

Automatic Generation of Fuzzy Models By using Iteration through Training Data

Abdulrazaq Alsuhail Almutairi

Information & Computer Center

The Public Authority for Applied Education and Training

The Ministry of Education

The State of Kuwait

Abstract

The fuzzy logic technique has been widely employed in different applications. It has the capability of mimicking the human mind to adequately apply modes of reasoning, which are approximate rather than exact. In this paper, one of the fuzzy logic challenges, related to learning capabilities, is handled by proposing an approach that auto-generates fuzzy models using previous observations. This approach involves generating membership functions and fuzzy rules. In other words, this paper shows how to use fuzzy logic as a machine learning algorithm, based on a training dataset. The approach is based on iteration. In each iteration, some parameters are changed to generate fuzzy rules. The iteration process will stop once the error goes below a specified value. Results show that the fuzzy system, based on rules generated using the proposed approach, has achieved a high level of accuracy.

1. Introduction

The classical models that depend on complex mathematical equations, and ignore qualitative and subjective information, are not suitable or practically complex enough to apply to many problems [1]. Recently, models based on fuzzy logic have been widely employed in different fields such as engineering, economics, and biotechnology.

Fuzzy logic is considered as a data mining technique that translates defined rules to mathematical equivalents for performing the input-output mapping. It has the capability for handling vagueness and incompleteness of information [2]. Fuzzy logic is very effective in cases where the available information is not accurate enough to justify the use of numbers, as there is a tolerance for imprecision. In addition, it is able to describe non-linear relationships. Moreover, models based on fuzzy logic are transparent as relationships between inputs and outputs are expressible [3]. In other words, it is not a black box like some of the machine learning algorithms. Although the advantages of fuzzy logic are mentioned, it is usually based on a subjective estimation that is highly dependent on human understanding for modelling problems. This means that one of the fuzzy logic shortfalls is the absence of learning capability as in machine learning algorithms. The main challenge of fuzzy modelling is the extraction of fuzzy rules that suit the data and with a high level of extent.

In the past few years, many approaches were proposed to adjust fuzzy membership functions and generate fuzzy rules by analysing previous observations (training data). Some of those approaches focus only on adjusting fuzzy membership functions, while others look at fuzzy rules extraction. Most of those approaches are based on data clustering techniques, such as genetic algorithms that target grouping data into clusters based on similarity measurements.

The issue with fuzzy models based on those approaches is a lack of harmony between membership functions and fuzzy rules as they generate only one of them, not both of them. In other words, they do not achieve the maximum level of fitness between membership functions and fuzzy rules. As a result of that issue, poor results are obtained in some scenarios.

In this paper, an approach is proposed to auto-generate fuzzy models based on previous observations (training data set). The approach generates both membership functions and fuzzy rules rather than focusing on one side, to increase the level of fitness between data and fuzzy rules. In addition, the proposed approach gives users control over speed and accuracy by specifying some parameters in training that will be explained later in section 3.

The paper is divided as follows: an overview of fuzzy logic and its different components is provided in section 2. The proposed approach is presented in section 3. Section 4 and 5 go through results obtained by applying the proposed approach to some classification problems. Section 5 presents related work. Section 6 provides a conclusion and future work plan.

2. Brief Background

2.1 Fuzzy logic overview

Fuzzy logic systems consist of four main components; fuzzifier, fuzzy rules, inference engine, and defuzzifier. The mechanism of fuzzy logic is shown in Figure 1. The First step is known as fuzzification which converts crisp input values to a fuzzy set by using fuzzy linguistic variables and membership functions. The inference engine then applies fuzzy set obtained from the fuzzifier to specified rules (fuzzy rules) to produce a fuzzy output. The final step is mapping the fuzzy output to a crisp value through the defuzzifier.

