Evolving Fuzzy Rules for Breast Cancer Diagnosis

Carlos Andrés Peña-Reyes† and Moshe Sipper†

†*Logic Systems Laboratory, Swiss Federal Institute of Technology, CH-1015 Lausanne, Switzerland.*
E-mail: carlos.andres@di.epfl.ch, moshe.sipper@di.epfl.ch. Web: lslwww.epfl.ch.

Abstract— We present an evolutionary approach for discovering fuzzy systems for breast cancer diagnosis. By judiciously designing an appropriate representation scheme (genome) and fitness function, the genetic algorithm is then able to produce successful systems. These surpass the best known systems to date in terms of combined performance and simplicity.

I. Introduction

Fuzzy logic is a computational paradigm that provides a mathematical tool for dealing with the uncertainty and the imprecision typical of human reasoning [1]. A prime characteristic of fuzzy logic is its capability of expressing knowledge in a linguistic way, allowing a system to be described by simple, "human-friendly" rules. A *fuzzy inference system* is a rule-based system that uses fuzzy logic, rather than boolean logic, to reason about data [1]. Its basic structure comprises four main components: (1) a fuzzifier, which translates crisp (real-valued) inputs into fuzzy values, (2) an inference engine that applies a fuzzy reasoning mechanism to obtain a fuzzy output, (3) a defuzzifier, which translates this latter into a crisp value, and (4) a knowledge base, which contains both an ensemble of fuzzy rules, known as the rule base, and a database, which defines the membership functions used in fuzzy logic.

Fuzzy modeling is the task of identifying the parameters of a fuzzy inference system so that a desired behavior is attained. There are several works on fuzzy modeling, based on neural networks [2,3], genetic algorithms [4–6], and hybrid methods [7], which automate some stages of the process. One of the more important problems in fuzzy modeling is the *curse of dimensionality*, meaning that the computation requirements grow exponentially with the number of variables. Selection of important variables and adequate rules is critical for obtaining a good model.

The parameters of fuzzy inference systems can be classified into four categories (Table 1): logic, structural, connection, and operational. Generally speaking, this order also represents their relative influence on system behavior (with logic being the most influential and operational the least).

Usually, in fuzzy modeling logic parameters are predefined by the designer. Structural, connection, and operational parameters may be either predefined, or obtained by synthesis or search methodologies. Generally, the search space, and thus the computational effort, grows exponentially with the number of parameters. Thus, one can either invest more resources in the chosen search methodology, or infuse more a priori, expert knowledge into the system (thereby effectively reducing the search space).

Table 1 Parameters of fuzzy inference systems

Class	Parameters
Logic	Reasoning mechanism Fuzzy operators Membership function types Defuzzification method
Structural	Relevant variables Number of membership functions Number of rules
Connection	Antecedents of rules Consequents of rules Rule weights
Operational	Membership function values

This paper presents a genetic-algorithm strategy for discovering fuzzy systems for breast cancer diagnosis, based on the Wisconsin Breast Cancer Diagnosis (WBCD) database. The problem (henceforth denoted WBCD) involves classifying a presented case as to whether it is benign or malignant. It admits a relatively high number of variables and consequently a large search space. Our encoding of solutions (the genome) takes advantage of previous knowledge about the problem, thus reducing the search space while favoring the extraction of the most significant variables in order to provide more human-comprehensible rules. Referring to Table 1, the evolved parts of the fuzzy system in this work are: the relevant variables, the antecedents and consequents of rules, and the values of input membership functions. Thus, we evolve structural, connection, and operational parameters at the same time.

II. The Breast Cancer Diagnosis Problem

Breast cancer is a common disease and a frequent cause of death in women in the 35-55 year age group. The presence of a breast mass[1] is an alert sign, but it does not always indicate a malignant cancer. Fine needle aspiration[2] of breast masses is a mostly non-invasive diagnostic test that obtains information

[1]Most breast cancers are detected as a lump or mass on the breast, either directly by self-examination, by mammography, or by both [8].

[2]Fine needle aspiration is an outpatient procedure that involves using a small-gauge needle to extract fluid directly from a breast mass [8].

needed to evaluate malignancy. In order to assist medical professionals in diagnostic based on microscopic examination of fine needle aspirates, a computational tool is currently used at the University of Wisconsin Hospital. The WBCD database [9] consists of nine measures obtained from fine needle aspirates, each of which is ultimately represented as an integer value between 1 and 10. The measured variables are as follows: (1) Clump Thickness (v_1); (2) Uniformity of Cell Size (v_2); (3) Uniformity of Cell Shape (v_3); (4) Marginal Adhesion (v_4); (5) Single Epithelial Cell Size (v_5); (6) Bare Nuclei (v_6); (7) Bland Chromatin (v_7); (8) Normal Nucleoli (v_8); and (9) Mitosis (v_9).

