Genetic Algorithms in Transparent Fuzzy Modeling

Andri Riid, Pavel Jartsev and Ennu Rüstern

Department of Computer Control, Tallinn Technical University, Ehitajate tee 5, 19086, Tallinn, Estonia, andri@dcc.ttu.ee

ABSTRACT: This paper observes the modeling issues of non-differentiable fuzzy systems and transparency of the modeled system. Transparency conditions and constraints for standard (Mamdani) and Takagi-Sugeno type fuzzy systems are suggested. Genetic algorithms, considered a serious attempt toward universal optimization tool for fuzzy systems, are capable of simultaneous structure determination and parameter identification if suitably configured. Included are several examples of nonlinear functions modeled with genetic algorithms.

KEYWORDS: Fuzzy modeling, transparency, genetic algorithms.

1 Introduction

The optimization algorithms can be divided into two groups - gradient-based and gradient-free algorithms. Gradient-based approaches are basically local search techniques and not very well performing when the approximated function has several local minima. Several hybrid approaches have been proposed to solve this problem in fuzzy modeling, most notably ANFIS [1], with good results.

ANFIS and other gradient-based algorithms, however, can be applied only to a limited group of fuzzy systems (first or zeroth order prod-sum Sugeno systems), because of the requirement of differentiability.

While there is no universal fuzzy approximation algorithm yet, the need for efficient and flexible algorithms is persistent. Natural evolution-inspired genetic algorithms (GAs) are a good candidate. Our purpose here is to demonstrate that GAs when suitably configured, are potentially able to solve several problems in fuzzy modeling (including the one of transparency).

2 Transparency

First fuzzy logic controllers were derived on the basis of the experience of human operators [2]. The lack of any systematic methodology for such design initiated the birth of Sugeno-type fuzzy systems [3] with half-linguistic/half-functional rules. Further research in this field has lead to algorithms with what the semantic clarity of fuzzy systems has been sacrificed to approximation properties [1]. There are authors, though, who have explored the issues of semantic clarity of fuzzy systems in their works, developing the subject further. The need for constraints for preserving the semantic clarity (i.e. transparency) of the system during the training has been acknowledged in some works [4], [5], although the authors have not developed the idea to the full extent. Some authors [6], [7] have proposed algorithms that produce *de facto* transparent fuzzy systems. We propose the following definition of transparency, the extension of the one introduced in [8]:

Consider a multi input/multi output (MISO) standard fuzzy system

$$\begin{cases} \text{IF } x_1 \text{ is } A_{1r} \text{ AND} \ldots \text{AND } x_i \text{ is } A_{ir} \text{ AND} \ldots \\ \text{AND } x_N \text{ is } A_{Nr} \text{ THEN } y \text{ is } B_r, \\ r = 1 \ldots R \\ y = Y\left(\bigcup_{r=1}^{R}\left(\bigcap_{i=1}^{N}\mu_{ir}(x_i)\right)\cap\gamma_r\right) \end{cases}, \tag{1}$$

where A_{ir} and B_r denote the linguistic labels of the i^{th} input variable and the output, associated with the r^{th} rule, having one-to-one correspondence with normal convex membership functions (MFs) μ_{ir} and γ_r, respectively; x_i denotes the numerical value of the i^{th} input variable and *Y(*)* denotes the defuzzification function (center-of-gravity (CoG), mean-of-maximum (MoM), etc.) used.

