text{IF } x \text{ is mf1 AND } y \text{ is mf3 THEN } \Phi = 135^\circ \\
& \text{IF } x \text{ is mf2 AND } y \text{ is mf1 THEN } \Phi = 180^\circ \\
& \text{IF } x \text{ is mf2 AND } y \text{ is NOT mf1 THEN } \Phi = 135^\circ \\
& \text{IF } x \text{ is mf3 THEN } \Phi = 90^\circ \\
& \text{IF } x \text{ is mf4 AND } y \text{ is mf1 THEN } \Phi = 0^\circ \\
& \text{IF } x \text{ is mf4 AND } y \text{ is NOT mf1 THEN } \Phi = 45^\circ \\
& \text{IF } x \text{ is mf5 AND } y \text{ is mf1 THEN } \Phi = 45^\circ \\
& \text{IF } x \text{ is mf5 AND } y \text{ is mf2 THEN } \Phi = 0^\circ \\
& \text{IF } x \text{ is mf5 AND } y \text{ is mf3 THEN } \Phi = -45^\circ
\end{aligned} \tag{28}$$

7. Applications

7.1. Acidogenic state modeling

We consider a fault diagnosis problem in a wastewater anaerobic digestion process (where organic matter is decomposed into biogas through biological processes taking place in the absence of oxygen). We focus on the acidogenic state of the process that is important to detect properly. The data set that originates from the LBE (Le Laboratoire de Biotechnologie de l'Environnement) situated in Narbonne, France, consists of 559 samples coming from a pilot-scale up-flow anaerobic fixed bed reactor with a volume of $0.984m^3$. Four input variables - pH (pH in the reactor), vfa (volatile fatty acid concentration), qIn (input flow rate) and CH_4 (CH_4 concentration in biogas) - are considered. The output is a number from 0 to 1, measuring to what extent the actual state can be considered acidogenic.

The original model in [14] consisting of 53 rules and producing RMSE of 0.046 is obtained by the application of interpretability preserving modification of orthogonal least squares (OLS). The output singletons of the model are shown in Figure 8. It is noteworthy that quite a few of these are located outside $[0,1]$, which can be contributed to the properties of OLS algorithm

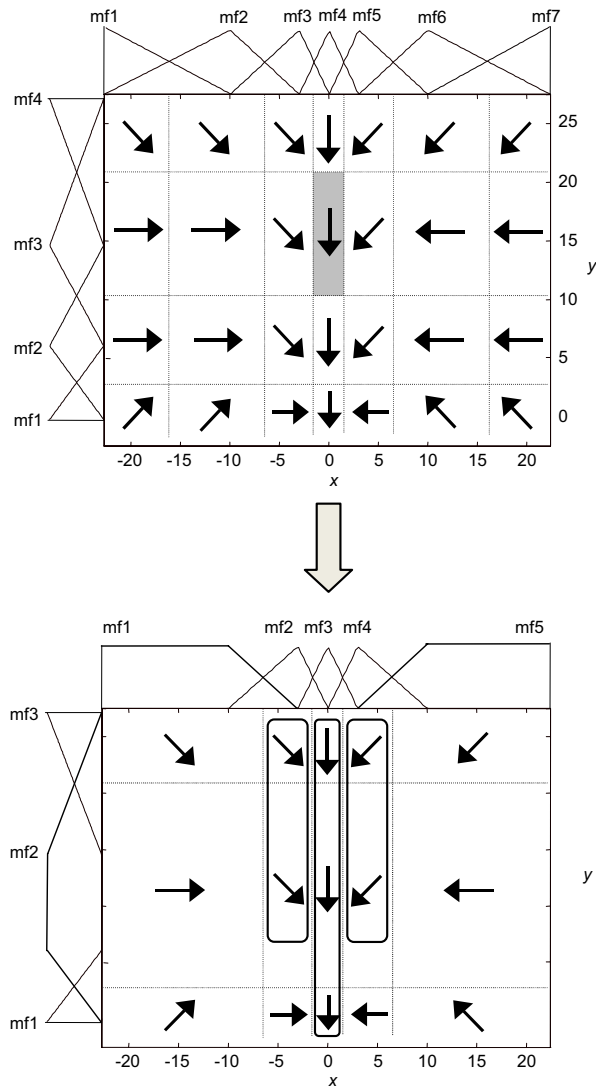


Figure 7: TMU of the truck backer-upper before (above) and after (below) the simplification.

