A preliminary study to apply the Quine McCluskey algorithm for fuzzy rule base minimization

Leonardo Jara
Departamento de Ciencias de la Computación e Inteligencia Artificial
University of Granada
18071-Granada, Spain
E-mail: ljara@correo.ugr.es

Antonio González
Departamento de Ciencias de la Computación e Inteligencia Artificial
University of Granada
18071-Granada, Spain
E-mail: A.Gonzalez@decsai.ugr.es

Raúl Pérez
Departamento de Ciencias de la Computación e Inteligencia Artificial
University of Granada
18071-Granada, Spain
E-mail: fgr@decsai.ugr.es

***Abstract*—The Fuzzy Rule-Based Classification Systems (FRBCS) are classification models that use fuzzy rules to represent knowledge. FBRCS are popular today, with numerous applications and studies of their behavior and efficiency. This work is dedicated to studying a method that allows the minimization of FBRCS generated by the Chi Algorithm, using the Quine-McCluskey method so that the number of generated rules can be reduced, without greatly altering the accuracy, thus improving the simplicity of the model.**

***Index Terms*—Interpretability, Classification problems, Fuzzy rules, XAI.**

I. Introduction

The use of Fuzzy Rule-Based Classifications Systems (FRBCS), is usually related to the purpose of obtaining the greatest possible interpretable knowledge of a given problem. Interpretability has been and is a frequent topic of discussion [1], however one of the parameters that is always present in any analysis is the number of rules obtained by a classifier. Thus, it is not always possible to obtain a reduced number of rules as a solution to a problem. This happens for example with one of the most frequently used fuzzy rule-learning algorithms, such as the Chi algorithm [2], which usually obtains a high number of rules, and therefore makes it difficult to interpret the provided solution.

In this way, we are interested in minimizing fuzzy rule bases. By using logical minimization, a set with a high number of rules can be reduced to a set that uses fewer variables and/or rules that are to some extent equivalent. Thus, the aim of this work is given a set of fuzzy rules, obtained for example by the Chi algorithm, to define a procedure that is able to reduce the number of fuzzy rules while maintaining its predictive ability.

To achieve this goal, we will first define a binary encoding of the fuzzy rule set so that it can serve as input for the Quine-McCluskey method [3]. This algorithm defines a procedure for the simplification of Boolean functions. By using appropriate coding we can simplify a fuzzy set of rules, and thus reduce the number of rules required.

This work has been partially funded by the Spanish Projects TIN2015-71618-R, RTI2018-098460-B-I00 and co-financed by FEDER funds (European Union), and also by the National Program of Postgraduate Scholarships Abroad "Don Carlos Antonio López, BECAL", granted by the Government of the Republic of Paraguay.

By using the Quine-McCluskey method, we can modify a set of basic fuzzy rules (Mamdami-type fuzzy rules), to a set of extended fuzzy rules like those used by the NSLV algorithm [4].

A similar attempt to reduce the number of fuzzy rules was made in [5] using Karnaugh's maps [6]. The main limitation of this technique is that it requires the use of a maximum of 6 binary variables (two fuzzy variables with three linguistic labels each one), which is why it becomes impracticable for the datasets that are normally used.

Another more recent study focuses on the design of fuzzy models that can be interpreted through semantic cointension [7], and uses the Espresso algorithm as a minimization method [8]. Obtaining good results in the analysis of interpretability with a different approach to the one proposed in this work.

In the next section we present some preliminary of the work, such as the basic model of fuzzy rules with weight and the extended rule model, the Chi algorithm and its use, and the Quine-McCluskey algorithm. In section III we describe the coding used to represent the fuzzy rule set and the use we make of the Quine-McCluskey algorithm with that coding. Section IV will show the experimentation done that demonstrates the interest of the proposal presented. Finally, the paper ends with some conclusions.

II. PRELIMINARIES

The basic elements that this work requires as previous elements to the development of the proposal are the rule model in which is expressed the knowledge that is going to be minimized, the algorithm used for the extraction of such rules and the Quine-McCluskey algorithm that is used as a method for the minimization.

Below we briefly describe each of these elements.

A. *Rule model*

The minimization proposal that we carry out in this work starts from a set of basic fuzzy rules that could be obtained in multiple ways, however, with the idea of being able to make a complete experimentation with the model we will assume that the initial set of rules has been obtained from the Chi algorithm [2].

