Learning a fuzzy rule based knowledge representation

Hartmut Surmann
GMD - German National Research Center for Information Technology
Schloss Birlinghoven, D-53754 Sankt Augustin, Germany
Phone:+49-2241-142518, Fax: +49-2241-142342
email: surmann@gmd.de

ABSTRACT: This paper presents an automatic learning algorithm which generates a fuzzy rule based knowledge representation. While learning the membership functions and rules the internal structure of the rule base is also considered. This is done by definition of 1) a complexity cost function and 2) a minimal Fuzzy System. A Genetic Algorithm is used to estimate the Fuzzy Systems which capture a low comlex and minimal rule base. Optimization of the "entropy" of a fuzzy rule base leads to a minimal number of rules, of membership functions and of subpremises together with an optimal input/output behavior. The resulting Fuzzy System is comparable to systems designed by an expert but with a better performance. The approach is compared to others by a standard benchmark (a system identification process) and different results for symmetric and non-symmetric membership functions are presented.

KEYWORDS: Knowledge representation, Fuzzy System, Genetic Algorithm, Fuzzy Rule Base Entropy, Machine Learning.

I. Introduction

Fuzzy logic controllers model the human decision making process with a collection of rules. In most of the automatically learned Fuzzy Systems, computational effort is spent in finding parameters e.g. fuzzy sets and linguistic terms that give a desired behavior to the system. Since both the rules and the fuzzy sets used in these rules play a crucial role in the outcome of the system, choosing the right rules and fuzzy sets becomes an important issue. To choose the parameters the human decision making process has also to be consider. So additional attention has to be given to the structure of the system. However, sometimes the parametric adjustment of the selected structure may be too poor. So, it does not represent the problem good enough or the behavior is satisfactory but the structure is too complex and the dimension of the problem increases unreasonably. Besides the large tuning period and the significant amount of wasted resources, a structure of unreasonable high complexity could also lead to biasing effects such as the well known overfitting. It is known [3], [12] that a more simple structure leads to a more robust system.

First approaches of structural tuning were based on classification methods applied to the input/output space [1]. Some deterministic [2] and neural based approaches [8] have also been proposed. In 1989 Karr et al. introduced the optimization of fuzzy systems by using genetic algorithms [7]. Since then some genetic based approaches for structural and parametrical tuning have been proposed with the aim to minimize the rule base by e.g. [4], [6], [11] or to minimize the number of membership functions (MFs) per variable by e.g. [18], [6], [5].

A new approach to optimal Fuzzy Systems in which the influence of the entropy of fuzzy rules is introduced was proposed by Surmann et al. 1993 [14], [15]. In contrast to neural approaches minimal entropy of Fuzzy Rule Based Systems (FRBS) leads to rule bases with little overlapping MFs. This, in conjunction with a fast rule processing algorithm, leads to a significant saving of computational resources [15]. In the following the notion of minimal entropy for a FRBS is extended. Based upon this a repetitive step by step genetic method is proposed for both structural and parametrical tuning at the same time. A supervised learning method which is able to generate minimal FRBSs with an optimal input/output behavior by using appropriate parameter values is presented.

The paper is organized as follows. In section 2 a brief introduction to FRBSs and Genetic Algorithms is presented together with the definition of optimal FRBS. In the next section (3), the genetic procedure for finding the FRBSs is given. To illustrate the good basic characteristics of the proposed method the Box & Jenkins data is used as a standard benchmark in section 4. The paper finishes with a conclusion in section 5.

