Optimization of Fuzzy Systems by Evolutionary Algorithms

Camille Messien

***Abstract*— Fuzzy logic is a mathematical theory that deals with approximation and handles the concept of partial truth. Reasoning systems based on this theory exhibit both good numeric performance (precision) and linguistic representation (interpretability).**

However, designing a fuzzy system is an arduous task, that requires the identification of many parameters. To solve this issue, a genetic algorithm has been implemented in this paper in order to design efficient fuzzy systems. This optimization process is then used on the Wisconsin breast cancer diagnosis (WBCD) problem. This method was successful in designing fuzzy systems with low number of rules and performance greater than 93%.

I. INTRODUCTION

Introduced in 1965 by Lofti Zadeh [1], fuzzy logic is a many-valued logic in which the truth values of variables can be any real number between 0 and 1. Inspired by the processes of human perception and cognition, fuzzy logic uses linguistic variables instead of crisp values, and is thus able to handle the concept of partial truth. Systems based on this logic are called fuzzy systems and can be used in many applications: fuzzy controllers [2,3], diagnosis systems [4], classification [5], etc.

The performance of a fuzzy system depends on various parameters like the fuzzy rules used or the membership functions that map crisp inputs into linguistic variables. Designing these parameters is often a complex task, that requires either an expert in the field in which the fuzzy system is used or an efficient optimization process. In prior work, evolutionary algorithms have proven to be successful in designing effective fuzzy systems [4,5,6]. Thus an implementation of evolutionary fuzzy modeling is presented here, and has then be used to solve the Wisconsin breast cancer diagnosis (WBCD) problem.

This paper is organized as follows. In Section II, a general description of fuzzy systems is presented. The optimization process through a basic evolutionary algorithm is then explained in Section III. In Section IV we will discuss the implementation used to solve the Wisconsin breast cancer diagnosis (WBCD) problem and the results obtained. In Section V, we draw some conclusions.

II. FUZZY SYSTEMS

Fuzzy logic is a mathematical theory that deals with approximate reasoning. By introducing the notion of degree in the verification of a condition, fuzzy logic provides flexibility and the possibility to take into account inaccuracies and uncertainties. Fuzzy systems are rule-based systems that use fuzzy logic to reason about data. The following sections presents the main concepts of fuzzy logic and fuzzy systems.

C. Messien is a graduate student in the Department of Aeronautics and Astronautics, Stanford University.

A. Fuzzy sets and fuzzy logic

As for ordinary, or *crisp*, set, a fuzzy set A in a universe of discourse U can be defined by conditions to identify the elements $x \in A$. These conditions are characterized by a membership function $\mu_A(x) : U \to [0,1]$, that denotes the degree to which x is a member of A. We call *fuzzification* the operation that assigns a membership value $\mu(x)$ to an input value x. Common membership functions are triangular, trapezoidal, or bell-shaped, which produces a simple visual representation of the fuzzy set they describe.

Using these membership functions, we can define the main operators of the fuzzy logic. As for the boolean logic, the AND, OR and NOT operators can be defined for fuzzy logic. These operators are extensions of the boolean operators and should reduce to the commonly known operators if the fuzzy set is reduced to a crisp set. There exist several possible operators for AND and OR, and we will use minimum and maximum operators in this paper.

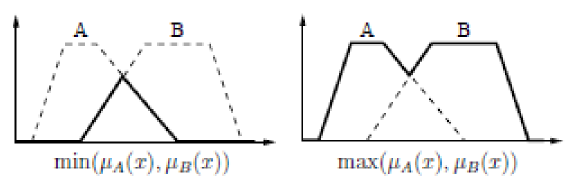

Fig. 1. AND (left) and OR (right) operators.

The NOT operator is called *fuzzy complement* operator and is universally used in fuzzy systems. It is defined by $1 - \mu_A(x)$.