The diagnostics in the WBCD database were furnished by specialists in the field. The database itself consists of 683 cases, with each entry representing the classification for a certain ensemble of measured values:

case	v_1	v_2	v_3	...	v_9	*diagnostic*
1	5	1	1	...	1	*Benign*
2	5	4	4	...	1	*Benign*
:	:	:	:		:	:
683	4	8	8	...	1	*Malignant*

Note that the diagnostics do not provide any information about the degree of benignity or malignancy. There are several studies based on this database, usually with data divided into two sets: training and test. The training set is used for system synthesis (i.e., finding good parameters) and the test set is used for verification purposes. Bennet and Mangasarian [10] used linear programming techniques, obtaining 100% classification on the training set and 98.3% on the test set. However, their solution exhibits little understandability, i.e., diagnostic decisions are essentially black boxes, with no explanation as to how they were attained. Kermani *et al.* [11] used a genetic algorithm to extract the most important variables, their attained performance level being lower (94.7% on all cases, no training/test data was given). Setiono [12] proposed a method based on pruned neural networks for finding a set of rules to explain the diagnostic. His results are encouraging, exhibiting both good performance, and a reduced number of rules and relevant input variables. However, the extraction of rules is a manual, experience-based process.

III. The Experimental Set-up

This section focuses on the two-component set-up we used in order to evolve fuzzy rules for the WBCD problem: (1) the fuzzy inference system itself, and (2) the genetic algorithm.

A. Fuzzy system parameters

Previous knowledge about the WBCD problem represents valuable information to be used for our choice of fuzzy parameters. Following Table 1, we delineate below the fuzzy system set-up:

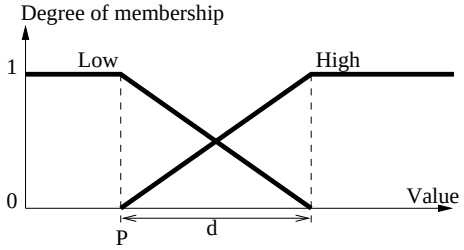

Figure 1 Orthogonal membership functions and their parameters, plotted above as degree of membership versus input values. The orthogonality condition means that the sum of all membership functions at any point is one. P and d define the start point and the length of membership function edges, respectively (as shown above).

1. *Logic parameters*
   - Reasoning mechanism: zero-order, Takagi-Sugeno-Kang (TSK) fuzzy system, meaning that consequents of rules (i.e., output membership functions) are real values (also called singletons), rather than fuzzy ones.
   - Fuzzy operators: min and max.
   - Input membership function type: orthogonal, trapezoidal (see Figure 1).
   - Defuzzification method: weighted average.
2. *Structural parameters*
   - Relevant variables: there is insufficient a priori knowledge to define them, therefore this will be one of the genetic algorithm's goals.
   - Number of membership functions: two membership functions, denoted *Low* and *High* are used for the input variables (see Figure 1). We also experimented with three membership functions but the results were less satisfactory, probably due in part to the increased search space size. Two output values are used, corresponding to *Benign* and *Malignant* diagnostics.
   - Number of rules: results from Setiono [12] show that few rules are needed to achieve good performance. Thus, we limited the number of rules to be in the range [1,4]. These rules are evolved.
3. *Connection parameters*
   - Antecedents of rules: to be found by evolution.
   - Consequent of rules: the implemented strategy has the algorithm find rules for one of the possible consequents (malignant or benign), the other being an `else` condition.
   - Rule weights: active rules have a weight of value 1, and the `else` condition has a weight of 0.1.
4. *Operational parameters*
   - Input membership function values: to be found by evolution.
   - Output membership function values: following the WBCD database, we used a value of 2 for *Benign* and 4 for *Malignant*.

B. The genetic algorithm

As noted, we use a genetic algorithm to search for four parameters: relevant variables, antecedents and consequents of rules, and input membership function values (Table 1).