II. Basic terms of Genetic Algorithms and Fuzzy Rule Based Systems

A. Fuzzy Rule Based Systems

Fuzzy Systems are based on Fuzzy Set Theory. A number of rules is used to model the way that a system behaves. Fuzzy rules are constructed in the well known IF–THEN form. When crisp values are given as input for a FRBS, they are converted to degrees of membership in the various linguistic values used by the system. Each crisp value activates a number of rules to some degree. The extent to which rules are activated is calculated first. The *AND* operator is used to combine different input subpremises in the rules. This leads to the total activation of the rule. Supposing that the k–th rule of the system is: IF x_1 is $I_{1,k}$ AND x_2 is $I_{2,k}$ AND $\ldots$ x_n is $I_{n,k}$ THEN y is O_k, where $x_1,\ldots,x_n$ and y represent input and output variables and $I_{1,k},\ldots,I_{n,k}$, O_k their respective MFs. Then the extent to which a rule is activated is calculated as: $\alpha_k = \mu_{I_{1,k}}(x_1)\ AND\ \mu_{I_{2,k}}(x_2)\ AND\ \ldots\ AND\ \mu_{I_{n,k}}(x_n)$ where $\mu_{I_{i,k}}(x_i)$ is the membership value of x_i in the $I_{i,k}$ fuzzy set. The result of the *AND* combination is used as a measure for the truth of the rule. When a rule is activated with a truth value α_k, the inferencing process states that the MF of the output set is $\mu^{inf}_{O_k}(y) = \alpha_k\ AND\ \mu_{O_k}(y)$, which is the fuzzy result of the rule. Here, the minimum operator *min* is used as an *AND* operator for both cases.

The total output fuzzy set is calculated by the compositional rule of inference as $\mu^{total}_{O}(y) = \mu^{inf}_{O_1}(y)\ OR\ \mu^{inf}_{O_2}(y)\ OR\ \cdots\ OR \mu^{inf}_{O_K}(y)$ where it is supposed that the system is described by K rules. In order to get a crisp number as output usually a de-fuzzyfication method is used. The calculation of the fuzzy result function $\mu^{total}_{O}(y)$ and the final crisp value is the bottleneck during computation. Therefore a modified center of gravity algorithm is used to calculate the area A_i and the center of area M_i of each MF before runtime [16]: $A_i = \int O_i(y)dy, M_i = \int y \times \mu_{O_i}(y)dy$. The output value is computed as

$$y_{crisp} = \frac{\sum_{i=0}^{K} \alpha_i \times M_i}{\sum_{i=0}^{K} \alpha_i \times A_i} \qquad (1)$$

Definition 1 (Fuzzy Rule Based System *FRBS*) A Fuzzy Rule Based System $FRBS = (LV, R, T, I,\ T^*, DE)$ is a 6-tuple with a set LV of n linguistic input and one output variables, a set R of k fuzzy rules, a T-Norm T, for evaluation of the subpremises, an implication strategy I, a T-CO-Norm T^*, for evaluation of the fuzzy output sets of the fuzzy rules and a defuzzyfication strategy DE. The set FS is the power set over all FRBSs.

For the automatic learning of the structure of a FRBS one needs to have a definition of the complexity and with it of the costs of a fuzzy system:

Definition 2 (Complexity Cost Function, E) A fuzzy system with k fuzzy rules, n linguistic variables, $k*n$ subpremises and $m = m_1 + \ldots + m_n$ MFs is less complex than one with $k+1$ fuzzy rules, $k*n$ subpremises and m MFs or k fuzzy rules, $(k*n)$+1 subpremises and m MFs or k fuzzy rules, $k*n$ subpremises and $m+1$ MFs.

The weighting of the above three terms depends on the application. A mathematical expression of the complexity function E is given later. On the basis of the cost function E a two step learning procedure is described. Firstly an "opening step" is any operation on a FRBS that increases the cost function E. Secondly a "closing step" is any operation on a FRBS that decreases the cost function E. Any opening step results in a new fuzzy system with more rules, MFs or subpremises. Whereas any closing step results in a new fuzzy system with less rules, MFs or subpremises. Finally we are interested in minimal fuzzy systems and we define them by using the cost function E:

Definition 3 (Minimal Fuzzy-System) Let us approximate a real system $f(\vec{x})$, using a fuzzy system $X(\vec{x})$ and let $\varepsilon > 0$ be a maximal accepted error limit. A fuzzy system X with $sup\{\mid f(\vec{x}) - X(\vec{x}) \mid \quad \mid \vec{x} \in U\} \leq \varepsilon$ is called minimal if there is no other fuzzy system Y with $sup\{\mid f(\vec{x}) - Y(\vec{x}) \mid \quad \mid \vec{x} \in U\} \leq \varepsilon$ and $E(Y) < E(X)$, where E is the complexity cost function.