Fuzzy sets are used to represent *linguistic variables* like *Low*, *High*, *Cold*. Several membership functions can be applied on a crisp value. For example, a temperature of $25°C$ could have the membership values $\mu_{Warm} = 0.8$ and $\mu_{Hot} = 0.2$, meaning that $25°C$ belongs to the sets *Warm* and *Hot*. We can then easily defined some *fuzzy conditions* like "Temperature is Warm", which then allow us to make meaningful reasoning, using *fuzzy rules*.

In a fuzzy inference system, the fuzzy rules take the form: **if** (*input fuzzy condition*) **then** (*output fuzzy assignment*). The input condition is the *antecedent* of the rule, while the output assignment is the *consequent*. A fuzzy inference system may contains several rules, and all rules are evaluated during a reasoning. As a crisp input can belong to several fuzzy sets, several rules can apply, with different truth level and different output fuzzy set.

We are expected to take a single decision, thus an *aggregation* operation is used to obtain a single fuzzy set: generally the `OR` operator is used to create this single fuzzy set. Then a *defuzzification* process is used to obtain a crisp output. The two commonly used methods for the *defuzzification* process are the *Center of Areas* (COA), also called centroid, and the *Mean of Maxima* (MOM). These methods are illustrated below.

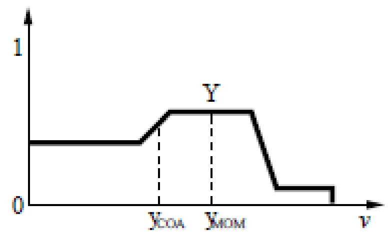

Fig. 2. Defuzzification

B. Fuzzy inference systems

A fuzzy system is composed of several elements: (1) a fuzzifier, that converts some crisp input into fuzzy values; (2) an inference engine that applies the fuzzy reasoning mechanism; (3) a defuzzifier, that converts the fuzzy output into a crisp value; (4) a knowledge base, that contains the definitions of the rules, the membership functions, the fuzzy operators...

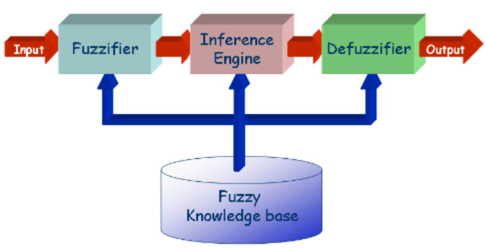

Fig. 3. Basic structure of a fuzzy system

There exists three types of fuzzy systems: Mamdani, TSK and Signleton fuzzy system. They differ by the consequents of their rules. Mamdani fuzzy system uses the definitions of consequents of previous parts, while TSK and Singleton fuzzy systems replace the fuzzy consequent by, respectively, function of the input variables and singletons.

Parameters of a fuzzy system can be classified into four categories [7]:

- The logical parameters contain the type of the membership functions, the fuzzy logic operators and the defuzzyfication method.
- The structural parameters include the number of membership functions per linguistic variables, the number of rules of the system and the number of antecedents of the rules.
- The connective parameters define the antecedents, the consequents and the weights of the rules.
- The operational parameters characterize the values of the membership functions.

III. Genetic Algorithms

Evolutionary algorithms are nature-inspired algorithms that can take use of Darwinian principles to solve problems. Imitating the process of natural selection on a specific population, they potentially produce progressively better solutions to a problem by using basic mechanisms such as selection, mutation or crossover. An objective function called the fitness then determined the performance of each individual of a population.