978-1-7281-6932-3/20/$31.00 ©2020 IEEE

The usual structure of a fuzzy rule set using the Chi algorithm is [9]:

$$\text{Rule } R_j: \text{ If } x_1 \text{ is } A_{j1} \text{ and ... and } x_n \text{ is } A_{jn} \text{ then} \\ \text{Class } = C_j \text{ with } RW_j \tag{1}$$

Where R_j is the label of the $_{j-th}$ rule, $x = (x_1, ..., x_n)$ is a $n - dimensional$ pattern vector that represents the example, A_{ji} is a linguistic label modeled by a triangular membership function, C_j is the class label and RW_j is the rule weight. The rule weight used in this work is the Penalized Certainly Factor [10] (PCF):

$$RW_j = PCF = \\ \frac{\sum_{x_p \in ClassC_j} \mu_{A_j}(x_p) - \sum_{x_p \notin ClassC_j} \mu_{A_j}(x_p)}{\sum_{p=1}^{P} \mu_{A_j}(x_p)} \tag{2}$$

where $\mu_{Aj}(x_p)$ is the matching degree of the example x_p with the antecedent part of the fuzzy rule R_j and it is calculated as follows.

$$\mu_{A_j}(X_p) = \prod_{i=1}^{n} \mu_{A_{ji}}(X_{pi}) \tag{3}$$

being $\mu_{A_{ji}}(X_{pi})$ the membership degree of the value x_{pi} to the fuzzy set A_{ji} of the Rule R_j.

B. The CHI Algorithm

The algorithm of Chi is based on the model proposed by Wang and Mendel [11] but with focus on classification systems and the steps to apply are the following.

1) The first step is to define a set of linguistic labels for each antecedent variable.
2) The second step is to generate a fuzzy rule for each example. A fuzzy rule is generated for each example x_p as follows.
   a) The membership degrees of each value x_{pi} to all the different fuzzy sets if the $i - th$ variable are computed. For each variable, the linguistic label with the greatest membership degree is selected.
   b) The antecedent part is determined by the intersection of the selected linguistic labels and the consequent is the class label of the example (y_p). All rules will have exactly the same number of antecedents as variables in the problem (n).
   c) The rule weight is computed using the PCF weight (2).

When we have rules with the same antecedent, we make the rule that has most weight and assign the class corresponding to that rule.

Once the structure is obtained, to classify the new examples we will carry out the following steps.

1) Finding the maximum matching degree of coincidence between the examples and the rule using the matching degree define in (3).
2) The degree of association of the example with each rule is calculated as

$$b_j(x_p) = \mu_{A_j}(x_p).PCF \tag{4}$$

3) The class corresponds to the rule with the highest level of association.

$$Class = argmax_{c=1,...,m}(max_{R_j \in RB; C_j = c} b_j(x_p)) \tag{5}$$

The main advantage of the method described is that it is a very simple method that achieves good levels of efficiency, which has led to its use in many applications, but the main disadvantage is that it produces in general a very high number of rules that makes it difficult to interpret them.

As an example of this problem we can analyze the results obtained in [12] using this algorithm and shown in Table I on several well-known databases obtained from the UCI repository [13]. We can see that the percentage of rules is very high in relation to the examples in some cases. Thus, on average the number of rules represents just over 41% of the examples, so each rule covers approximately two and a half examples. This fact is especially relevant in the BREAST CANCER database, where the number of rules represents more than 78% of the number of examples present in Table I. These data show that the Chi algorithm is very dependent on the number of antecedent variables that need to be used, and therefore in some cases the rules are not very representative, many rules are required to represent a problem and the interpretability of an FRBCS can be lost using this algorithm.

The aim of this work is to decrease the rule base generated by the Chi algorithm using a Boolean function minimization developed with Quine McCluskey in order to improve the simplicity of the problem and generate a set of extended rules.

TABLE I
SUMMARY RESULTS USING CHI ALGORITHM AND THREE LINGUISTIC LABELS PER VARIABLE

Dataset	**Summary**			
	Examples	***Attributes***	***Rules***	***R vs E****
Iris	150	4	14	9.3%
Wine	178	13	127	71.3%
Breast Cancer	569	32	448	78.7%
Glass	214	9	41	19.1%
PIMA	768	8	154	20%
MEAN	375.8	13.2	157	41.7%

*Percentage of rules vs examples.