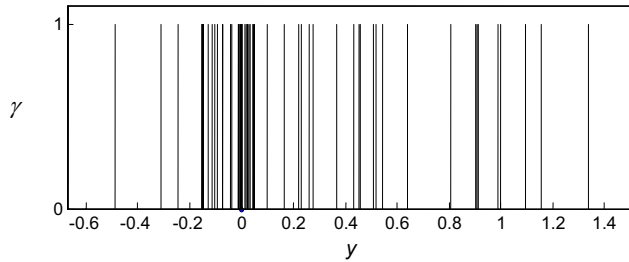


Figure 8: Original singletons of the acidogenic state model

and is a sign of phenomenon of unreliability of the model. Output vocabulary reduction in [14] by which the number of distinct output singletons reduces to 6 from 51 is based on k-means clustering and includes some additional fiddling so as to get all singletons (that are shown in Figure 9) into $[0,1]$. As a consequence, the modeling RMSE increases to 0.056.

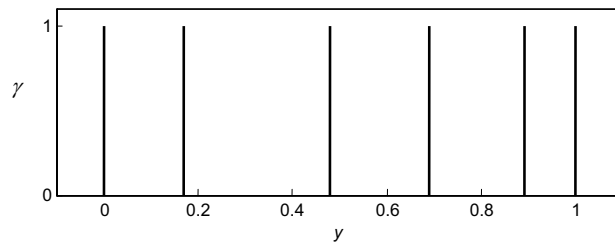


Figure 9: Reduced singleton set of the acidogenic state model [14]

Application of the identification algorithm from Sect. 4 with parameters $\mathbf{res} = 0.15$, $\tau_{min} = 0.2$ (it takes a few attempts to find out that these values are appropriate for the given problem) builds a 54-rule model from scratch (in seven iterations) with RMSE= 0.050 and output singletons that are depicted in Figure 10. It can be seen immediately that in contrast to the original model all singletons remain within $[0, 1]$ and form distinct groups. The course of learning is reproduced in Table 2 from what we can see that the number of rules (R) as well as the number of MFs per individual input variable (in the column labelled partition) are both self-evolving (subject to restrictions imposed by \mathbf{res} and τ_{min}). The resulting input partition (Figure 12) is substantially different from the one belonging to the original OLS model (Figure 11). Note that this input partition does not change through subsequent simplification stages.

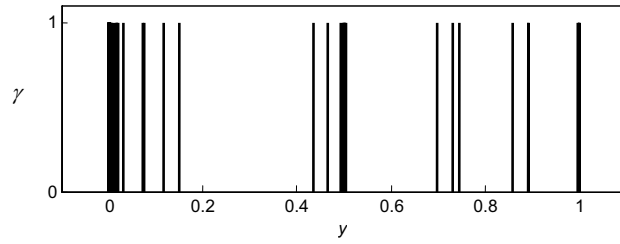


Figure 10: Initial singletons of the newly identified model

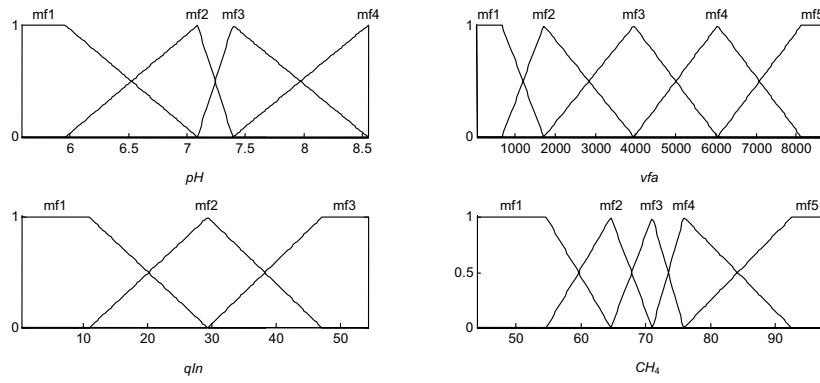


Figure 11: Input partition of the original acidogenic state model [14]