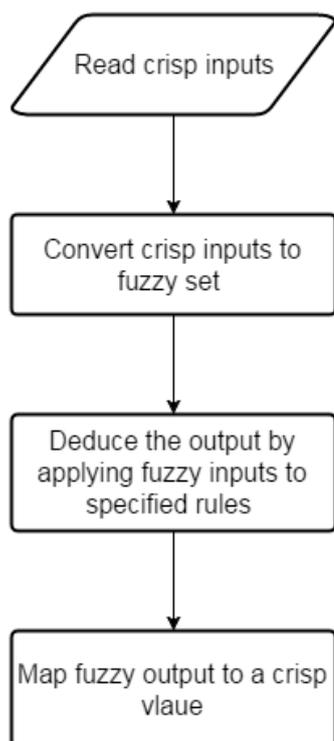


Figure 1: Fuzzy Logic Systems Architecture

2.2 Linguistic Variables

Linguistic variables represent input or output variables of a system in terms of natural language rather than numerical values [1]. A linguistic variable generally involves using a set of linguistic terms. For example, Let velocity (v) is the linguistic variable which represents the velocity of a vehicle. To qualify the speed, terms such as “high” and “slow” are used in real life. These are the linguistic values of the velocity. Therefore, $V(v) = \{\text{too-high, high, slow, hot, too-slow}\}$ is the set of decompositions for the linguistic variable speed.

2.3 Membership functions

Membership functions are employed in both the fuzzification and defuzzification processes to convert the crisp input values to fuzzy set and vice versa. In other words, membership functions are used to quantify a linguistic term. A crisp value can be fuzzified using multiple membership functions but with different levels of membership. Membership functions can be presented in different forms such as triangular, trapezoidal, or Gaussian as shown in figure 2.

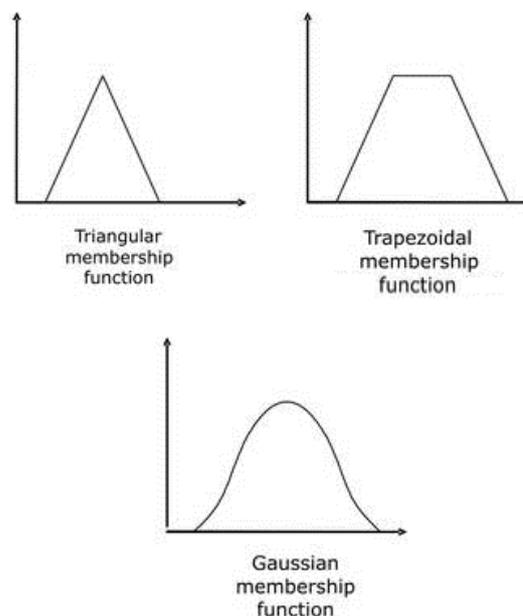


Figure 2: Membership functions

2.4 Fuzzy Rules

A fuzzy rule is an IF-THEN rule that includes conditions (AND, OR, NOT operations between linguistic variables) and a conclusion. The evaluation of a fuzzy rule to deduce an output is carried out through fuzzy set operations. The OR operation is mapped to the max, while AND is mapped to min. The NOT operation is mapped to $(1 - \text{membership value})$ [1].

3. The Proposed Approach

The idea of the approach in this paper is inspired by neural network training, being similar in generating a model then checking the error to find out whether it is acceptable or not. The iteration continues to adjust the model until hitting a max number of iterations or obtaining an error below a specified value.

Figure 3 shows how the approach works to generate fuzzy systems, as follows:

- 1- The first step is reading design parameters which include: max root mean square (RMS) error; max number of epochs; membership functions to be used; min number of linguistic terms used in membership functions. Table 1 explains each of those parameters.
- 2- The second step is selecting the most relevant attributes from the training data set and excluding attributes that do not have an impact on specifying the output, using a Correlation Feature Selection algorithm (CFS).