The extended fuzzy rule model is a very similar fuzzy rule model to the one used so far but in which the value assigned to a variable is allowed to be a subset of the fuzzy labels of the variable's domain. When a variable is assigned all the labels in its domain, the variable is irrelevant and can be removed from the rule.

C. Quine McCluskey Method

In this section we present the basic definitions and the steps to apply the Quine McCluskey method. For a more detailed description of this algorithm see [3].

In the algorithm:

- Literal: It is a logical variable or its negation (q or $\bar{q}$).
- Minterm: It is a product of the literals where each a variable appears exactly once either true or complemented form.
- Prime implicant: it is the product which cannot be combined with de another term to eliminate a variable for further simplification.
- Essential prime implicant: it is a prime implicant that is able to cover an output of the function which is not covered by any combination of prime implicant called essential prime implicant.

The Quine McCluskey (QM) method uses the following three basic simplification laws:

- $q + \bar{q} = 1$ (Complement)
- $q + q = q$ (Idempotent)
- $q(w + z) = qw + qz$ (Distributive)

Where q, w and z are literals

The QM method has the following steps:

- Find the prime implicant: In this step, we replace the literal in form of 0 and 1 and generate a table. Initially, the number of rows in table is equal to the total number of minterms of the original un-simplified function. If two terms are only different in one bit, one variable is appearing in both form (variable and negation), then we can use complement law. Iteratively, we compare all terms and generate the prime implicant.
- Find the essential prime implicant: Using prime implicant from above step, we generate the table to find essential prime implicants. Note that some prime implicants can be redundant and may be omitted, but if they appear only once, they cannot be omitted and provide prime implicant.
- Find other prime implicant: It is not necessary that essential prime implicant cover all minterms. In that case, we consider other prime implicant to make sure that all minterms has been covered.

In general, QM method provides a better method for the function simplification than the Karnaugh map, but still is an NP-Hard problem, and it became impractical for large input size due to exponential complexity.

III. PROPOSED METHOD OF CODING FUZZY RULES

To apply the QM method for the minimization of fuzzy rules sets obtained by the Chi algorithm, and since QM method is able to minimize only Boolean function, we must first adapt the linguistic values of the rule base to a set of binary variables representing those linguistic variables.

The process is as follows, from the set of fuzzy rules obtained by the Chi algorithm, we use a process of coding such rules so that they can be used as input for the QM method. After applying this method, we decode and simplify the set of rules obtained. As we'll see later this rule set can be interpreted as an extended fuzzy rule set. The process is described in the "Fig. 1".

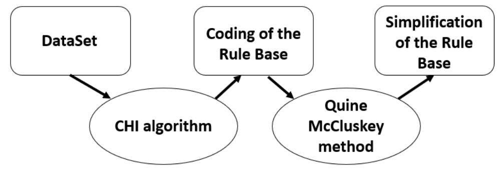

Fig. 1. Flowchart for the minimization of a fuzzy rule set.

In this way we start from the following set of fuzzy rules which have been obtained by the Chi algorithm (1)

Each linguistic variable has an associated fuzzy domain, with the idea of simplifying the description in the following explanation and in the experiment, we will take as a reference for all the linguistic variables a domain with three labels, as described in "Fig. 2".

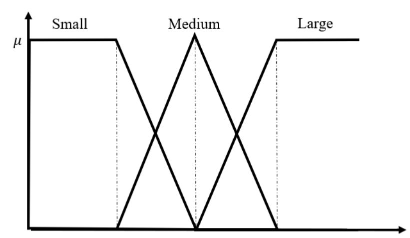

Fig. 2. Fuzzy domains for variables $x_1, ..., x_n$.

The process is the same for a larger number of linguistic labels.

A. Coding of the Rule Base

As we said before, to use the QM method, it is necessary to transform the rule base generate by Chi algorithm to a QM compatible model, for this we must transform the linguistic labels into binary labels in order to change the coding of the model.

According to QM method, if two terms are only different in a bit, the law of complement can be applied by generating the prime implicants. This is a very important point when coding linguistic label.

As explained above, these linguistic labels must be changed to binary values. The number of bits needed to represent a linguistic label is given by the following equation.

$$Bits = Round(\frac{log_2(L)}{log_2(2)} + 1) \tag{6}$$

where L is the number of linguistic labels in the domain and the rounding is down. For example, 3 linguistic labels use 2 bits, 5 linguistic labels use 3 bits and so on. This equation is a generalization used to have a reference of how many bits are needed for an odd number of linguistic labels.

