AUTOMATIC PROCESS FOR THE SYNTHESIS OF FUZZY SYSTEMS FROM INPUT-OUTPUT DATA

Carles Garriga-Berga
Enginyeria La Salle
Pg. Bonanova, 8
08022 Barcelona
cgarriga@salleURL.edu

Summary

In this paper we present a set heuristic criteria devised to address the problems encountered in designing a fuzzy system to fit a set of input-output data. In particular, we discriminate unsignificant linguistic variables, determine the number of fuzzy sets, place them in the universe of scope, propose a set of linguistic rules and give the number of bits necessary to represent each variable. The objective is to obtain in a simple and fast manner a good starting model to undergo further refinements.

Keywords: fuzzy identification and clustering.

1 INTRODUCTION

A common fuzzy system design method consists in fitting a set of input-output data pairs. A well known example of such methods is [10] where a fuzzy controller for the backer-upper truck is build out of a collection of samples generated by an expert driver.

Given a set of raw data, a number of problems arise during the design procedure. For instance, determining what are the relevant variables, how the the universe of scope should be partitioned or which is the best rule base.

There are different methods in the literature refering to fuzzy identification, most of them based in clustering techniques. Among the more popular ones, we can mention Fuzzy C-means [8] or Possibilistic C-means [5]. After clustering the transfer function, one can approximate the fuzzy sets needed to define the system. Some authors combine clustering techniques with genetic algorithms to obtain a fine tunning for the fuzzy sets [2]. In other methods ellipsoidal regions for the clusters are defined obtaining a good performance but at the price of increasing the computation cost [1], [3]. Alternative methods include taking clustering decision to obtain the minimum entropy [4].

In this paper we present a set heuristic criteria devised to adress the design problems. In particular, we are able to discriminate unsignificant linguistic variables, determine the number of fuzzy sets, place them in the universe of scope, propose a set of linguistic rules and give the number of bits necessary to represent each variable if there are enough samples (otherwise that would be only a necessary step of the algorithm). The goal is to obtain in a simple and fast manner a good starting model to undergo further refinements. These criteria can be set up in a three-step algorithm which first computes the relevance of the variables, then determines the number, position and shape of the necessary fuzzy sets to describe each relevant variable and finally computes a fuzzy rule base. The result is a simpler algorithm than the standard ones with minimum computational cost but still similar performance.

2 OUTLINE OF THE ALGORITHM

In this section the three basic steps necessary to implement the algorithm are given. In the first we show how to compute the relevance of each linguistic variable and give a first approximation of the necessary number of bits. The relevance is a real number between zero and one proportional to the importance of each variable and so the number of fuzzy sets. For instance if the relevance is 0.38 then we will need 4 fuzzy sets. The second one computes the placement of the fuzzy sets using the previously estimated number of fuzzy sets. The last one is a modified version of the Wang-Mendel algorithm that assigns the fuzzy set closer the desired output to the output fuzzy set in each rule.

2.1 RELEVANCE OF THE VARIABLES

The relevance of the variables is obtained by computing the mean of all the output samples that are close to 2^N points equally spaced in the universe of the scope, being N the number of bits. If some of these points do not have any output sample nearby, the number of bits used for this variable is decresed by one (necessary for the robustness of the algorithm). If a linguistic variable is weakly correlated with the output variable, all points have a similar mean value, so the difference between the largest and the smallest mean gives information of the relevance of each variable. These steps are summarised in the following pseudocode.

```
For each Input variable
   Optimal_partition is FALSE
   Numer_of_bits is a constant,
   While  Optimal_partition is FALSE
     Generate 2^(Number_of_bits) - 1 equidistant
     points (Pi) from min(Input) to max(Input)
     Compute the mean of the output samples
        (Oi) in the neighbourhood of each Pi
     If some Pi does not have output samples
        Decrease Number_of_bits by one
     else
        Optimal_partition is TRUE
   Relevance is (max(Oi)-min(Oi)) over
     (max(Output)-min(Output))
```