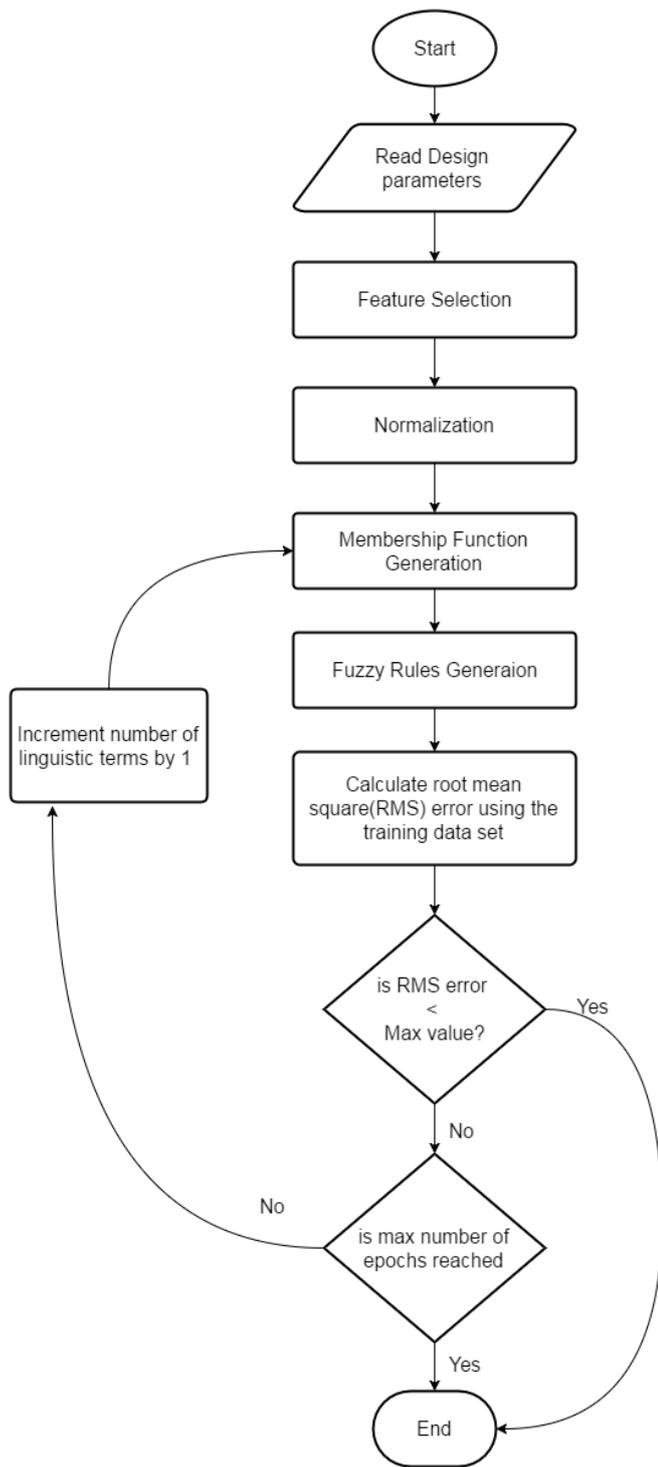


Figure 3: The Proposed Approach

3- Thirdly is normalising the selected attributes so all attributes will be in the range of zero to one. This step will simplify the generation of membership functions, as all attributes will be in the same range, so the design of membership functions for all inputs will be similar.

4- Next is to generate input and output membership functions, based on the number of linguistic terms. Note that singleton

membership functions will be used with attributes that have discrete values.

5- Fuzzy rules are then generated by going through each record in the training dataset and mapping each numeric value into a description, based on the number of linguistic terms used in the iteration as follow:

$$0 \leq \text{attribute value} < A_1 \rightarrow A_1$$

$$A_{i-1} \leq \text{attribute value} < A_i \rightarrow A_i$$

$$A_{n-1} \leq \text{attribute value} \leq 1 \rightarrow A_n$$

Where:

n is a number of linguistic terms.

$$A_i = i/n, \quad i \leq n$$

1- The next step is to calculate the output for each training record through the generated fuzzy model, then to calculate the error.

2- The root mean square error is then calculated and compared with the max value specified in the first step. The iteration will stop in cases where the calculated error is less than the max value.