A simple genetic algorithm using binary numbers can be described by the following steps:

- Step 1: Creation of an initial random population of size n.
- Step 2: Evaluation of the fitness of the current population.
- Step 3: Selection of k individuals of the population to create a mating pool. The selection process used in this paper is the *Roulette wheel selection*. A scaling scheme is first applied on the fitness of each individual of the population, to obtain a scaled fitness $f'_i \in [0,1]$. The sum $S = \sum_{i=1}^{n} f'_i$ is then computed, and a random number $r \in [0,1]$ is generated. If j is the index such that $f'_1 + \ldots + f'_{j-1} \leq rS \leq f'_1 + \ldots + f'_j$, then the jth member is copied into the mating pool. This process is then repeated k times. With this method, the chance of being selected is proportional to the scaled fitness value.
- Step 4: Crossover operation to create n children. Two parents are randomly selected, and a random part of their DNA is exchanged to create two children.
- Step 5: Mutation of the new population. Every bit of the individual is visited and can be modified with a probability b_p. In this paper, b_p was equal to 0.05.

We can then go back to the second step until the desired number of generations or the desired performance is reached.

IV. Application: The Wisconsin breast cancer diagnosis (WBCD) problem

In this part, a fuzzy system for the diagnosis of breast cancer has been designed using the Wisconsin breast cancer diagnosis (WBCD) database [8,9]. This dataset is the result of the efforts made at the University of Wisconsin Hospital for the diagnosis of breast tumors based solely on fine-needle aspiration biopsy (FNA). This test involves fluid extraction from a breast mass using a small-gauge needle. The fluid is then inspected under a microscope.

Nine characteristics of an FNA sample are considered relevant for diagnosis of breast cancer:

- Clump thickness, v_1;
- Uniformity of cell size, v_2;
- Uniformity of cell shape,v_3;
- Marginal adhesion, v_4;
- Single epithelial cell size, v_5;
- Bare nuclei, v_6;
- Bland chromatin, v_7;

- Normal nucleoli, v_8;
- Mitosis, v_9.

For the diagnosis, each characteristic is assigned an integer value between 1 and 10. The WBCD database contains 699 cases with their diagnosis. 16 of these entries had an unknown value for one of the above characteristics and have thus been removed from the database used here.

Case	v_1	v_2	...	v_8	v_9	Diagnosis
1	5	1	...	1	1	benign
2	5	4	...	2	1	benign
.	.	.	.	.	.	.
682	4	8	...	6	1	malignant
683	4	8	...	4	1	malignant

Fig. 4. The WBCD database

The fuzzy system designed here uses the nine characteristics of FNA samples as inputs and then computes a continuous appraisal value of the malignancy of a case. A threshold then outputs a benign or malignant diagnosis using the fuzzy system's output. In the WBCD database, a benign diagnosis has a value of 2, while a malignant one has a value of 4. A fuzzy system's output less than 3 is thus classified as a benign diagnosis by the threshold, and a value over 3 would then be a malignant diagnosis. The distance between the fuzzy system's output and the value of 2 or 4 represents the confidence measure of the diagnosis.

A. Fuzzy system parameters

Our designed fuzzy system must possess two characteristics to become a good computerized diagnosis tool: it must attain a high performance while being easily interpretable. Those two characteristics are often in conflict. A high performance system should give a correct diagnosis for most of the cases, while an interpretable system should give a concrete explanation of how the diagnosis has been made and a confidence measure of the diagnosis. A fuzzy system with a large number of complex rules and complex membership functions could attain a high performance but would result in a low interpretability.

Several studies have been based on the WBCD database. Using linear programming techniques, Bennet and Mangasarian [10] obtained a 99.6% classification rate on 487 cases (the reduced database available at the time). However their diagnosis decisions are black boxes, with no explanation of the diagnosis. I have thus decided to take into account the interpretability criteria in the design of my fuzzy system. My final fuzzy system should thus contained a small number of rules, with a small number of variables per rules. Below is defined the fuzzy-system setup:

- *Logical parameters:* mamdani-type fuzzy systems; min-max fuzzy operators; trapezoidal input and ouput membership functions; centroid defuzzification method.
- *Structural parameters:* two membership functions per inputs and output, *Low* and *High* for the inputs, *benign* and *malignant* for the output; number of rules fixed by the user.
- *Connective parameters:* the antecedents of the rules for the *benign* diagnosis are to be found by the evolutionary algorithm. As at least one rule needs to be used for any set of inputs, a default-rule with *malignant* diagnosis has been implemented. As Matlab Fuzzy Logic toolbox does not have a default rule, two basic rules with the same weight $w \in [0;1]$, the same unique antecedent $v_i, i \in \{1,...,9\}$ (but with opposite value) and the consequent *malignant* are used. The weight and the chosen antecedent are to be found by the evolutionary algorithm.
- *Operational parameters:* the input membership-function values are searched by the evolutionary algorithm. The output membership values are fixed at $P = 2.7$ and $d = 0.6$.

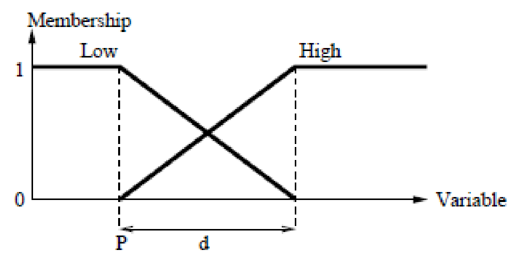

Fig. 5. Input membership functions, defined by P and d, the start point and the length of membership function edges, respectively.

B. The evolutionary setup

The evolutionary algorithm has to search for several fuzzy-system parameters: input membership functions variables P and d, antecedents of rules, and weight of default rules. A Pittsburgh-like approach has been used, with a simple genetic algorithm: each individual of a population represents an entire fuzzy-system, and the evolutionary algorithm maintains a population of candidate fuzzy systems. Genome of individuals encodes three parameters:

- Membership-function parameters. Two parameters P and d are defined for each inputs ($v_1 - v_9$) and represent the characteristics of the input membership-functions. P and d can take integer values between $\{1,...,8\}$.
- Antecedents. A *benign* rule has the form:
  if (v_1 **is** A_1^i) **and ... and** (v_9 **is** A_9^i) **then** (*out* **is** *benign*)
  where A_j^i represents the linguistic variable applicable to v_j. Values of A_j^i can be: 1 (*Low*), 2 (*High*), or 0 or 3 (*Don't Care*). A *Don't Care* value means that the variable v_j won't be used in the rule. If the evolutionary algorithm gives *Don't Care* value for all antecedents of a rule, then this rule is removed from the set of rules.
- Default rules parameters. The weight $w \in \{0.1,...,1.0\}$ and the index $k \in \{1,...,9\}$ of the appropriate antecedent variable are chosen by the algorithm.

The evolution has been made with a fixed population size of 100 individuals and terminates when the maximum number of generations, G_max, is reached. I used $G_max = 20$

Parameter	Values	Bits	Qty	Total bits
P	$\{1,...,8\}$	3	9	27
d	$\{1,...,8\}$	3	9	27
A	$\{0,1,2,3\}$	2	$9N_r$	$18N_r$
w	$\{0.1,...,1.0\}$	4	1	4
k	$\{1,...,9\}$	4	1	4

Fig. 6. Parameter encoding of an individual's genome. Total genome length is $62+18N_r$, where N_r is the number of rules fixed by the designer.

in my experiment, in order to keep a low computational time, but this value could be increased if a higher performance is required. 25 individuals were selected during the creation of the mating pool.

The performance of a fuzzy system is computed according to three criteria: the classification performance F_c, which is the percentage of cases correctly classified; the average quadratic error F_e between the fuzzy system output and the correct discrete diagnosis (either 2 or 4); and the average number of variables per active rule F_v. The fitness function of a system is then given by $F = F_c - \alpha F_v - \beta F_e$, where α and β are small empirical values. Here I fixed $\alpha = 0.05$ and $\beta = 0.01$.