2.2 FUZZY SETS OF THE VARIABLES

The fuzzy sets are obtained by evaluating the intersection of $N-1$ equidistant lines with the function that interpolates between the mean values obtained in the previous block (recall N is the number of bits). The range of values of the universe of scope which deserves a fine accuracy will offer more intersections than the range of the universe of scope which requires less accuracy. Therefore, when projecting the intersection points to the universe of scope the edge points for the different fuzzy sets will be obtained, thus, getting a greater number of fuzzy sets in the range of the universe of the scope that requires a fine accurancy. The pseudocode version is :

```
For each Input variable
   Generate 2^(Number_of_bits) - 1 equidistant
      points (Pi) from min(Input) to max(Input)
   Compute the mean of the output samples (Oi)
      in the neighbourhood of each Pi
   Generate the function (Fi) interpolating Oi
   Generate Number_of_sets-2 equidistant lines
      (Li) from min(Output) to max(Output)
   Compute the intersections between Li and Fi
   Project the intersections to the universe of
      scope to obtain the edges of the sets
   Compute the mean of the closest points
      (less than the LSB)
```

2.3 LINGUISTIC RULES

This modified version of the Wang-Mendel's algorithm takes into account the fact that the output of a fuzzy systems does not only depend on the output fuzzy set of a single rule but also of the output fuzzy sets of the neighbouring rules. So for each possible rule we create an array of *number of output fuzzy sets* positions where each position stores the sum of the implication values of all input- output pairs if the output is near to the desired output. Then the output fuzzy set for each rule is the fuzzy set with the largest value in the array.

```
For each rule
   For each input-output pair
      Choose the output fuzzy set (Si)
         nearest to the desired output value
      Compute the implication (Ii)
      Accumulate Ii in the (Si) position
   Select the position with largest value
      as the output fuzzy set
```

3 RESULTS

The presented algorithm has been implemented and several proofs have been done to verify its correct functionality, some of which are presented in this section.

The first example is the non-linear equation

$$\exp(-y^3 + 0.5y) + \tanh(3x^5 - 0.5x) + n \tag{1}$$

which illustrates the benefits of the proposed algorithm. We have considered the 33% of 441 samples of the variables x and y. Variable n is non-biased gaussian noise. To prove the robustness of the algorithm, the samples of x and y have also been perturbed with additive zero-mean Gaussian noise whose variance is 1% of the output range.

First we compute the relevance values for the three input variables which are respectively 0.76, 0.36 and 0.02 (depending on the generated noise). From these numbers we initially assign 8 fuzzy sets to variable x, 4 fuzzy sets to variable y and eliminate variable n. The algorithm also considers 4 bits for variable x and 3 bits for variable y.

The second block of the algorithm keeps the same number of bits and gives the fuzzy sets that are shown in the figures 1 and 2. Two plots can be observed in these figures. The upper one shows the samples, the function Fi and the lines Li considered to obtain the fuzzy sets, while the bottom one plots the different fuzzy sets. It is interesting to see, looking at the first variable, that the algorithm can increase the number of fuzzy sets obtained in the previous block if the number of intersections is large. It is even able to merge proposed fuzzy sets that are too close.

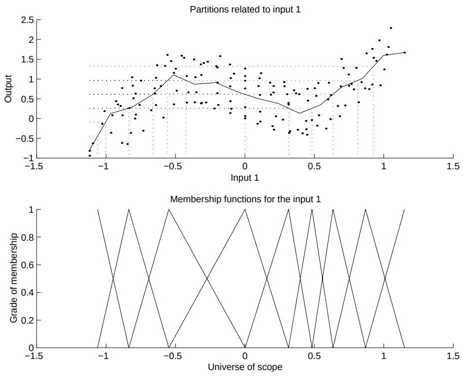

Figure 1: Fuzzy sets for the variable x

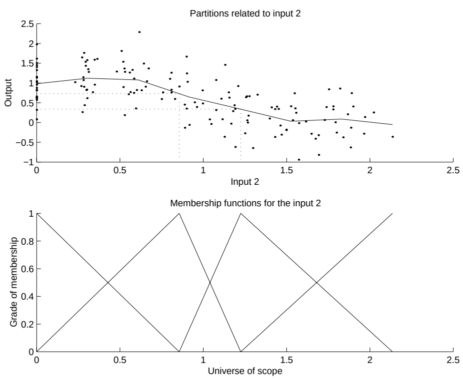

Figure 2: Fuzzy sets for the variable y

The last block gives the rule matrix for the fuzzy system written in table 1. The fuzzy sets are named by a number which is equal to the order of the fuzzy set in the universe of scope.