3- If the max number of epochs is not hit, the number of linguistic terms can be increased by one, then proceed again from step 4.

The idea behind increasing the linguistic terms in each iteration is to generate rules that cover more scenarios than previous epochs.

Table 1: Design Parameters

Parameter	Description
Max RMS Error	Max value accepted for differences between values produced by the generated model and those actually observed.
Max Number of Epochs	Max number of iterations to generate a fuzzy model in the training phase.
Min Number of Linguistic Terms	The min value that will be used to construct membership and fuzzy rules

4. Experiment 1

4.1 Overview

In this experiment, we are trying to create a decision model that evaluates a car from different perspectives, then decides whether the car passes the evaluation or not. The training dataset [4] contains 1200 labelled samples, with a class indicating whether it passed the evaluation or not. Each sample consists of six attributes described as follows:

- **Buying:** Buying Price. It is rated in a range of zero to four, where four means a very high price and zero means a very low price.
- **Maintenance:** Maintenance price. It is rated in a range of zero to four, where four means a very high price and zero means a very low price.
- **Doors:** Number of doors. It takes one of the following values (2, 3, 4, 5 or more).
- **Persons:** Capacity in terms of the number of individuals. It takes one of the following values (2, 3, or more).
- **Lug Boot:** The size of the luggage boot. It is rated in a range of 0 to 3, where 3 means big.
- **Safety:** Safety level. It is rated in a range of 0 to 3, where 3 means a high degree of safety.

The class can be either one or zero, where one indicates passing the evaluation and zero refer to failure.

4.2 Training

The training has been run by setting max RMS error to 0.2, the max number of epochs to 10, the min number of linguistic terms to 4, and triangle membership functions. The training stopped at the fourth epoch achieving 0.09 RMS. Table 3 shows the number of linguistic terms, the number of fuzzy rules generated, and RMS error in each epoch.

Table 2: Experiment 1 Training Results

Epoch No.	No. Linguistic Terms	No. Fuzzy Rules	RMS Error
1	5	128	0.51
2	6	297	0.33
3	7	468	0.29
4	8	864	0.09

4.3 Evaluation

The evaluation process was carried out using a dataset consisting of 1730 samples. The process evaluated the performance of the generated model in terms of true positive, true negative, false positive, and false negative. Table 3 presents the performance evaluation of the generated model; it shows that the generated model achieved a high true positive at 0.96.

Table 3: Performance Evaluation for Experiment 1

True Positive	True Negative
0.96	0.99
False Positive	False Negative
.01	.04

5. Experiment 2

5.1 Overview

In this experiment, we are trying to create a model that finds out the presence of heart disease in a patient. The training dataset [5] contains 200 labelled samples with a class indicating the presence or absence of heart disease in a patient. Each sample consists of thirteen attributes described as follows:

- **Age:** Age of the patient in years.
- **Sex:** Sex of the patient (1 for male; 0 for female).
- **CPT:** Chest Pain Type (1 for typical; 2 for atypical; 3 for non-angina; 4 for asymptomatic)
- **trestbps:** Resting blood pressure in mm Hg.
- **Chol:** Cholesterol level in mg/dl.
- **fbs:** A flag that indicates whether fasting blood pressure is greater than 120 mg/dl (0 for false; 1 for true).
- **restecg:** Resting electrocardiographic results (0: normal; 1: having ST-T wave abnormality, 2: showing probable or definite left ventricular hypertrophy).
- **thalach:** Maximum heart rate measured.

- **exang:** A flag that indicates whether exercise-induced angina (0 for false; 1 for true).
- **oldpeak:** ST depression generated by exercise, compared to rest.
- **slope:** The slope of the peak exercise ST segment (1 for upsloping; 2 for flat; 3 for downsloping).
- **ca:** Number of major vessels
- **thal:** (3 for normal; 6 for fixed defect; 7 for reversible defect).

The class can be either one or zero, where one indicates the presence of heart disease and zero refers to absence.