C. Results

The number of *benign* rules were fixed between one and five. The genetic algorithm was trained on the set containing all 683 cases of the WBCD database: a future improvement of this implementation could be defining a test set and a training set, to avoid overfitting. The fuzzy systems found had the following performances:

N_r	1	2	3	4	5
F_c	0.9531	0.9502	0.9370	0.9458	0.9356

Fig. 7. Performance of the fuzzy systems. N_r is the number of rules fixed by the designer and F_c is the percentage of cases correctly classified.

As we can see, a larger number of rules may not result in a higher performance. It would be interesting to perform a large number of evolutionary runs to find out the evolution of the mean performance with the number of *benign* rules and with the maximum number of generations. However, due to the computational cost of evaluating numerous fuzzy systems, this hasn't been implemented in the current paper.

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9
P		4	1			1			
d		1	5			6			

	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9
R_1		1				1			
R_d			1						

Fig. 8. Parameters of the fuzzy system with one rule. R_d represents one of the default rule, and has a weight $w = 0.3$ here.

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9
P	6		8	1	1	7		5	3
d	1		6	3	2	5		6	7

	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9
R_1				2	1				
R_2	1		1			1		1	1
R_d						1			

Fig. 9. Parameters of the fuzzy system with two rules. R_d represents one of the default rule, and has a weight $w = 0.7$ here.

The fuzzy systems found with one and two *benign* rules are shown above. P and d are the parameters of the membership functions. When a variable is not used in any rule, these parameters are not relevant and thus have not been written in the following tables. The A_i are the antecedents of the rules: $A_i = 1$ means that v_i is *Low*, while $A_i = 2$ means that v_i is *High*.

Thus, when optimized using a genetic algorithm, our fuzzy systems are able to reach a great performance in a reasonable number of generations.

V. CONCLUSIONS

In this paper, an optimization process using genetic algorithm has been implemented in order to design high performance fuzzy systems. This process has then been used on the Wisconsin breast cancer diagnosis (WBCD) database. Using a population of 100 individuals and 20 generations, we designed fuzzy systems of one to five *benign* rules with performances higher than 93%.

Future work could involve computing the mean performance of the fuzzy systems by performing a large number of evolutionary runs. Adding a test set to the genetic algorithm could also be interesting.

REFERENCES

[1] L.A. Zadeh. Fuzzy sets. Information and Control, 8:338-353, 1965.

[2] P. Singhala, D.N Shah and B. Patel. Temperature Control using Fuzzy Logic. IJICS, Volume 4, Number 1, January 2014.

[3] J. Y. Zhang and Y. Li. Application of Genetic Algorithm in Optimization of Fuzzy Control Rules. IEEE, 2006.

[4] Z. Jing, G. Qiang, H. ZhiGang and H. ZhaoTing. Optimization of Fuzzy Rules by Multi-objective Genetic Algorithm in Avionic Fault Diagnosis System. IEEE, 2009.

[5] G. Saggio, P. Cavallo *et al.*. Comparison of Tzo Different Classifiers for Mental Tasks-Based Brain-Computer Interface: MLP Neural Networks vs. Fuzzy Logic. IEEE, 2009.

[6] P.A.D. Castro and H.A. Camargo. Learning and Optimization of Fuzzy Rule base by means of Self-Adaptive Genetic Algorithm. IEEE, 2004.

[7] C.A. Peña Reyes. Coevolutionary Fuzzy Modeling. Lausanne, EPFL, 2002.

[8] O.L. Mangasarian and W.H. Wolberg. Cancer diagnosis via linear programming. SIAM News, Volume 23, Number 5, September 1990, pp 1 & 18.

[9] William H. Wolberg and O.L. Mangasarian. Multisurface method of pattern separation for medical diagnosis applied to breast cytology. Proceedings of the National Academy of Sciences, U.S.A., Volume 87, December 1990, pp 9193-9196.

[10] K.P. Bennett and O.L. Mangasarian. Neural network training via linear programming. Elsevier Science, Advances in Optimization and Parallel Computing, 1992, pp 56-57.