- Relevant variables are searched for implicitly by letting the algorithm choose non-existent membership functions as valid antecedents; in such a case the respective variable is considered irrelevant.
- Membership function parameters: there are nine variables, each with two parameters P and d which define, respectively, the start point and the length of the membership function edges (Figure 1).
- Antecedents and consequents: the i-th rule has the form:

  if (v_1 **is** A_1^i) **and ... and** (v_9 **is** A_9^i) **then** (*output* **is** C^i)

  where A_j^i represents the membership function applicable to variable v_j. A_j^i can take on the values: 1 (*Low*), 2 (*High*), or 0 or 3 (v_j is not used by rule). C^i can take on the values: 1 (*Benign*) or 2 (*Malignant*).

Table 2 delineates the parameters encoding, which together comprise one individual's genome.

Table 2 Parameters encoding of an individual's genome. Total genome length is $64 + 18N_r$, where N_r denotes the number of rules.

Parameter	Values	Bits	Qty	Total bits
P	[1-10]	4	9	36
d	[0-7]	3	9	27
A	[0-3]	2	$9N_r$	$18N_r$
C	(1,2)	1	1	1

To evolve the fuzzy inference system, we used a simple genetic algorithm, with a fixed population size of 200 individuals, no generational overlap, and fitness-proportionate selection. As for the fitness function of the genetic algorithm, the classification performance of an individual was tested with a training set of 342 cases extracted from the database. The fitness of an individual is the ratio of *correct diagnostics* to *size of training set*. The algorithm terminates when the maximum number of generations is reached, or when the increase in the performance of the best individual over five successive generations falls below a certain threshold (in our experiments 10^{-4}). The test set we used for verifying the performance of the evolved fuzzy inference system contains the remaining 341 individuals.

IV. Results

This section describes our results, starting with general ones concerning the genetic algorithm, followed by two illustrative examples of evolved fuzzy systems.

A. The genetic algorithm...

We performed 40 evolutionary runs, all of which ended with high-performance systems: considering the best individual per run, 39 runs led to a fuzzy system whose performance exceeds 96%, and 9 runs ended with performance exceeding 97%.

B. ...and the fuzzy systems it discovered

We next describe two of our top-performance systems, demonstrating the interesting solutions that the genetic algorithm was able to discover. The first system consists of two rules (note that the `else` condition is not counted as an active rule):

if (v_2 **is** *Low*) **and** (v_4 **is** *Low*) **and** (v_6 **is** *Low*) **and** (v_8 **is** *Low*) **and** (v_9 **is** *Low*) **then** (*output* **is** *Benign*)

if (v_1 **is** *High*) **and** (v_2 **is** *High*) **and** (v_3 **is** *High*) **and** (v_4 **is** *High*) **and** (v_5 **is** *High*) **and** (v_6 **is** *Low*) **and** (v_7 **is** *High*) **and** (v_8 **is** *Low*) **and** (v_9 **is** *High*) **then** (*output* **is** *Benign*)

else (*output* **is** *Malignant*)

The evolved parameters P and d (Figure 1) are as follows:

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9
P	7.6	2.8	2.2	2.8	7.0	5.2	2.8	2.2	7.6
d	2	2	2	7	3	1	2	6	7

Table 3 summarizes the performance of our two-rule fuzzy system, and compares it with a three-rule Boolean system reported by Setiono [12]. The latter was obtained by manually extracting the rules from a pruned neural network. Interestingly, when applying an automatic verification tool, we found that the second rule of our system is never actually activated by any of the cases, both in the training and test sets. Thus, our evolved single-rule system compares very favorably with the more complex three-rule one.

Table 3 Performance values of our evolved two-rule fuzzy system and Setiono's three-rule Boolean system, manually extracted from a pruned neural network. Performance level is given for both training and test sets. Also shown are separate performance values for the benign and malignant cases, as well as the overall average of the two.

	Our system		Setiono's	
	Training	Test	Training	Test
Benign	98.7%	97.8%	97.4%	96.5%
Malignant	95.9%	91.5%	98.4%	96.7%
Overall	97.7%	95.6%	97.7%	96.6%

Can the genetic algorithm automatically discover a simple, single-rule system, i.e., without recourse to any

post-processing? Our results have shown that this is indeed the case. One such single-rule solution, with but three relevant variables, is delineated below:

if (v_2 **is** *Low*) **and** (v_6 **is** *Low*) **and** (v_8 **is** *Low*) **then** (*output* **is** *Benign*)

else (*output* **is** *Malignant*)

where the evolved P and d parameters are as follows:

	v_2	v_6	v_8
P	2.8	1.0	5.8
d	3	5	3

Table 4 delineates the performance of this system. We note that it is better than Setiono's single-rule one, trailing but shortly behind the two- and three-rule systems.

Table 4 Performance of an evolved single-rule system, compared with a single-rule system found by Setiono.