5.2 Training

The training was run by setting max RMS error to 0.2, the max number of epochs to 10, the min number of linguistic terms to 4, and triangle membership functions. The training stopped at the fifth epoch achieving 0.2134 RMS. Table 3 shows the number of linguistic terms, the number of fuzzy rules generated, and RMS error in each epoch.

Table 4: Experiment 2 Training Results

Epoch No.	No. Linguistic Terms	No. Fuzzy Rules	RMS Error
1	3	167	0.33
2	4	212	0.25
3	5	244	0.66
4	6	251	0.22
5	7	268	.14

5.3 Evaluation

The evaluation process was carried out using a dataset consisting of 350 samples. The process evaluated the performance of the generated model in terms of true positive, true negative, false positive, and false negative. Table 5 presents the performance evaluation of the generated model; it shows that the generated model achieved a high true positive at .9604 and a low false positive a 0.006.

Table 5: Performance Evaluation for Experiment 2

True Positive	True Negative
0.9604	0.9940
False Positive	False Negative
0.006	0.0396

6. Related Work

There have been some attempts by researchers to auto-generate fuzzy rules; one of those researchers [6] proposed an approach to generate classification rules using granulation-based adaptive clustering. The approach carries out data clustering without the need to define any parameters as the number of clusters. Data clustering is performed at different levels of granulation and the best clustering to generate fuzzy rules is then selected. The average accuracy achieved by that approach was around 78.64%.

Another attempt [7] aimed at auto-generating a fuzzy classification system using hyper-cone membership functions, based on genetic algorithms. In this approach, rules are added gradually until the best performance is achieved. The average accuracy obtained by that approach was around 73.23%.

In [8], researchers tried to generate fuzzy rules based on subtractive clustering and genetic algorithms. The proposed method searches for the fuzzy interference system structure and the number of rules that achieve the best performance. The average accuracy achieved was 74.50%

7. Conclusion and Future Work

It was found that the proposed approach can be effectively used to auto-generate a fuzzy-based classification system. The experiments carried out to test the proposed approach showed that the generated classification systems achieved a higher accuracy compared to approaches proposed in the related work section. The approach increases the number of linguistics in each epoch to cover more scenarios and reduce error. However, it has been found that increasing the number of linguistic terms in each epoch leads to increasing the number of rules. As a result of this growth, the generated system becomes more complex and slower.

The future work plan will focus on improving the approach by decreasing the number of rules generated, by using a rules reduction technique. It will also involve using a min and max number of linguistic terms for each linguistic variable, rather than using the same number of linguistic terms for all variables (inputs), to reduce the complexity of generated systems.

8. References

- [1] L. Zadeh, "Fuzzy sets," *Inf. Control*, 1965, vol.353, pp.338-353.
- [2] K. Pulo, "Fuzzy Logic vs. Machine Learning" , [Online] Available from <http://www.kev.pulo.com.au> [Accessed: 20th April 2015].
- [3] A. Oscar, "Historical review of evolutionary learning methods for Mamdani-type fuzzy rule-based systems: Designing interpretable genetic fuzzy systems", *International Journal of Approximate Reasoning*, 2011, v.52 n.6, p.894-913.
- [4] M. Bohanec And B. Zupan, "Car Evaluation Dataset", [Online] Available from: <http://archive.ics.uci.ed> [Accessed: 19th April 2016].
- [5] R. Detrano, "Heart Disease Dataset", [Online] Available from: <http://archive.ics.uci.ed> [Accessed: 19th April 2016].
- [6] M. Al-Shammaa And M. Abbod, "Automatic Generation of Fuzzy Classification", *International Conference on Neural Networks-Fuzzy Systems* , 2014, pp.74 - 79.
- [7] L. HNG, "Automatic Generation of Fuzzy Classification Systems Using Hyper-Cone Membership Functions", *Computational Intelligence in Robotics and Automation*, 2003, pp.658-663.
- [8] M. Al-Shammaa and M. Abbod, "Automatic generation of fuzzy classification rules using granulation-based adaptive clustering", *Systems Conference*, 2015, pp.653 – 659.