Definition: r^{th} rule of the fuzzy system (1) is transparent if it's activation degree

$$\tau_r = \bigcap_{i=1}^{N}\mu_{ir}(x_i) = 1, \tag{2}$$

results in system's output

$$y = \text{core}(\gamma_r(y)). \tag{3}$$

To satisfy (2,3), μ_i^s, the MFs of the i^{th} input variable must be defined so that

$$\forall x_i \in X_i : 0 < \sum_{s=1}^{S_i}\mu_i^s(x_i) \le 1, \tag{4}$$

where S_i denotes the number of MFs per i^{th} variable and $X_i = \left[X_i^{\min}, X_i^{\max}\right]$ denotes the domain of the i^{th} input variable. In case of CoG defuzzification method, output MFs must satisfy (5), i.e. only symmetric MFs are allowed.

$$Y_{cog}(\gamma_r(y)) = \frac{\int_{y_{\min}}^{y_{\max}} y\gamma_r(y)dy}{\int_{y_{\min}}^{y_{\max}} \gamma_r(y)dy} = \mathrm{core}(\gamma_r(y)) \quad (5)$$

With maximum-based defuzzification algorithms (MoM, SoM, LoM), however, (5) is not needed, because these algorithms take care of output transparency.

For Sugeno class of systems, (4) similarly solves the problem of transparency, and first-order Sugeno systems, not violating (4), can be interpreted as piecewise linear input-output mappers. The main issue with Sugeno systems is that commonly used classes of MFs - Gaussian, generalized bell - are non-compact. Bikdash [5] has proposed compact functions such as cubic spline and cosinus that can be easily used to implement (4).

3 Genetic algorithms

Inspired by Darwin's evolution theory "survival of the fittest", GAs were introduced by John Holland in early eighties [9] and have been explored and exploited by many authors since then.

GAs simulate those processes in natural selection which are essential to evolution and are able to find solutions to real world problems if they are suitably encoded. They work with a population of "individuals" each representing a possible solution to a given problem.

It is assumed that a potential solution to a problem may be represented by a set of parameters. These parameters (genes) are joined together to form a chromosome. It is generally believed that the ideal is to use a binary alphabet.

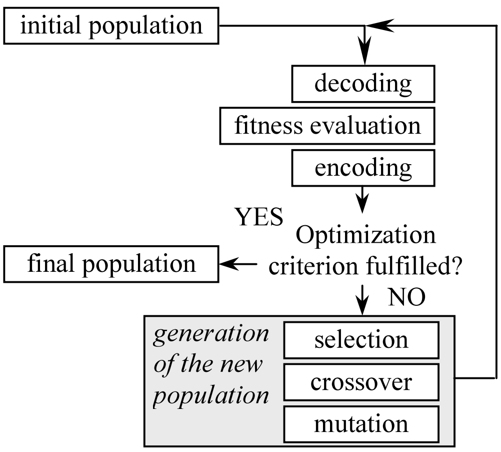

Fig. 1. Classic genetic algorithm.

Each individual is assigned a fitness score according to how good a solution to the problem is. The highly fit individuals (selection) are given opportunities to "reproduce" by "cross breeding" with other individuals. Crossover takes two individuals and cuts their chromosome strings at some randomly chosen position. The "tail" segments are then swapped over to produce two new full length chromosomes (one-point crossover). A whole new population of possible solutions is thus produced. Crossover is not usually applied to all pairs selected for mating but with a likelihood typically between 0.6 and 1.0. Mutation is applied to each child individually after crossover and randomly alters each gene with a small probability (~0.001). Mutation adds a small amount of random search

If the GA has been designed well, the population will converge to an optimal solution of the problem.

The above described classic GA is depicted in Fig. 1.

4 Fuzzy system identification

Fuzzy system identification can be considered a typical modeling problem, including structure determination (variables, partition style, the number of MFs per each variable, number of rules, contents of the rulebase, etc.) and parameter identification (MF parameters). Typically, most optimization algorithms developed so far (e.g. gradient descent), deal with parameter identification. Structure determination problem relies heavily on human expertise, although, some research in this direction has been conducted [10]. The crucial point of structure determination is the definition of the rulebase that directly affects the modeling quality and limits the possibilities of parameter identification.

The common but not efficient solution to the problem is to assign individual output MF to each rule so that rulebase tuning is indirectly carried out in parameter identification phase. This unavoidably leads to a large number of tuning parameters if the number of inputs and/or the number of MFs per variable is large (curse of dimensionality).