Having as reference the three linguistic labels (Small, Medium and Large) and considering that the generation of the prime implicants occurs when the minterm differ in a bit (Hamming distance between linguistic labels is equal to 1). We will only need 2 bits and therefore we will use the following coding for the 3 linguistic labels shown below in the figure "Fig. 3".

Label	Code
SMALL	00
MEDIUM	01
LARGE	11
NOT USE	10

Fig. 3. Linguistic labels

Therefore, we only have to replace each linguistic label with its binary coding. In all cases, the first linguistic label is represented by the binary number "0" in all the bits that make up the coding and the contiguous linguistic labels must be differentiated by only one bit.

In view of the above, the code "10" above is not used because the bit difference between adjacent linguistic labels must be equal to 1.

We can show an example of such coding for a problem with four antecedent variables. Let's suppose two rules for the same class in the consequent:

If x_1 is MEDIUM and x_2 is SMALL and x_3 is MEDIUM and x_4 is SMALL then Class = C_1 with RW_1

If x_1 is MEDIUM and x_2 is SMALL and x_3 is MEDIUM and x_4 is MEDIUM then Class = C_1 with RW_2

First, we replace the linguistic labels with the binary coding defined above, we take the weight off the rules and we obtain:

If x_1 is 01 and x_2 is 00 and x_3 is 01 and x_4 is 00 then Class = C_1

If x_1 is 01 and x_2 is 00 and x_3 is 01 and x_4 is 01 then Class = C_1

Finally, assuming these are the only two rules for that class, we transform it into a Boolean expression with which we can apply the QM method.

01000100 + 01000101 $\rightarrow$for Class C_1

Through this structure we manage to create a truth table for each class, in order to use the QM method independently of each class as shown in the "Fig. 4".

Repeating the same process for all the rules by grouping them with their respective classes.

bit 1	bit 2	bit 3	bit 4	bit 5	bit 6	bit 7	bit 8	Class
0	1	0	0	0	1	0	0	1
0	1	0	0	0	1	0	1	1

Fig. 4. Truth table for Class 1

In summary, the steps to perform the coding are as follows:

1) Describe the number of bits required for the linguistic labels used using (6).
2) Encode the linguistic labels with maintaining the difference of 1 bit between each contiguous label.
3) Concatenate the different assignments.

B. Simplification of the Rule Base

After applying the QM method, two types of cases can be generated. Either the original bits are kept and there is no simplification or one of the bits is replaced by a *, where * symbolizes the elimination of a boolean variable in the minimization process. In this case, the options are:

- 0*, which means that the assignment to the variable is SMALL or MEDIUM.
- *1, which means MEDIUM or LARGE,
- and *0 and 1* that do not imply simplification, the first is equivalent to using SMALL and the second to using LARGE.

Using the previous example, we obtain the following simplified Boolean function:

If x_1 is 01 and x_2 is 00 and x_3 is 01 and x_4 is 0* then Class = C_1

Thus, the two previous rules have been transformed into a single extended rule. This rule is:

If x_1 is MEDIUM and x_2 is SMALL and x_3 is MEDIUM and x_4 is $\{$SMALL or MEDIUM$\}$ then Class = C_1

Finally, the weight of this rule is recalculated using an adaptation of (2) to the extended rule model and we obtain the following extended rule:

If x_1 is MEDIUM and x_2 is SMALL and x_3 is MEDIUM and x_4 is $\{$SMALL or MEDIUM$\}$ then Class = C_1 with RW_3

where RW_3 is the weight of the new rule.

This process would apply to all classes and would allow us to obtain the final set of rules.

IV. EXPERIMENTAL STUDY

In this section we want to check if the proposed method allows to reduce the number of fuzzy rules keeping the accuracy level. To do this we use the databases described in Table II and obtained from the UCI repository [13].

In this experiment we use fuzzy domains composed by three symmetric triangular membership function with the upper and

TABLE II
DATABASE USED FOR THE EXPERIMENTAL STUDY

Datasets	Summary		
	Examples	***Attributes***	***Number of Classes***
Iris	150	4	3
Wine	178	13	3
Appendicitis	106	7	2
Pima	768	8	2
Wifi	2000	7	4
Balance	625	4	3
Banana	5300	2	2
Ecoli	325	7	8
Haberman	290	3	2
Newthyroid	201	5	3
Bupa	322	6	2
Phoneme	5390	5	2
Titanic	2185	3	2
Page Block	5463	10	5
Winequality	1599	11	6

lower labels open, for all variables. We use the Chi algorithm to obtain the initial set of fuzzy rules.