The real system $f(\vec{x})$ is represented by referential vectors $\vec{r}$.

B. Genetic Algorithms

Traditional optimization methods are based on the fact that certain functions are differentiable. Unfortunately, in many real world problems such functions can not be defined. But even if they can, gradient search methods may not find global optimal solutions. A possible way to overcome such problems is to use genetic algorithms (GAs). Generally a GA consists of a problem, a number of encoded solutions for that problem, some genetic operators which produce new solutions and a fitness function which says how good a particular solution for the problem is (Fig. 1).

Usually the *fitness function* describes the aggregation of some desired properties for the solutions and is not necessarily differentiable. Each solution is *encoded* as a chromosome by binary or real values. A *population* consists of a number of individuals represented by chromosomes. A population at a certain time step is a *generation*. Genetic operators are applied to each generation to produce the next generation. Common genetic operators are *selection, crossover* and *mutation*.

Genetic Algorithm
begin (1)
$t = 1$
Initialize $Population(t)$
Evaluate fitness $Population(t)$
While (Generations < Total Number) **do**
begin (2)
select $Population(t+1)$ out of $Population(t)$
Apply Crossover on $Population(t+1)$
Apply Mutation on $Population(t+1)$
Evaluate fitness $Population(t+1)$
$t = t+1$
end (2)
end (1)

Fig. 1. The Structure of Standard Genetic Algorithm

During selection individuals with high fitness values within the current population are selected to build the basis for the new generation. Crossover is a way of creating new solutions by randomly selecting two chromosomes of previous solutions from the gene pool and exchanging portions of their strings. Mutation is performed upon a selected chromosome by randomly changing a part of its coded value. Mutation is needed to ensure diversity in the population.

III. Optimizing FRBSs

Up to now a lot of work has been done in combining fuzzy systems and genetic algorithms [3]. Fuzzy–genetic combinations can be classified in two categories. On one hand fuzzy techniques are used to improve GA behavior [6] and to model GA components [9]. On the other GAs are used to optimize the structure of the Fuzzy System and the input/output behavior [6]. In the following attention will be paid to the second category and especially to achieving an optimal structure of a fuzzy system by means of GAs.

A. Membership function shape and rule base construction

Membership functions will be constructed from probability density functions by $\mu(x) = \lambda p(x)$. The constant λ is calculated using the constraint $\sup \mu = 1$. Here the Gaussian probability distribution $p(x) = \varphi(x;m,\sigma^2)$ is chosen, because this distribution fits a lot of real world problems. Therefore, the membership functions used in the system are defined by

$$\mu(x) = \sigma\sqrt{2\pi}\,\varphi(x;m,\sigma^2) = \exp\left(-\frac{(x-m)^2}{2\sigma^2}\right). \quad (2)$$

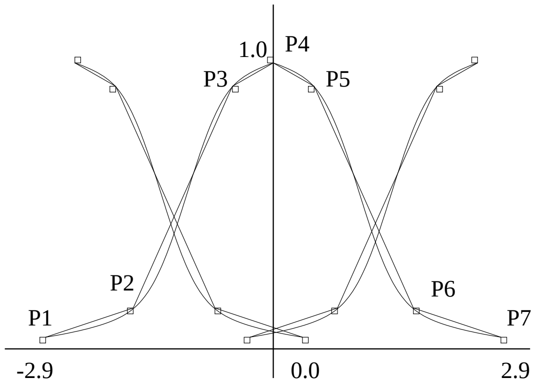

Fig. 2. Linearized model of Gaussian MFs. Six lines are used to describe them.

The MFs are approximated by 6 straight lines and they are limited by $\mu(x) = 0 \;\; for\; x \notin [m-2.9\sigma, m+2.9\sigma]$. The line approximation speeds up the evaluation process of the FRBS by a factor two [16]. Start and end points of these six lines are the roots of the second derivative of the normalized Gaussian MF ($m = 0$, $\sigma^2 = 1$). Special margin MFs are defined at the left and right border of a linguistic variable (Fig. 2):

$$\mu_{\text{left}}(x) = \begin{cases} 1 & , x < 0 \\ e^{-\frac{1}{2}x^2} & , x \geq 0 \end{cases} \quad (3)$$

$$\mu_{\text{right}}(x) = \begin{cases} 1 & , x \geq 0 \\ e^{-\frac{1}{2}x^2} & , x < 0 \end{cases} . \quad (4)$$

Non–symmetric Gaussian MFs consist of two parts (left and right) with a common mean but different σ_{left} and σ_{right} for the left and right side. They are used to achieve more flexible systems.