GAs, on the other hand, solve this problem efficiently, using integrated chromosome that allows simultaneous MF parameter and rulebase training (although both can be trained individually if such need arises). We demonstrate it in the next section.

5 Parameter encoding/decoding

Typically, the goal of modeling is to reduce the approximation error, e.g. absolute average error

$$\varepsilon = \frac{1}{K}\sum_{k=1}^{K} \left| y_k - \hat{y}_k \right|, \quad (7)$$

minimizing the average difference between y (the modeled function) and $\hat{y}$ (output of the fuzzy system), consisting of K data pairs, by optimizing the parameters of the fuzzy system (1).

To accomplish that, the chromosomes are composed in the following manner: Each chromosome contains two substrings. First substring contains all encoded MFs of the system and the configuration depends on the type of the MFs used. In this work we use triangular functions and employ an input partition style proposed by Jager [7].

$$\mu_i^s(x_i) = \begin{cases} \dfrac{x - a_i^{s-1}}{a_i^n - a_i^{n-1}}, & a_i^{s-1} < x_i < a_i^s \\ 0, & a_i^{s+1} < x_i < a_i^{s-1} \\ \dfrac{a_i^{s+1} - x}{a_i^{s+1} - a_i^s}, & a_i^s < x_i < a_i^{s+1} \end{cases}, \tag{7}$$

($s = 1 \ldots S_i$). (7) forms a partition therefore ensuring that

$$\forall x_i \in X_i : \sum_{s=1}^{S_i} \mu_i^s(x_i) = 1 . \tag{8}$$

Because of CoG defuzzification, symmetric triangular output MFs must be used, given by

$$\gamma^t(y) = \begin{cases} -(s^t/2)(y_j - a^t) + 1, & a^t - s^t/2 < y_j < a^t \\ (s^t/2)(y_j - a^t) + 1, & a^t < y_j < a^t + s^t/2 \\ 0, & a^t + s^t/2 < y_j < a^t - s^t/2 \end{cases}, \tag{9}$$

where t = 1…T (T - the number of MFs per output variable).

Another option is to use singleton consequent MFs:

$$\gamma^t(y) = \begin{cases} b^t & , y = b^t \\ 0 & , y \neq b^t \end{cases} . \tag{10}$$

The second substring contains the integers that encode the rule information so that each integer I_r represents one MF in the space of the output variable and corresponds to the r^{th} rule.

The integrated chromosome corresponding to (7) and (9), similar to the one presented in [11] is depicted in Fig. 2.

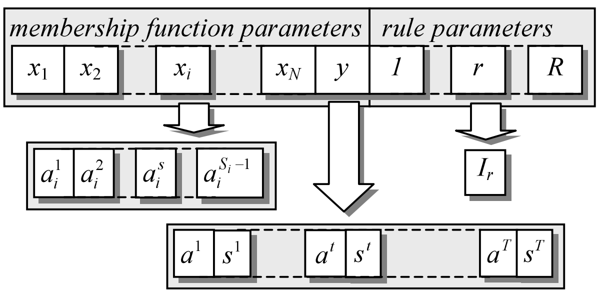

Fig. 2. Chromosome configuration.

Prior to converting to binary alphabet, the real number values of the encoded parameters p are scaled by applying

$$p'_{[p_{\min}, p_{\max}]} = (2^m - 1)\frac{p - p_{\min}}{p_{\max} - p_{\min}}, \tag{11}$$

where m denotes the precision of the coding. After each epoch (Fig. 1) and transforming to decimal alphabet, in turn, obtained values are rescaled using

$$p = p_{\min} + p' \frac{p_{\max} - p_{\min}}{2^m - 1} . \tag{12}$$

The fitness function used in this work is evaluated using

$$f(\varepsilon) = 100 - \frac{100\varepsilon}{y_{\max} - y_{\min}} . \tag{13}$$

6 Examples

Several functions were modeled, using GAs and other algorithms (gradient descent and ANFIS), for comparison purposes.