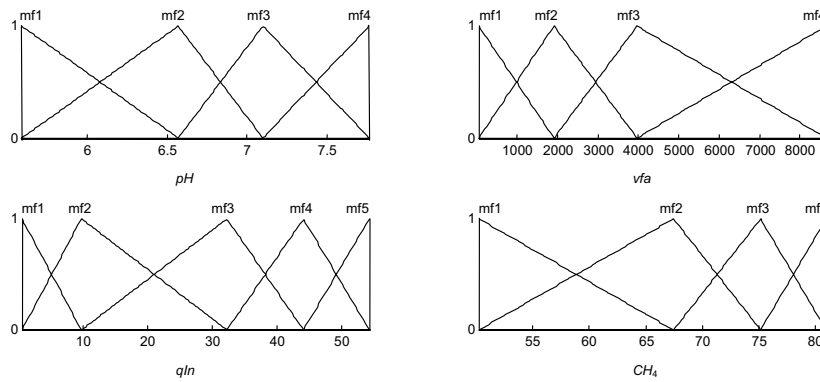


Figure 12: Input partition of the acidogenic state model identified by the proposed algorithm

Table 2: Evolution of the acidogenic state model

iteration	$RMSE$	R	partition
0	0.129	13	2 2 2 2
1	0.227	17	3 3 3 2
2	0.073	32	3 4 4 2
3	0.061	35	4 4 4 2
4	0.056	41	4 4 4 3
5	0.059	45	4 4 4 4
6	0.053	46	4 4 4 4
7	0.050	54	4 4 5 4

In the next step two reduced models are produced by the algorithm from Sect. 5. One of them has 5 singletons and the other one 3. As we see from Figures 13 and 14, all singletons remain conveniently within $[0, 1]$.

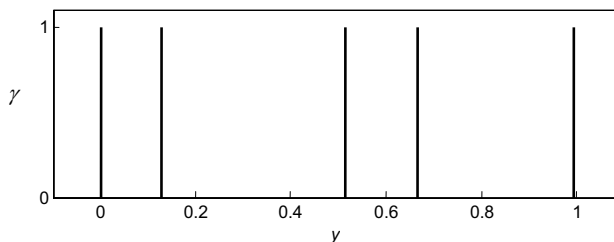


Figure 13: Reduced set of singletons of the newly identified model ($T = 5$)

Finally, we apply redundancy removal tool that does not affect in any way reliability and accuracy of reduced models but only reduces the number of rules in those to 45 and 42, respectively. Final results in terms of R , $RMSE$ and reliability measure (ρ) are given in Table ??, the measure of information loss (J_q) is also provided where applicable.

7.2. Modeling the gas furnace system

The gas furnace data set [8] has been used extensively as a benchmark example for process identification. The data set consists of 296 input-output measurements sampled at a fixed interval of 9 seconds. The measured input $u(k)$ represents the flow rate of the methane gas in a gas furnace and the

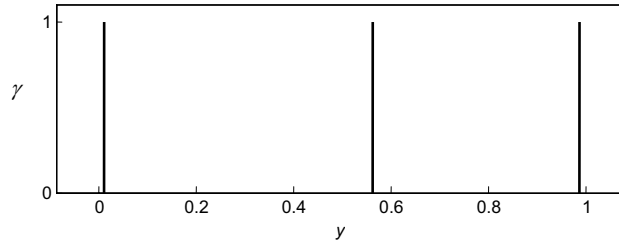


Figure 14: Reduced set of singletons of the newly identified model ($T = 3$)

Table 3: Various acidogenic state models

	original [14]	k-means [14]	proposed $T = 54$	proposed $T = 5$	proposed $T = 3$
RMSE	0.046	0.056	0.050	0.051	0.054
ρ	0.043	0.035	0.013	0.015	0.013
J_q	-	0.017	-	0.007	0.009
R	54	51	53	45	42

output measurement $y(k)$ represents the concentration of carbon dioxide in the gas mixture flowing out of the furnace under a steady air supply. Most studies (e.g. [41, 29, 43, 39, 42]) have used the inputs $y(k-1)$ and $u(k-4)$ which have the highest correlation with the output $y(k)$. Some studies (see Table 5) have used different and/or more inputs.