B. Using Genetic Algorithms

When the optimization process is initialized, it is desirable to produce a generation that has enough parameters to gain flexibility. For that reason, at the beginning of the optimization process the structure of the Fuzzy System is opened. This means that for each multiply used MF in the rule base, copies of the function for each rule to which it participates are produced. Thus each rule has its own MFs. All copies are equal at this point of the process, but they are allowed to vary independently in the next steps. Now, the GA has the possibility to eliminate the extra parameters while running.

The chromosomes of the GA are built by two or three genes (parameter sets). Two genes are selected if the MF is symmetric, three if it is non–symmetric (mean gene, right sigma gene, left sigma gene). The first gene consists of the means of the MF, while the second one consists of the respective sigmas or σ_{right} and σ_{left}. Each mean and sigma is converted from a real number to a fixed binary number of $l = 8, 16$ bits accuracy using the *gray encoding* scheme. Graycoding makes genetic operators more robust against single bit changes: The value of the gray-coded gene is only 1 higher or lower, if one bit of the encoding is changed. On the contrary several bits of a binary coded number may change if it increases or decreases by 1, e. g. 4 bits if 7 increases to 8.

Every chromosome S contains all fuzzy set parameters m_{ij} and $\sigma_{ij}^{(left,right)}$, where $i = 1, \ldots, n$ is the number of variables and $j = 1, \ldots, q_n$ is the number of respective MFs per variable.

During fitness evaluation the reverse process (decoding) is done to get back the means and sigmas as reals. With this real values the MFs of the fuzzy variables are constructed and the fitness function is evaluated. The length of a string is: $L = 2l \sum_n q_n$ for the symmetric and $L = 3l \sum_n q_n$ for the non–symmetric case.

B.1 Objective Function

The definition of a suitable objective function is basic for the genetic process. It expresses in a formal way the desired properties of solutions. Here we define an objective function with two main items. The first part considers the I/O behavior and the second part the structure and complexity of the resulting fuzzy system. The structural part of the objective function has three items.

$$OF = OF^{I/O} \times (OF^E \times OF^S \times OF^{UZ}). \tag{5}$$

OF is the objective function that the GA has to maximize.

The only property for the I/O behavior of the FRBS is the maximization of the reciprocal of the Mean Square Error (MSE).

$$OF_{pv}^{I/O} = pv / \sum_{i=1}^{pv} \sum_{j=1}^{n_{out}} (r_{i,j} - o_{i,j})^2, \tag{6}$$

where pv is the number of referential data pairs, n_{out} is the dimension of the output vector, $o_{i,j}$ is the current computed output value of the FRBS and $r_{i,j}$ is the respective referential value ($i = 1 \ldots pv$, $j = 1 \ldots n_{out}$).

Three criteria for the structure of the FRBS are also considered. The first criterion (OF^E) for the structure of a FRBS is the *degree of fuzziness* or the *entropy* of a FRBS and was introduced in [15]. It is defined by the average number of activated rules: $R_\phi = \frac{1}{pv} * \sum_{i=1}^{pv} R_{cur,i}$, where $R_{cur,i}$ is the current number of activated rules for the i–th input/output pair and pv the number of those pairs. If the number of activated rules is minimal i.e. 1 then a FRBS is maximal understandable by humans. FRBS with a high number of activated rules behave like a neural network i.e. a lot of rules (neurons) determine the output values. To decrease the entropy of a FRBS and with it the overlap of the MFs, a maximal number R_{max} of activated fuzzy rules has to be defined and the objective function is extended as:

$$OF^E = \begin{cases} \dfrac{1}{(\frac{R_{act}}{R_{max}} - 1)a + (\frac{R_\phi}{R_{max}} - 1)b + 1} & , R_{max} < R_{act} \\ \dfrac{1}{(\frac{R_\phi}{R_{max}} - 1)b + 1} & , \text{else.} \end{cases} \tag{7}$$

where a and b are predefined weighting factors. The influence of the entropy part of the objective function for the benchmark of section IV is shown in figure 3 and 4.