Table 1: Linguistic rules for the first example

Rule matrix									
$y \setminus x$	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
1	4	4	7	6	5	5	5	7	8
2	4	5	7	6	5	5	5	7	7
3	1	2	4	4	3	3	3	5	7
4	1	4	4	4	3	2	3	5	5

The real transfer function and the transfer function obtained with the fuzzy system are shown in figures 3 and 4, respectively.

A very good benchmark to evaluate the efficiency of an algorithm to identify using fuzzy systems consists in creating a fuzzy system and sample out some input-output pairs. These pairs are then used to feed the algorithm under test. The original fuzzy system should be recovered.

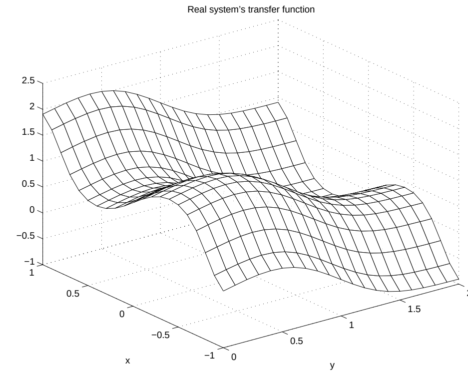

Figure 3: Transfer function of the real system

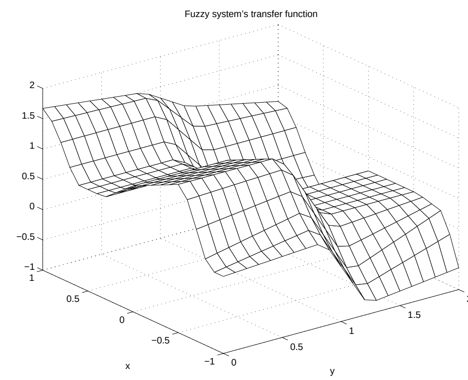

Figure 4: Transfer function of the fuzzy system

For this purpose, we define a highly non-linear fuzzy system with two input variables. Under the same conditions of the previous example but without noise, the algorithm obtains the membership functions of figure 5 where the dashed functions are the original functions and the continuous functions are the identified functions. The rule matrix for the fuzzy system is written in table 2 where the italic fuzzy sets are the original rules and the rest are the identified rules. As can be observed, the fuzzy sets are placed very close to the original ones and only 4 over 49 rules are different from the real ones.

Table 2: Linguistic rules for the second example

Rule matrix							
$x_1 \setminus x_2$	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	1/*1*	1/*1*	1/*1*	1/*1*	2/*2*	3/*3*	4/*4*
2	1/*1*	1/*1*	2/*2*	3/*3*	3/*3*	4/*4*	5/*5*
3	1/*1*	2/*2*	3/*3*	3/*4*	4/*4*	5/*5*	5/*6*
4	2/*2*	3/*3*	4/*4*	4/*4*	4/*4*	5/*5*	6/*6*
5	3/*2*	3/*3*	4/*4*	5/*4*	5/*5*	6/*6*	7/*7*
6	3/*3*	4/*4*	5/*5*	5/*5*	6/*6*	7/*7*	7/*7*
7	4/*4*	5/*5*	6/*6*	7/*7*	7/*7*	7/*7*	7/*7*