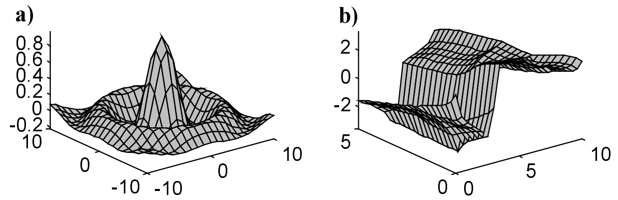

Fig. 3. a) "sombrero" function, b) fuzzy function

a) Approximation of "sombrero" function, a functional test for ANFIS [1] (Fig. 3a), (440 data pairs). This function was also modeled with unconstrained gradient descent, using equivalent definition of the fuzzy system (zeroth order prod-sum Sugeno system with triangular input MFs).

b) Approximation of fuzzy function used in [8] (Fig. 3b) (441 data pairs). Comparison results were obtained by ANFIS. The number of MFs for ANFIS was determined using the analogy between standard and Sugeno systems based on [12].

c) approximation of time series (Fig. 4), given by the Mackey-Glass differential delay equation.

$$\dot{x}(t) = \frac{0.2x(t-\tau)}{1 + x^{10}(t-\tau)} - 0.1x(t) . \tag{14}$$

We used the values $x(t - 12)$, $x(t - 6)$ and $x(t)$ to predict $x(t + 6)$. 500 samples were created between t = 118 … 617. Additional set of validation data was produced (t = 618…1117) to observe the generalization properties of those models. In obtaining the models all three methods were used

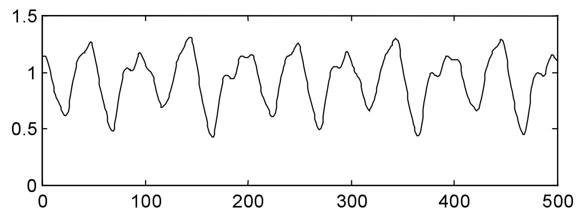

Fig. 4. Mackey-Glass time series.

The modeling results are shown in tables 1-3. N_p - number of trained parameters, N_t - number of training epochs.

Table 1. Modeling of "sombrero" function.

mod. alg.	[S_1 S_2]	*T*	*RMSE*	N_t	N_p
GA	[8 8]	7	0.0582	1000	87
GD	[8 8]	64	0.0143	1000	112

Table 2. Modeling of fuzzy function.

mod. alg.	[S_1 S_2]	*T*	*RMSE*	N_t	N_p
GA	[6 3]	5	0.3742	500	37
ANFIS	[5 2]	10	0.1643	500	51

Table 3. Modeling of Mackey-Glass time series.

mod. alg.	[S_1 S_2 S_3]	*T*	*trn. err.*	*chk. err.*	N_t	N_p
GA	[6 6 6]	6	0.0273	0.0299	500	240
ANFIS	[5 5 5]	125	0.0066	0.0095	20	545
GD	[6 6 6]	216	0.0102	0.0130	20	270

According to approximation error, ANFIS and gradient descent outperform GA. The comparison of training times generally also favors gradient-based algorithms. An interesting exception is the case of Mackey-Glass data, where it takes about 45 minutes to finish 20 of ANFIS epochs (CPU: P-II 350 MHz), while GA accomplishes its 500 epochs in 2 and half hours (CPU: AMD K-6 400 MHz). The resulting error of ANFIS is small enough, though, and further training brings no improvement. GA, in contrast, always requires substantial time to achieve the results of acceptable accuracy.

The gradient descent algorithm, implemented through Fuzzy Logic Toolbox 2.0 of MATLAB is not pure gradient descent because it is initialized with one-pass least squares estimation. To put the algorithms on the equal ground, we initialize the GA with the rule extraction algorithm of Wang and Mendel [13] and repeat the experiments (tables 4-6).