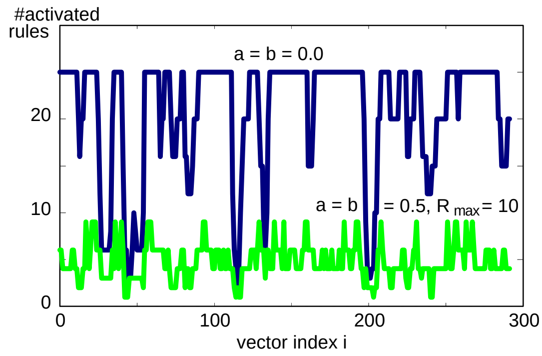

Fig. 3. Activated rules with and without entropy consideration

The second criterion (OF^S) for the structure of the FRBS is the number of MFs. For that, MFs with high degrees of overlap are counted. Such nearly equal MF pairs are desirable, because they can be easily unified to only one MF. The criterion for the highest possible number of similar MFs is :

$$OF^S = \left(\frac{s_{no}}{n_{in,out} R_{total}} \right) \gamma + 1 \tag{8}$$

where s_{no} is the number of similar MFs (see below), $n_{in,out}$ is the total number of input and output variables, R_{total} is the total number of rules and $\gamma \in \mathbf{R}$ is a predefined weighting factor for the number of similar MFs.

Let suppose we have two MFs denoted by A and B which are described by 7 Points ($P_i(x_{Ai}, y_{Ai})$, $P_i(x_{Bi}, y_{Bi})$) because a linearized model of Gaussian MFs is used (Fig 2). The average width is defined by

$$w = \frac{|x_{A2} - x_{A6}| + |x_{B2} - x_{B6}|}{2}. \tag{9}$$

Two MFs are similar if the distance of their second, third, fifth and sixth points is small enough compared to the average width of the MF. This similarity criterion is used

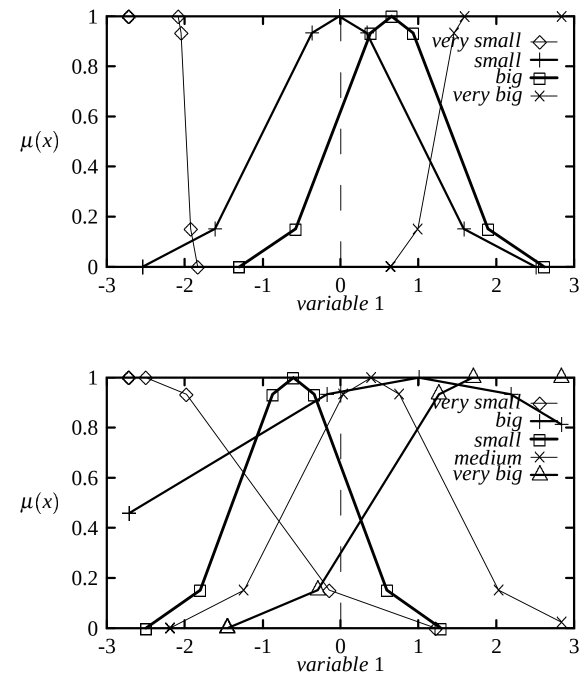

Fig. 4. *Membership functions with and without entropy consideration.*

for symmetric and non-symmetric cases:

$$s_{no} = \sum_{i}^{no} s_i, \ \ with \ \ s_i = \begin{cases} 1 & , \frac{w}{10} > |x_{A2} - x_{B2}| \ \& \\ & , \frac{w}{16} > |x_{A3} - x_{B3}| \ \& \\ & , \frac{w}{10} > |x_{A6} - x_{B6}| \ \& \\ & , \frac{w}{16} > |x_{A5} - x_{B5}| \\ 0 & , \text{else.} \end{cases} \tag{10}$$