	Our system		Setiono's	
	Training	Test	Training	Test
Benign	98.6%	98.2%	98.7%	97.4%
Malignant	95.0%	89.8%	93.4%	87.5%
Overall	97.4%	95.3%	96.9%	94.0%

Setiono's single-rule system is given below:

if ($v_2 \leq 4$) **and** ($v_6 \leq 5$) **then** (*output* **is** *Benign*)

else (*output* **is** *Malignant*)

This system is not fuzzy, i.e., it uses standard Boolean (binary) logic, which is (by definition) a special case of fuzzy logic. We can transform our evolved fuzzy system into a Boolean one, thereby enabling a more detailed comparison between the two. (Note: such a transformation involves no reduction in performance, though it becomes no longer possible to provide a degree of truth, or reliability, of the output diagnostic). The transformed system is given below:

if ($v_2 \leq 5$) **and** ($v_6 \leq 5$) **and** ($v_8 \leq 8$) **then** (*output* **is** *Benign*)

else (*output* **is** *Malignant*)

We note that the two systems are quite similar, which is notable given the use of completely different methodologies to obtain them. However, ours is refined, with the added v_8 variable, and a different right term for the v_2 comparison. This refinement is sufficient in order to give rise to a marked increase in performance.

V. Conclusions

We presented an evolutionary approach for discovering fuzzy systems for breast cancer diagnosis. By judiciously designing an appropriate representation scheme (genome) and fitness function, the genetic algorithm was then able to produce high-performance systems. Comparing these with some of the best known systems to date, we remarked that not only did we attain higher performance (at least in some cases), but, as importantly, our evolved systems are perhaps the simplest ones available. As for the latter, we ended up with minimal, single-rule systems, with but a small number of variables. Thus, we obtain high-performance, human-comprehensible systems that are able to solve this important medical classification problem.

These promising results incited us to engage in further investigation of this approach. We are currently extending our experiments, using other representations and more elaborate fitness functions, with preliminary encouraging results, which we hope to report in the near future.

References

[1] R. R. Yager and L. A. Zadeh, *Fuzzy Sets, Neural Networks, and Soft Computing*, Van Nostrand Reinhold, New York, 1994.

[2] J.-S.R. Jang and C.-T. Sun, "Neuro-fuzzy modeling and control," *Proceedings of the IEEE*, vol. 83, no. 3, pp. 378–406, March 1995.

[3] P. Vuorimaa, "Fuzzy self-organizing map," *Fuzzy Sets and Systems*, vol. 66, pp. 223–231, 1994.

[4] M. A. Lee and H. Takagi, "Integrating design stages of fuzzy systems using genetic algorithms," in *1993 IEEE International Conference on Fuzzy Systems*. IEEE, 1993, pp. 612–617.

[5] H. Heider and T. Drabe, "Fuzzy system design with a cascaded genetic algorithm," in *Proceedings of 1997 IEEE International Conference on Evolutionary Computation*. IEEE and IEEE Neural Network Council and Evolutionary Programming Society, 1997, pp. 585–588.

[6] N. E. Nawa, T. Hashiyama, T. Furuhashi, and Y. Uchikawa, "A study on fuzzy rules discovery using pseudo-bacterial genetic algorithm with adaptive operator," in *Proceedings of 1997 IEEE International Conference on Evolutionary Computation*. IEEE and IEEE Neural Network Council and Evolutionary Programming Society, 1997.

[7] T. Fukuda and K. Shimojima, "Fusion of fuzzy, nn, ga to the intelligent robotics," in *Proceedings of the 1995 IEEE International Conference on Systems, Man and Cybernetics*. IEEE, 1995, vol. 3, pp. 2892–2897.

[8] O.L. Mangasarian, W.N. Street, and W.H. Wolberg, "Breast cancer diagnosis and prognosis via linear programming," Mathematical Programming Technical Report 94-10, University of Wisconsin, 1994.

[9] C.J. Merz and P.M. Murphy, "Uci repository of machine learning databases," http://www.ics.uci.edu/~mlearn/MLRepository.html, 1996.

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

[11] B.G. Kermani, M.W. White, and H.T. Nagle, "Feature extraction by genetic algorithms for neural networks in breast cancer classification," in *Proceedings of the 1995 IEEE Engineering in Medicine and Biology International Conference*, 1995, pp. 831–832.

[12] R. Setiono, "Extracting rules from pruned neural networks for breast cancer diagnosis," *Artificial Intelligence in Medicine*, pp. 37–51, 1996.