We apply the proposed algorithm to obtain the models with 2, 4 and 5 inputs (for which there exists comparison material in literature) using the settings $res = 0.2, \tau_{min} = 0.4$ for two-input models; $res = 0.2, \tau_{min} = 0.2$ and $res = 0.3, \tau_{min} = 0.15$ for four-input models; and $res = 0.4, \tau_{min} = 0.1$ for the five-input models. It seems that as the number of input increases res needs to be increased and τ_{min} must decrease to maintain modeling accuracy. Inappropriate definition of these parameters may result in a non-convergent model. The models are identified within 4-13 iterations (with a model with less inputs more iterations are required). Selection of input variables, along with obtained MSEs before (MSE_1) and after vocabulary reduction (MSE_2) as well as other corresponding measures - the number of singletons after vocabulary reduction (T), the number of rules before (R_1) and after redundancy removal (R_2) and reliability measures before (ρ_1) and after vocabulary reduc-

tion (ρ_2) along with the reduction-induced information loss (J_q), are given in Table 4. We can see that vocabulary reduction frequently improves accuracy of the models (because of having a more potent singleton identification procedure) and slightly reduces reliability of the models (ρ remains below 0.4, which is good, considering that the singletons are from the range [45,60]), whereas redundancy removal is mostly (by no surprise) efficient for initially more complex models.

Table 4: Comparison of gas furnace models

inputs	MSE ₁	R ₁	ρ_1	T	MSE ₂	R ₂	J _q	ρ_2
$y(k-1), u(k-4)$	0.187	10	0.295	8	0.167	10	0.139	0.390
$y(k-1), u(k-3)$	0.186	11	0.293	9	0.173	11	0.107	0.373
$y(k-1), u(k-3),$ $u(k-4), u(k-5)$	0.219	27	0.168	8	0.189	23	0.161	0.273
$y(k-1), y(k-3),$ $u(k-3), u(k-6)$	0.297	25	0.245	8	0.190	16	0.231	0.378
$y(k-1), y(k-2),$ $u(k-3), u(k-4),$ $u(k-5)$	0.300	37	0.195	9	0.238	26	0.309	0.391

To put the things into perspective, final figures of Table 4 are compared to those found from literature in Table 5. It must be noted that various researchers have used various techniques ranging from fuzzy relational models to fuzzy neural networks, though a substantial group of researchers utilizing 1-st order TS models can be distinguished ([39, 42, 40, 19, 21]). Furthermore, comparing these results on the basis of anything else than accuracy can be troublesome - even a simple enough figure such as the number of rules can be deceptive because, for example each rule of a 1-st order TS system contains $N + 1$ independent parameters vs. a single one in a 0-th order TS rule, and is also dependent on input variables.

In terms of accuracy, the models identified by the proposed methodology compare surprisingly well with the majority of cited works.

Interestingly, though, the least complex models can be considered the best. In [15] it is brought out that the gas furnace data set is known to represent an approximately linear input-output behavior and this trend is

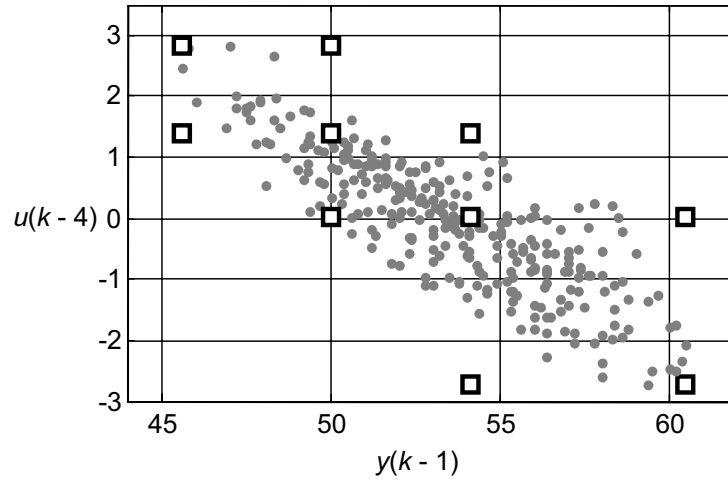


Figure 15: Approximation of gas furnace data. First data set projected onto input plane. Distribution of rule nodes in input space.

