Hybrid Fuzzy Classification Algorithm with Modifed Initialization and Crossover

Tatiana Pleshkova[a] and Vladimir Stanovov[b]
Siberian Federal University, Krasnoyarsk, Russian Federation

Keywords: Classification, Fuzzy Logic, Genetic Algorithm.

Abstract: The article proposes two modifications of initialization and crossover operations for the design of a genetic fuzzy system. A fuzzy logic system is used to solve data classification problems and is automatically generated by a genetic algorithm. The paper uses a genetic algorithm to encode of several fuzzy granulations into a single rule, while each individual encodes a rule base. The proposed algorithm uses several training objects of the same class to create a single rule during initialization. The modified crossover creates a new rule base from most efficient rules selected from parents. To evaluate the effectiveness of the modification, the computational experiments were carried out on several datasets, followed by verification using Mann-Whitney U test. The proposed initialization modification allows reducing the number of rules in a fuzzy rule base and increasing the accuracy and F-score on some datasets. The crossover modification shows higher efficiency only on one dataset.

1 INTRODUCTION

In the modern world, data analysis is a necessary process in all spheres. One of the popular methods of knowledge systematization is classification, which implies the organization of data into categories for effective use. Various tasks are solved with classification methods, for example, the classification of data from satellite images or medical diagnosis and prognosis of diseases (Ştefan, 2012). There is no regular method for solving data classification problems. Depending on the formulated conditions, it is necessary to select a method that would classify the data into the appropriate categories. There are many methods for solving data classification problems that can be divided into two groups, based on “black box” and “white box” principles (Komargodski et al., 2017). Depending on the problem, a suitable method is selected before starting the classification. First of all, due to the fact that the user of the system that classifies data does not always have the opportunity to understand the decision-making process, there is a need to create a system that allows the classification process to be performed while getting information about which characteristics influence the classifications. In other words, the users’ involvement is aimed at studying the decision-making process, which gives them the opportunity to use their knowledge of the subject area to increase the reliability of data classification. To fulfill these conditions, it is necessary to use the “white box” model when implementing the data classification method. If the solver type (neural network and so on) is not important, then it makes sense to use the data classification method without providing information about which characteristics influence the classification results.

In this study it was decided to use Fuzzy Rule-Based System due to its flexibility and interpretability. Many researchers use Fuzzy Rule-Based System to implement the “white box” model (Bodenhofer and Herrera, 1997). L. Zadeh proposed a fuzzy logic in 1965 that was able to cope with the task of describing vague definitions of human language (Zadeh, 1965). And in 1974, I. Mamdani designed the first functioning controller for a steam turbine (Mamdani, 1974), which operated based on the L. Zadeh algebra. Modern researchers also continue to develop this area and have achieved great success. For example, Y. Nojima, H. Ishibuchi and F. Herrera are doing research with the classification of objects based on fuzzy rules. In the current study, it was decided to use the method discussed in the article (Ishibuchi et al., 2013) as baseline, where H. Ishibuchi proposed a machine learn-

[a] https://orcid.org/0000-0001-7761-7808
[b] https://orcid.org/0000-0002-1695-5798

Pleshkova, T. and Stanovov, V.
Hybrid Fuzzy Classification Algorithm with Modifed Initialization and Crossover.
DOI: 10.5220/0011587500003332
In *Proceedings of the 14th International Joint Conference on Computational Intelligence (IJCCI 2022)*, pages 225-231
ISBN: 978-989-758-611-8; ISSN: 2184-2825

ing algorithm based on fuzzy logic. Our goal is to improve the quality of fuzzy classifiers by proposing new genetic operations, to make the algorithm work more efficiently, that is, to find better solutions faster, but with the same level of complexity. In particular, the contributions of this study are: initializing rules from several objects is more efficient and selecting better rules during crossover may improve performance.

This paper is organized as follows: Section II contains related works. In Section III we briefly explain the basic method and its features. Next we explain proposed modifications for initialization and crossover in Section IV. In Section V we present the results. Finally we conclude this paper in Section VI.