Table 4. "Sombrero" function approximation.

initial error(GA)	*Final error (GA)*	*initial error (GD)*
0.0758	0.0484	0.5071

Table 5. Fuzzy function approximation.

initial error(GA)	*Final err.(GA)*	*initial error (ANFIS)*
0.8274	0.2505	0.1407

Table 6. Modeling of Mackey-Glass time series.

initial error			*final error(GA)*	
ANFIS	*GD*	*GA*	*training*	*checking*
0.0179	0.0135	0.0534	0.0202	0.0211

7 Conclusions

Theoretically, GAs are capable to solve problems of arbitrary complexity. The schema theorem [9] which proves this is, however, based on several assumptions (infinite number of chromosomes and training epochs, to name two) that cannot be realized in practice. Typically, convergence is premature or too slow. Because GAs work with a number of potential models, each training epoch also requires a time to accomplish. The bottleneck of GAs when applied to fuzzy systems is the evaluation of fitness function. On the other hand, GAs are not so sensitive to the "curse of dimensionality" problem.

With GAs we are able to obtain transparent models of modeled functions. The universal nature of GAs allows to train fuzzy systems of arbitrary configuration. An interesting perspective is to employ GAs in combination with other techniques. For example, GA could be initialized with Wang-Mendel rule extraction algorithm, and employed for obtaining the optimal rulebase and sub-optimal definition of MFs. The result could be then fine-tuned with other algorithm.

8 References

[1] J.-S.R. Jang, "ANFIS: Adaptive-network-based fuzzy inference system". *IEEE Trans. System, Man, Cybern.*, vol. 23, No. 3, pp. 665-685, 1993.

[2] E.H. Mamdani, S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller". *Int. J. Man-Machine Studies*, vol. 7, pp. 1-13, 1975.

[3] T. Takagi, M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control". *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-15, No. 1, pp. 116-132, 1985.

[4] J.V. de Oliveira, "Semantic constraints for membership function optimization". *IEEE Trans. Systems, Man Cybern.*, Part A, vol. 29, No. 1, pp. 128 -138, 1999.

[5] M. Bikdash, "A highly interpretable form of Sugeno inference systems". *IEEE Trans. Fuzzy Systems*, vol. 7, No. 6, pp. 686 -696, 1999.

[6] Babuška, R. *Fuzzy Modeling and Identification*, Ph.D. dissertation, Technical University of Delft, 1997.

[7] Jager, R. *Fuzzy Logic in Control*, Ph.D. dissertation, Technical University of Delft, 1995.

[8] A. Riid, E. Rüstern, "Transparent fuzzy systems and modeling with transparency protection". In Proc. *IFAC Symp. on Artificial Intelligence in Real Time Control* (AIRTC 2000, Budapest, Oct 2-4, 2000), to be published.

[9] J.H. Holland. *Adaptation in Natural and Artificial Systems*, MIT Press, Ann Arbor, 1975.

[10] C.-T. Sun, "Rule-Base Structure Identification in an Adaptive-Network-Based Fuzzy Inference System". *IEEE Trans. Fuzzy Systems*, vol. 2, No. 1, pp. 64-73, 1994.

[11] G.V. Tan, X. Hu, "More on designing fuzzy controllers using genetic algorithms: guided constrained optimisation". In Proc. *6th IEEE Int. Conf. on Fuzzy Systems* (FUZZ-IEEE'97, Barcelona, July 1-5, 1997), vol. 1 pp. 497-502.

[12] R. Babuska, H.B. Verbruggen, "A new identification method for linguistic fuzzy models". In Proc. *4th IEEE Int. Conf. on Fuzzy Systems/2nd Int. Fuzzy Eng. Symp.* (FUZZ-IEEE/IFES'95, Yokohama, March 20-24, 1995), pp. 1207-1212.

[13] L.-X. Wang, J.M. Mendel, "Generating fuzzy rules by learning from examples". In Proc. *IEEE Int. Symp. on Intelligent Control* (ISIC'91, Arlington, August 13-15, 1991), pp. 263-268.