The third criterion (OF^{UZ}) for the structure of the FRBS are *never activated* ($\mu(x) = 0, \ \forall x \in U$) and *always completely activated* ($\mu(x) = 1, \ \forall x \in U$) MFs. On the one hand, a subpremise of a rule can be eliminated if the belonging MF is always completely activated for all of the training pairs. On the other hand, MFs which are never activated can be eliminated together with all the rules that they participated. The criterion for the highest possible number of zero and unity MFs with the predefined weighting factors $\zeta, \eta \in \mathbf{R}$ is:

$$OF^{UZ} = \left(\frac{u_{no}}{n_{in,out} R_{total}} \right) \zeta + \left(\frac{z_{no}}{n_{in,out} R_{total}} \right) \eta + 1, \tag{11}$$

where u_{no} is the number of unity MFs, z_{no} is the number of zero MFs. Unity MFs are replaced by a trapeze covering the total variable domain, while zero MFs are replaced by an impulse.

B.2 Producing new Generations

After estimating the fitness value of each chromosome in the current population, genetic operators are applied to produce the new generation. Two new operators are introduced: *Set zero-unity* is used to produce unity and zero functions. It randomly selects a MF for each input/output variable and changes its sigma value to infinity ($+\infty$) in case of a unity MF or to zero in case of a zero MF. The other genetic operator called *set similar*, selects randomly for each input/output variable a membership function and makes it equal to the MF that is most similar to it (Eq. 10).

A mate pool is built by selecting a predefined percentage of the best individuals from the current population (truncation selection). Each chromosome of the new population is produced by randomly selecting its two parents from the mate pool. The crossover operator is applied to them to produce two children. Crossover is done for each gene of the two chromosomes and not for the total chromosome. Mutation, set similar and set zero–unity operators are randomly applied to the selected children as described above. The last two operators are used only in case of FRBSs with MSE less than the predefined upper limit.

B.3 Fuzzy Rule Based System Minimization

To achieve a faster convergence only the reciprocal of the mean square error (MSE) is used as an optimization criterion in the first steps of the GA: $OF = OF^{I/O}$ (eq. 5). When the GA produces FRBSs with MSE less than a predefined threshold ε, the optimization criterion is changed to the global fitness function OF of equation 5. The above is according to the definition of minimal fuzzy systems of a predefined error limit (ε of Def. 3).

After *opening* the structure of the rule base at the beginning of the GA it now has to be *closed* back. Rules with a zero MF ($\sigma = 0$) are deleted, unity MFs ($\sigma = +\infty$) are eliminated from the rules and similar MFs ($\sigma_i \approx \sigma_j$ and $m_i \approx m_j$, $i \neq j$) are unified by the genetic operators. The unification of similar MFs results in a new MF with the average of the given means as the new mean and the maximum of the given sigmas as the new sigma. Simulations showed that 50–100 generations are enough for the GA to stabilize the FRBS structure. After the FRBS structure is stable a closing step is done. Then the whole process repeats with this new structure.

If no more closure steps can be applied, the most compact structure of the FRBS has been found. Nearly optimal means and sigmas for the MF have been estimated and the GA terminates. It is well known that the estimation of an exact global optimum is very time consuming with GAs. For that reason an easy local optimization method is used after the GA to climb the "remaining of the hill". It starts by selecting the first encoded parameter value and a small increment or decrement is applied to it. If that change results in an FRBS with a better performance, a new small change is done in the same direction until those changes do not increase the fitness function any more. This procedure is done for all coded parameters. The process terminates if no better performance can be gained.

IV. Application - Gas Furnace Data

The design algorithm will be illustrated by means of a system identification example. A number of simulations are performed on the data set of Box & Jenkins' *gas furnace data* which is a common benchmark and a collection of representative results is given. The task is to build a rule base model from the referential data set which identifies the process. The data set consists of 296 pairs of input/output observations. The input is the gas flow rate into the furnace and the output is the concentration of CO_2 in the exhausted gas.