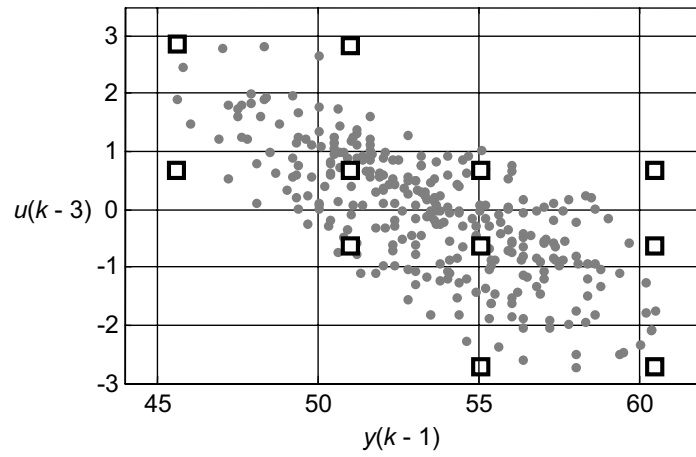


Figure 16: Approximation of gas furnace data. Second data set projected onto input plane. Distribution of rule nodes in input space.

Table 5: Comparison of gas furnace models from literature

study	inputs	MSE	R
Tong [41]	$y(k-1), u(k-4)$	0.469	19
Pedrycz [29]		0.320	81
Xu [43]		0.328	25
Sugeno [39]		0.359	2
Wang [42]		0.158	5
proposed		0.167	10
Sugeno [40]	$y(k-1), u(k-3)$	0.190	6
Kim [19]		0.129	2
proposed		0.173	11
Sugeno [40]	$y(k-1), u(k-3),$	0.190	6
proposed	$u(k-4), u(k-5)$	0.189	23
Lin [21]	$y(k-1), y(k-3),$	0.261	6
proposed	$u(k-3), u(k-6)$	0.190	16
Nie [26]	$y(k-1), y(k-2),$	0.169	45
proposed	$u(k-3), u(k-4), u(k-5)$	0.238	26

also apparent in the relationships between the commonly used predictors $y(k-1)$ and $u(k-4)$ and the output variable $y(k)$.

A look at the Figures 15 and 16 indeed confirms it. These figures also give us an idea how the algorithm places the rule nodes in input space so as to surround the area covered by data and to reproduce a relationship that is not orthogonal to axes (well-known bottleneck of linguistic fuzzy systems).

8. Conclusions

In order to fully exploit the potential of linguistic fuzzy systems in modeling one needs to pay equal attention to accuracy, transparency, complexity and reliability of the identified model.

All these criteria are more easily fulfilled if employed algorithms have built-in mechanisms for preservation some of these properties and enhancing others thus working hand in hand toward the common goal.

Sometimes the synergy of algorithms shows up almost unexpectedly. For example, pre-determined clearance between input MFs of the model not only

improves its readability but also improves convergence. Sometimes vocabulary reduction not only makes the model more interpretable but, again, improves its accuracy.

It can be therefore said with certain assurance that the family of algorithms that were introduced in current paper exhibit the properties that make them a viable tool for identifying transparent, accurate, reliable and moderately complex linguistic fuzzy systems from data.

References

- [1] J. Abonyi, *Fuzzy Model Identification for Control*, Birkhauser, Boston, 2003
- [2] R. Alcalá, P. Ducange, F. Herrera, B. Lazzerini and F. Marcelloni, “A Multiobjective Evolutionary Approach to Concurrently Learn Rule and Data Bases of Linguistic Fuzzy-Rule-Based Systems,” *IEEE Trans. Fuzzy Systems*, vol. 17, No. 5, pp. 1107-1122, 2009.
- [3] J. M. Alonso, L. Magdalena, and S. Guillaume, “HILK: A new methodology for designing highly interpretable linguistic knowledge bases using the fuzzy logic formalism,” *Int. J. Intelligent Systems*, vol. 23, No. 7, pp. 761-794, 2008.
- [4] J. M. Alonso, L. Magdalena, and G. Gonzalez-Rodriguez, “Looking for a good fuzzy system interpretability index: An experimental approach,” *Int. J. Approximate Reasoning*, vol. 51, pp. 115-134, 2009.
- [5] R. Babuska *Fuzzy Modeling for Control*, Kluwer Academic Publishers, 1998.
- [6] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function (Advanced Applications in Pattern Recognition)* , Plenum Press, New York, 1981.
- [7] A. Botta, B. Lazzerini, F. Marcelloni and D. C. Stefanescu, “Context adaptation of fuzzy systems through a multi-objective evolutionary approach based on a novel interpretability index,” *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, vol. 13, No. 5, pp. 437-449, 2009.