2 RELATED WORKS

Genetic fuzzy systems have attracted considerable attention in the artificial intelligence community in the last decades. Since 2000 there was a growing need to find a compromise between interpretability and accuracy in the tasks of linguistic fuzzy modeling. First of all in (Ishibuchi and Nakashima, 2000) authors introduce the effective use of rule weights in fuzzy rule-based classification systems. In (Alcalá et al., 2007) a new post-processing method was introduced. This method was based on the well-known SPEA2 algorithm. In (Fernández et al., 2010) evolutionary approaches that help to search for a set of rules were considered. The hybrid fuzzy genetics-based machine learning (GBML) algorithm was proposed in (Ishibuchi et al., 2013), where algorithm had a Pittsburg-style framework in which a rule set is handled as an individual. The operation of the algorithm will be described in more detail in the next section.

Each of the works uses the representation of the decision-making process in the language understandable for the expert. It is possible thanks to the use of fuzzy rules in the classification, which determine whether an object with known characteristics belongs to a particular class. A fuzzy rule consists of a condition of the type “if... then...” with fuzzy terms in the“if...” part and the corresponding class number in the “then...” part (Herrera and Magdalena, 1997).

$$\text{Rule } R_n : \text{if } x_1 \text{ is } L_{q1} \text{ and...and } x_v \text{ is } L_{qv} \quad \text{then Class } C_q \text{ with } CF_q, \quad (1)$$

where n – number of rules in the rule base, v – number of variables in the data sample, L – this is a linguistic term, C – class label, CF – rule weight (which is a real number in the unit interval $[0,1]$).

3 GENETICS-BASED MACHINE LEARNING

The method in (Ishibuchi et al., 2013) is based on the search for the optimal rule base. This search is carried out using evolutionary algorithms, where the main idea is based on Charles Darwin’s theory of natural selection (Bleckmann, 2006). The process of finding the best solution begins with a set of individuals, i.e. a population. The process of natural selection begins with the choice of the individuals with better fitness from the population. The selected individuals produce an offspring that inherits the characteristics of the parents. If the parents are better i.e. have a higher fitness than the others, there is an opportunity that their offspring will be better than the parents and will have a higher chance of survival. This process continues to be repeated for a certain number of generations, and at the end of the generation the fittest individual is found (Mitchell, 1996). One individual consists of n fuzzy rules, where upper limit is $n \leq 50$. Each rule is designed using linguistic terms $L_1, L_2, ..., L_{14}$. One of the features of study (Ishibuchi et al., 2013) is the use of several fuzzy granulations for each linguistic variable. Figure 1 shows this concept. There are 14 linguistic variables and a “don’t care” condition (DC), which means that for this variable in this rule there is no difference what value the variable has. This is described in (Ishibuchi et al., 2013).

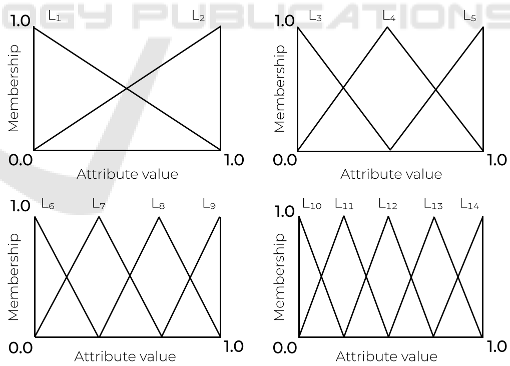

Figure 1: Fuzzy granulations ($L_1, L_2,...,L_{14}$).

After that, using the fitness function, the fitness of the individual is determined, i.e. how well the fuzzy rules base performs classification. After receiving fitness function value of each individual, we can determine the probability of choosing a particular individual for following reproduction. The genetic algorithm, which is one of the variations of the evolutionary algorithm and is used in the study, consists of several stages: initialization, selection, crossover, mutation and formation of a new generation (Banzhaf et al., 1998). The feature of the Ishibuchi algorithm is that after the mutation, the Michigan-style part is applied.