The study is focused in two parameters: the number of rules and the accuracy. On these parameters, we use a cross validation process to check the results obtained on the initial fuzzy rule set and the minimized fuzzy rule set obtained by the QM method.

Table III shows the results after performing the experiment. CHI represents the data obtained using the Chi algorithm, and CHI+QM represents the data obtained using the proposed method.

TABLE III
DATABASE USED FOR THE EXPERIMENTAL STUDY

Datasets	CHI		CHI + QM		Rule
	Rules	***Accuracy***	***Rules***	***Accuracy***	***Reduction %***
Iris	16	0.929545	8	0.925	50%
Wine	104	0.939583	88	0.939583	15%
Appendicitis	28	0.891666	20	0.895833	29%
Pima	192	0.756544	110	0.755497	43%
Wifi	127	0.918333	55	0.915666	57%
Balance	79	0.897861	26	0.844385	67%
Banana	8	0.604968	6	0.588616	25%
Ecoli	63	0.689690	38	0.685567	40%
Haberman	19	0.776623	7	0.787013	63%
Newthyroid	19	0.858333	11	0.831666	42%
Bupa	61	0.624	39	0.608	36%
Phoneme	65	0.710973	25	0.702727	62%
Titanic	10	0.779084	7	0.779084	30%
Page Block	44	0.920744	24	0.919097	45%
Winequality	355	0.585673	245	0.584527	31%
MEAN	79.3	0.792241	47.2	0.784150	42%

We can appreciate the reduction or rules for all the datasets used in this study in the "Fig. 5".

We have focused this analysis on the study of 2 parameters, the accuracy and the average number of rules. To make the comparison we have used the Wilcoxon test which is a non-parametric test to compare the mean range of two related samples and determine if there are differences between then.

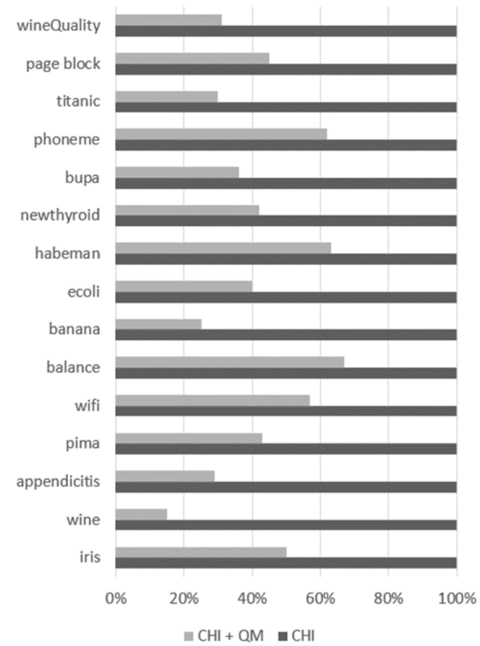

Fig. 5. Percentage reduction of rules using the proposed rule minimization method.

In relation to the first parameter, the accuracy, the Wilcoxon test obtained a p-value of 0.002382, determining that there is no significant difference between the accuracy of the two algorithm and that there is a 16.4% confidence that they are different.

In relation to the second factor, we performed the same analysis to compare the number of rules generated by both algorithm and the Wilcoxon test obtained a p-value of 0.0007193, determining that there is a significant difference in the number of rules generated by two algorithms.

We can see in Table III that the rule base accuracy of the rule base remains similar between the algorithm of CHI et. Al and the application of the rule minimization method of Quine McCluskey.

It is important to note that the reduction in the size of the rule set is on average 40% in relation to the initial fuzzy rule set.

The most extreme case is in the Balance dataset, which with 625 examples, the Chi algorithm needed 79 fuzzy rules, which after applying the minimization process have been transformed into 26 extended fuzzy rules.

The complexity of the proposed method increases as we increase the number of attributes, for this reason we choose a model based on 3 linguistic labels since due to the coding performed we increased the number of variables in a multiple given by (6), taking as an example the dataset wine that has 13 attributes and a coding system given by 2 bits equivalent to three linguistic labels, the number of variables used in the truth table for the QM method is equal to 13x2, giving a total of 26 variables. The computational cost increases as the number of attributes or the number of linguistic labels increases.