- [8] G. E. P. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control*, Holden-Day, San Francisco, 1970.
- [9] J. Casillas, O. Cordon, F. Herrera and L. Magdalena (Eds.), *Interpretability Issues in Fuzzy Modeling (Studies in Fuzziness and Soft Computing, vol. 128)*, Springer-Verlag, Heidelberg, 2003.
- [10] J. Casillas, O. Cordon, M. J. del Jesus and F. Herrera, "Genetic tuning of fuzzy rule deep structures preserving interpretability and its interaction with fuzzy rule set reduction," *IEEE Trans. Fuzzy Systems*, vol. 13, No. 1, pp. 13-29, 2005.
- [11] C.-L. Chen, S.-H. Hsu, C.-T. Hsieh, T.-C. Wang, "A simple method for identification of singleton fuzzy models," *Int. J. Systems Science*, vol. 36, No. 13, 2005, pp. 845-854, 2005.
- [12] S. L. Chiu, "Fuzzy Model Identification Based on Cluster Estimation," *Journal of Intelligent and Fuzzy Systems*, vol.2, pp. 267-278, 1994.
- [13] O. Cordon, F. Herrera, F. Hoffmann and L. Magdalena (eds), *Genetic fuzzy systems: evolutionary tuning and learning of fuzzy knowledge bases*, World Scientific, Singapore, 2001.
- [14] S. Destercke, S. Guillaume and B. Charnomordic, "Building an interpretable fuzzy rule base from data using Orthogonal Least Squares - Application to a depollution problem," *Fuzzy Sets and Systems*, vol. 158, no. 18, pp. 2078-2094, 2007.
- [15] W. Faraq and A. Tawfik, "On Fuzzy Model Identification and the Gas Furnace Data," *Proc. IASTED Int. Conf. Intelligent Systems and Control*, Honolulu, pp. 210-214, 2000.
- [16] J. A. Hartigan and M. A. Wong, "A k-means clustering algorithm," *Applied Statistics*, vol. 28, pp. 100-108, 1979.
- [17] H. Ishibuchi and Y. Nojima, "Analysis of interpretability-accuracy trade-off of fuzzy systems by multiobjective fuzzy genetics-based machine learning," *Int. J. Approximate Reasoning*, vol. 44, No. 1, pp. 4-31, 2007.
- [18] J.-S. R. Jang, C.-T. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*, Prentice Hall, Upper Saddle River, 1997.

- [19] M. S. Kim, C. H. Kim and J. J. Lee, "Evolving compact and interpretable Takagi-Sugeno fuzzy models with a new encoding scheme," *IEEE Trans. Syst., Man, Cybernet., B: Cybernet.*, vol. 36, no. 5, pp. 1006-1023, 2006.
- [20] B. Kosko, "Optimal Fuzzy Rules Cover Extrema," *Int. J. of Intelligent Systems*, vol. 10, no. 2, pp. 249-255, 1995.
- [21] Y. Lin and G. A. Cunningham, "A new approach to fuzzy-neural system modeling," *IEEE Trans. Fuzzy Syst.*, vol. 3, no. 2, pp. 190-198, 1995.
- [22] C. Mencar and A. M. Fanelli, "Interpretability constraints for fuzzy information granulation," *Information Sciences*, vol. 178, No. 24, pp. 4585-4618, 2008.
- [23] R. Mikut, J. Jäkel and L. Gröll, "Interpretability issues in data-based learning of fuzzy systems," *Fuzzy Sets and Systems*, vol. 150, No. 2, pp. 179-197, 2005.
- [24] G. A. Miller, "The magical number seven, plus or minus two: Some limits on our capacity for processing information," *The Psychological Review*, vol. 63, No. 2, pp. 81-97, 1956.
- [25] Y. Nakoula, S. Galichet, and L. Foulloy, "Simultaneous Learning of Rules and Linguistic Terms," *Proc. 5th IEEE Int. Conf. Fuzzy Systems*, New Orleans, pp. 1743-1749, 1996.
- [26] J. Nie, "Constructing fuzzy model by self-organizing counterpropagation network," *IEEE Trans. Syst., Man, Cybernet.* vol. 25, no. 6, pp. 963-970, 1995.
- [27] K. Nozaki, H. Ishibuchi and H. Tanaka, "A simple but powerful heuristic method for generating fuzzy rules from numerical data," *Fuzzy Sets and Systems*, vol. 65, pp. 251-270, 1997.
- [28] W. Pedrycz, "An identification algorithm in fuzzy relational systems," *Fuzzy Sets and Systems*, vol. 13, pp. 153-167, 1984.
- [29] W. Pedrycz, "Applications of fuzzy relational equations for methods of reasoning in presence of fuzzy data," *Fuzzy Sets and Systems*, vol. 16, pp. 163-175, 1985.