As well as in the literature, two input variables $n_{in} = 2 : x(t-\tau_1),\ y(t-\tau_2)$ and one output variable $y(t)$ are chosen. Here, $x(t-\tau_1)$ denotes the input at time $t-\tau_1$ and $y(t-\tau_2)$ the output of the process at $t-\tau_2$. With $\tau_1 = 4$ and $\tau_2 = 1$, the referential data set contains $p = 296 - \tau_1 = 292$ elements. The above data set is used for training and as validation set for testing.

For the initialization of the FRBS 4 MFs for each variable are used together with a rule base of 14 rules. So, after the FRBS is opened a system of 14 rules and 14 MFs per variable is selected. Different strategies are applied with symmetric and non–symmetric MFs. 1000 time steps are used for the GA. A closure step takes place 60 generations after the error limit has been reached. After a closure step a new genetic process is started to continue the optimization.

Literature. Symmetric FRBS

MSE	Init. Rules	Rules	Author
0.469	7x6	19	Tong [17]
0.355	5x5	6	Sugeno et.al [13]
0.320	9x9	81	Pedryz [10]
0.328	5x5	25	Xu-Lu [19]
0.172	7x7	38	Abreu et.al. [1]
0.138	3x5	15	Surmann [[15]]

TABLE III

RESULTS FOR THE SAME PROBLEM FROM LITERATURE. THE COLUMNS SHOW THE MSE, THE STRUCTURE OF THE INITIAL RULE BASE, THE FINAL NUMBER OF RULES AND THE AUTHOR.

Representative results are shown in the tables I and II. The number of unities is equal to the number of eliminated subpremises. It is worth to mention that none of the previous approaches in literature focused on the elimination of rules or MFs. Compared to the results from literature in table III with our proposed method it is possible to achieve much simpler FRBSs and also the best input/output behavior of all approaches.

V. Conclusion

This paper presents a learning algorithm for fuzzy rule based knowledge representations. Both structural and parametrical tuning of Fuzzy Systems are based of 1) a complexity cost function and 2) a minimal Fuzzy System. In contrast to other known approaches where only parametrical tuning takes place, here optimization of the entropy and the complexity of the fuzzy rule base leads to a minimal number of rules, of membership functions and of subpremises together with an optimal input/output behavior. In contrast to neural approaches the resulting Fuzzy System is comparable to systems designed by an expert but with a better input/output behavior. The most characteristic features of the proposed method were illustrated by means of a numerical example.

In future more attention should be given to structural tuning because it highly impacts the capabilities of a system. Simpler structures lead to systems that can be optimized more easily and that even more have a better performance.

Acknowledgment Special thanks to Claudia Hirsch for hints and corrections.

References

[1] Antonio Abreu, LuisCustodio Carlos Pinto-Ferreira, "Fuzzy Modeling: a Rule Based Approach", Proceedings of the fifth IEEE International Conference on Fuzzy Systems, pp. 162-168, 1996.

[2] G. Castellano, G.Attolico, A.Distante, "Automatic generation of fuzzy rules for reactive robot controllers", Robotics and Autonomous Systems 22, pp. 133-149, 1997.

[3] O. Cordon, F. Herrera, M. Lozano, "On the bidirectional Integration of Genetic Algorithms and Fuzzy Logic", http://www.bioele.nuee.nagoya-u.ac.jp/wec2/papers/p003.html.

[4] P.Y. Glorennec, "Constrained Optimization of FIS using an Evolutionary method", Genetic Algorithms and Soft Computing, Studies in Fuzziness and soft Computing series, Physica-Verlag, 1995.

[5] H. Gürocak, "A genetic-algorithm-based method for tuning fuzzy logic controllers", Fuzzy Sets and Systems 108, pp 39-47, 1999.

[6] F. Herrera, J.L. Verdegay (Eds.), "Genetic Algorithms and Soft Computing", Vol. 8 of Studies in Fuzziness and Soft Computing, 1996.

[7] C.L. Karr, L.M. Freeman, D.L.Meredith, "Improved fuzzy process control of spacecraft autonomous rendezvous using a genetic algorithm", SPIE Intelligent Control and adaptive Systems, vol 1196, pp. 274-288, 1989.