In summary, the rule base minimization method gives a good result with the databases used, managing to reduce the fuzzy rule set while maintaining the correctness.

V. CONCLUSIONS

The method proposed in the paper achieves the goal of reducing the initial set of fuzzy rules significantly without sacrificing the accuracy of the original fuzzy rule set. This reduction allows to improve the simplicity of the model, as well as having advantages in the efficiency of the associated inference processes.

One possible interpretation of the process is that we have been able to compact in some way the information contained in the initial set of fuzzy rules. In any case, the proposed model needs future improvements. First of all, it is necessary to revise the binary coding of the fuzzy rules since the defined model does not allow to completely eliminate variables from the rule, which is one of the great advantages of the extended fuzzy rules model. Secondly, although the QM method is an improvement over the initial model of Karnaugh's maps, the algorithm is still extremely complex, and this complexity grows exponentially with the number of variables required.

REFERENCES

[1] M. J. Gacto, R. Alcalá, and F. Herrera, "Interpretability of linguistic fuzzy rule-based systems: An overview of interpretability measures," Information Sciences, vol. 181, no. 20, pp. 4340–4360, Oct. 2011, doi: 10.1016/j.ins.2011.02.021.

[2] Z. Chi, H. Yan and T. Pham, T, "Fuzzy algorithms with applications to image processing and pattern recognition". World Scientific, 1996.

[3] P. W. C. Prasad, A. Beg, and A. K. Singh, "Effect of Quine-McCluskey simplification on Boolean space complexity," in 2009 Innovative Technologies in Intelligent Systems and Industrial Applications, Kuala Lumpur, Malaysia, 2009, pp. 165–170, doi: 10.1109/CITISIA.2009.5224219.

[4] D. García, A. González, and R. Pérez, "Overview of the SLAVE learning algorithm: A review of its evolution and prospects," International Journal of Computational Intelligence Systems, vol. 7, no. 6, pp. 1194–1221, Nov. 2014, doi: 10.1080/18756891.2014.967008.

[5] C.-C. Hung and B. Fernandez, "Minimizing rules of fuzzy logic system by using a systematic approach," in [Proceedings 1993] Second IEEE International Conference on Fuzzy Systems, San Francisco, CA, USA, 1993, pp. 38–44, doi: 10.1109/FUZZY.1993.327466.

[6] M. Karnaugh, "The map method for synthesis of combinational logic circuits," Transactions of the American Institute of Electrical Engineers, Part I: Communication and Electronics, vol. 72, no. 5, pp. 593–599, Nov. 1953, doi: 10.1109/TCE.1953.6371932.

[7] C. Mencar, C. Castiello, R. Cannone, and A. M. Fanelli, "Design of fuzzy rule-based classifiers with semantic cointension,"Information Sciences, vol. 181, no. 20, pp. 4361–4377, Oct. 2011, doi: 10.1016/j.ins.2011.02.014.

[8] R.K. Brayton, G.D. Hachtel, C. McMullen, A. Sangiovanni-Vincentelli, Logic Minimization Algorithms for VLSI Synthesis, Kluwer Academic Publishers Group, 1984.

[9] M. Elkano, M. Galar, J. Sanz, and H. Bustince, "CHI-BD: A fuzzy rule-based classification system for Big Data classification problems," Fuzzy Sets and Systems, vol. 348, pp. 75–101, Oct. 2018, doi: 10.1016/j.fss.2017.07.003.

[10] H. Ishibuchi and T. Yamamoto, "Rule weight specification in fuzzy rule-based classification systems," IEEE Trans. Fuzzy Syst., vol. 13, no. 4, pp. 428–435, Aug. 2005, doi: 10.1109/TFUZZ.2004.841738.

[11] L.-X. Wang and J. M. Mendel, "Generating Fuzzy Rules by Learning from Examples," vol. 22, no. 6, p. 14, 1992.

[12] D. Alvarez-Estevez and V. Moret-Bonillo, "Revisiting the Wang–Mendel algorithm for fuzzy classification," Expert Systems, vol. 35, no. 4, p. e12268, 2018, doi: 10.1111/exsy.12268.

[13] D. Dua, and C. Graff, UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science, 2019.