- [30] R. Penrose, "A generalized inverse for matrices," *Proc. Cambridge Philosophical Society*, vol. 51, pp. 406-413, 1955.
- [31] A. Riid and E. Rüstern, "Transparent fuzzy systems and modeling with transparency protection," *Proc. IFAC Symp. on Artificial Intelligence in Real Time Control*, Budapest, pp. 229-234, 2000.
- [32] A. Riid, E. Rüstern, "Fuzzy logic in control: truck backer-upper problem revisited," *Proc. IEEE Int. Conf. Fuzzy Systems*, Melbourne, Australia, vol. 1, pp. 513-516, 2001.
- [33] A. Riid and E. Rüstern, "Interpretability of Fuzzy Systems and Its Application to Process Control," *Proc. IEEE Int. Conf. Fuzzy Systems*, London, pp. 228-233, 2007.
- [34] A. Riid, K. Saastamoinen and E. Rüstern, "Error-free Simplification of Transparent Mamdani Systems," *Proc. IEEE Int. Conf. Intelligent Systems*, Varna, vol. 1, pp. 2-8-2-13, 2008.
- [35] A. Riid and E. Rüstern, "A Method for Heuristic Fuzzy Modeling in Noisy Environment," *Proc. IEEE Int. Conf. Intelligent Systems*, London, pp. 468-473, 2010.
- [36] A. Riid and E. Rüstern, "Interpretability Improvement of Fuzzy Systems: Reducing the Number of Unique Singletons in Zeroth order Takagi-Sugeno Systems," *Proc. IEEE Int. Conf. Fuzzy Systems*, Barcelona, pp. 2013-2018, 2010.
- [37] A. Riid, K. Saastamoinen and E. Rüstern, "Redundancy Detection and Removal Tool for Transparent Mamdani Systems," in V. Sgurev, M. Hadjiski, J. Kacprzyk (Eds.), *Intelligent Systems: From Theory to Practice*, Springer-Verlag, Heidelberg, pp. 397-415, 2010.
- [38] E. H. Ruspini, "A new approach to clustering", *Information and Control*, vol 15, pp. 22-32, 1969.
- [39] M. Sugeno and K. Tanaka, "Successive Identification of a Fuzzy Model and its Application to Prediction of a Complex System," *Fuzzy Sets and Systems*, vol. 42, pp. 315-334, 1991.

- [40] M. Sugeno and T. Yasukawa, "A fuzzy-logic-based approach to qualitative modeling," *IEEE Trans. Fuzzy Syst.*, vol. 1, no. 1, pp. 7-31, 1993.
- [41] R. M. Tong, "Synthesis of Fuzzy Models For Industrial Processes: Some Recent Results," *Int. J. General Syst.*, vol. 4, pp. 143-162, 1978.
- [42] L. Wang and R. Langari, "Complex Systems Modeling via Fuzzy Logic," *IEEE Trans. Syst., Man, Cybernet.*, vol. 26, no. 1, pp. 100-106, 1996.
- [43] C. W. Xu and Y. Z. Lu, "Fuzzy model Identification and self-learning for dynamic systems," *IEEE Trans. Syst., Man, Cybernet.*, vol. 17, pp. 683-689, 1987.