Symmetric FRBS

MSE	E.Limit	MFs	Unity	Rules	activated rules	α, b γ, ζ, η
0.121	0.16	4-5-6	1	7	4.73	0.5, 0.5, 0.5, 0.2, 0.3
0.116	0.15	4-7-7	2	8	5.19	0.5, 0.5, 0.4, 0.2, 0.3
0.120	0.15	4-6-6	2	8	4.49	0.5, 0.5, 0.15, 0.15, 0.15

TABLE I

RESULTS FOR *gas furnace data* OBTAINED FROM FRBS STRUCTURES WITH SYMMETRIC MFS. THE COLUMNS SHOWS THE MSE, THE UPPER ERROR LIMIT, THE NUMBER OF MFS PER VARIABLE ($input_1$, $input_2$, $outpout$), HOW MANY UNITY MFS EXIST, THE NUMBER OF RULES, THE AVERAGE NUMBER OF ACTIVATED RULES AND THE VALUES OF PREDEFINED FACTORS USED.

Non–Symmetric FRBS

MSE	E.Limit	MFs	Unity	Rules	activated rules	α, b γ, ζ, η
0.117	0.15	8-8-6	0	8	3.34	0.5, 0.5, 0.2, 0,1, 0.2
0.119	0.16	6-7-5	0	8	3.31	0.5, 0.5, 0.3, 0.2, 0.1
0.121	0.16	7-7-7	2	8	3.62	0.5, 0.5, 0.3, 0.4, 0.2

TABLE II

RESULTS FOR *gas furnace data* OBTAINED FROM FRBS STRUCTURES WITH NON-SYMMETRIC MFS.

[8] M. Lee, S.Y. Lee, Ch.H. Park, ”Neuro–Fuzzy Identifiers and Controllers”, Journal of Intelligent and Fuzzy Systems, Vol.2, pp. 1-14, 1994.

[9] W. Pedryz, M. Reformat, ”Genetic Optimization with Fuzzy Coding”, Genetic Algorithms and Soft Computing, Studies in Fuzziness and soft Computing series, Physica-Verlag, 1995.

[10] W. Pedryz, ”An identification Algorithm in fuzzy relational systems”, Fuzzy Sets and Systems 13, pp 1-25, 1991.

[11] K.Shimojima, N. Kubota, T. Fukuda, ”Virus Evolutionary Genetic Algorithm for Fuzzy Controller Optimization”, Genetic Algorithms and Soft Computing, Studies in Fuzziness and soft Computing series, Physica-Verlag, 1995.

[12] T. Slawinski, P. Krause, A. Krone, H. Kiendl, U. Hammel, D. Wiesmann ”Fuzzy Adaption der evolutionren Regelsuche im Fuzzy-ROSA–Verfahren”, 9. Workshop Fuzzy Control des GMA-FA 5.22, November 1999.

[13] M. Sugeno, T. Yasukawa, ”Linguistic Modeling based on Numerical Data”, Proc. of IFSA ’91, Brussels 1991.

[14] H. Surmann, A.Kanstein, K.Goser, ”Self-Organizing and genetic Algorithms for an Automatic Design of Fuzzy Control and Decision Systems”, in Proceedings of the First European Congress on Fuzzy and Intelligent Technologies, EUFIT’93, Aachen, pp. 1097-1104, 1993.

[15] H. Surmann, ”Automatisierter Entwurf von Fuzzy Systemen”, Phd, VDI-Verlag, Reihe 8, Nr. 452, priced with the AKI-Phd. award, 1995.

[16] H. Surmann, A. P. Ungering, ”Fuzzy Rule Based Systems on general purpose processors”, IEEE MICRO, Special issue on fuzzy systems, pp. 40-48, 1995.

[17] R.M.Tong, ”The evaluation of fuzzy models derived from experimental data”, Fuzzy Sets and Systems, no. 4, pp. 1-12, 1980.

[18] C.-C. Wong, N.-S. Lin, ”Rule extraction for fuzzy modeling”, Fuzzy Sets and Systems, no. 88, pp. 23-30, 1997.

[19] C.-W. Xu, Y.-Z. Lu, ”Fuzzy model Identification and self learning for dynamic systems”, IEEE Transactions on Systems, Man and Cybernetics, pp. 683-689, 1987.