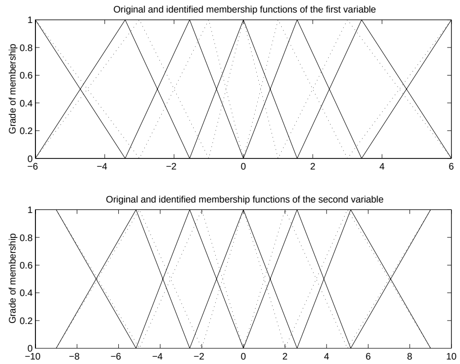

Figure 5: Fuzzy sets for the second example

Other systems have been identified and some of them can be found in the literature. For example in [9], many mathematical equations are identified from 2000 input-data pairs with zero-mean gaussian noise whose variance is a 5% of the output range. Using less samples, i.e. 100 input-data pairs with the same amount of noise, we obtain similar results. For example, in the first function being identified their algorithm presents a RMSE=0.07 while our algorithm offers a RMSE of 0.13. Therefore, we achieve similar results with a poor sampling rate and a simplest algorithm. Other comparisons are similar to this one.

4 DISCUSSION

A simple algorithm that is able to give a first-approach of the fuzzy systems and that relates input- output pairs has been presented and discussed. Its main features are that it is quite simple and that the parameters may be adjusted by the designer: the significant variables, the number and placement of the fuzzy sets, the number of bits and the linguistic rules. This is obviously a first-approach and the resulting system should be fine tuned using other, more powerful tecniques. We suggest the gradient descent algorithm [7] because if we have a good approach of the result, this algorithm converges fast to the optimal solution.

Some refinements of the presented algorithm can also be performed. For instance, the number of bits may not be fixed during the execution of the algorithm because the second block adjusts the number of bits when evaluating the fuzzy sets. In addition, some versions of the algorithm have been build including the *fuzzy-curves* algorithm presented in [6] to increase its robustness and some other versions provide an analytical module to decide wheter the linguistic rules are not coherent in order to compute again the fuzzy sets. The whole algorithm will be optimized and presented in a forthcoming work.

In spite of the surpresing results that this and other identification algorithms can offer, the most important thing one must bear in mind when trying to identify a system, is the necessity of a good set of samples of all the variables because they will be at the end the responsible of the result.

Acknowledgements

Author would like to thank Dr. Xavier Vilasís and Dr. Ferran Mazzanti for their help on writting this work.

References

[1] S. Abe and R. Thawonmas. A fuzzy classifier with ellipsoidal regions. *IEEE Trans. on fuzzy systems* vol. 5, n 3, pp. 358-368, 1997

[2] M. Delgado and F. Gǿmez-Skarmeta and F. Martín. A fuzzy clustering-based rapid prototyping for fuzzy rule-based modeling. *IEEE Trans. on fuzzy systems* vol. 5, n 2, pp. 223-232, 1997

[3] J. Dickerson and B. Kosko. Fuzzy function approximation with ellipsoidal rules. *IEEE Trans. on syst., man and cybern.* vol. 26, n 4, pp. 542-560, 1996

[4] C.J. Kim. An algorithmic approach for fuzzy inference. *IEEE Trans. on fuzzy systems* vol. 5, n 4, pp. 585-598, 1997

[5] R. Krishnapuram and J.M. Keller. The possibilistic c-means algorithm: insights and recommendations. *IEEE Trans. on fuzzy systems* vol. 4, n 3, pp. 385-395, 1996

[6] Y. Lin and G.A. Cunningham III. A new approach to fuzzy-neural system modeling. *IEEE Trans. on fuzzy systems* vol. 3, n 2, pp. 190-197, 1995

[7] H. Nomura and I. Hayashi and N. Wakami. A self-tuning method of fuzzy control by descent method. *Proceedings of the IFSA'91* Brussels, pp. 155-158, 1991

[8] N.R. Pal and J.C. Bezdek. On cluster validity for the fuzzy c-means model. *IEEE Trans. on fuzzy systems* vol. 3, n 3, pp. 370-379, 1995

[9] R. Rovatti and R. Guerreri. Fuzzy sets of rules for system identification. *IEEE Trans. on fuzzy systems* vol. 4, n 2, pp. 89-102, 1996

[10] L.X. Wang and J.M. Mendel. Generating fuzzy rules by learning from examples. *IEEE Trans. on syst., man and cybern.* vol. 22, n 6, pp. 1414-1427, 1992