The first stage is initialization, in which individuals are formed. When creating a rule, a random object is first selected from the training sample, then a rule is formed based on the parameters of the selected object, that is, the corresponding linguistic variable is selected for each parameter. Then the confidence is calculated and applied in evaluating the rule class number, which is the most corresponding to a particular rule. To specify the consequent class C_q and the rule weight CF_q, we first calculate the confidence of the association from the antecedent fuzzy vector L_q to each class k ($k = 1,2,\dots,M$) (Ishibuchi et al., 2013) as follows:

$$Conf(L_q \Rightarrow \text{Class } k) = \frac{\sum_{x_p \in \text{ Class } k} \mu_{L_q}(x_p)}{\sum_{p=1}^{m} \mu_{L_q}(x_p)} \tag{2}$$

where μ_{L_q} is the product of all memberships values of the antecedent fuzzy set L_{qi} at the input value x_{pi}.

If the confidence value for a particular class C_q is larger than 0.5 in eq. 2 in (Ishibuchi et al., 2013) computational experiments, we generate a fuzzy rule with the antecedent fuzzy rule and the consequent class C_q. Then the rule weight CF_q in (Ishibuchi et al., 2013) is specified as follows:

$$CF_q = 2 * Conf(L_q \Rightarrow \text{Class } C_q) - 1. \tag{3}$$

It is established that if $CF_q > 0$, then the rule is added to the fuzzy rule base. Fuzzy rules with negative weights are not applicable for classification. One of the advantages of this work is the use of a Michigan-style algorithm probabilistically applied to each set of rules. The Michigan-style part is used as local search after main genetic algorithm.

The fitness of the main genetic algorithm was calculated as shown in eq. 4, and the fitness of the Michigan-style part is the fitness of the individual rules at the selection stage, which was calculated based on the number of correctly classified objects.

$$fitness = w_1 * f_1 + w_2 * f_2 + w_3 * f_3 \tag{4}$$

where $w_1 = 100$, $w_2 = 1$, $w_3 = 1$ and f_1 – the error rate on training patterns in percentage, f_2 – the number of active fuzzy rules (rules that classified at least 1 object correctly), f_3 – the length of the rule (the number of variables other than 0 - don't care).

The second stage is selection, in which from the entire population, individuals are selected with the same probability in an amount equal to the size of the tournament, that is, the number pre-set by the user, the number must necessarily be greater than 2 and less than the population size. After the tournament is performed, the individual with the highest fitness in this group is selected (Banzhaf et al., 1998).

The selected individuals enter the next stage – crossover. Offspring is the random combination of the genes (fuzzy rules) of two individuals. Two individuals are selected with tournament in the algorithm, and the number of active rules for each of them is calculated. Then an offspring is formed from these individuals (Poli et al., 2008). The offspring should not have the number of active rules less than the number of classes.

The last stage is mutation, where certain parts of the rule base change in accordance with a given formula:

$$min(\frac{3}{v * f_2}, 1), \tag{5}$$

where v is the number of variables in the dataset.

In the next section the proposed modifications is described.

4 MODIFICATION

Due to the fact that the rule describes several objects, the following modification was proposed. Instead of forming a fuzzy rule on one random object, the use of several objects of the same class to create a rule was implemented. To form a rule, a random object a is first selected, then several more objects b_k of the same class are selected, where $k = 1 \dots r$ and r is a number of neighbors. The search is carried out for the nearest several objects to the selected one using the Euclidean distance, as shown in Figure 2. For example, based on three objects a, b_{k1}, b_{k2} a fuzzy rule is generated. The analysis of the best combination of the number of objects to form the rule was also performed.

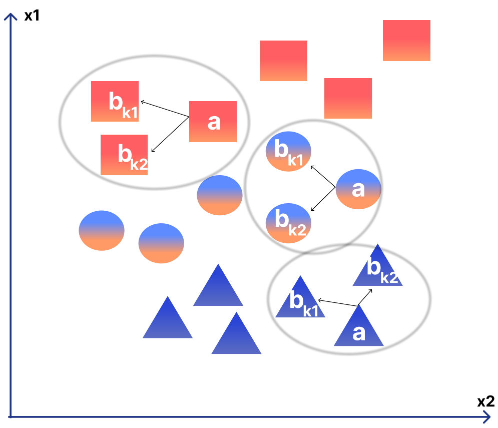

Figure 2: Visualization of fuzzy rules generation modification.

As for crossover modification, we proposed to change the choice of rules for the formation of an offspring. The new method selects the most suitable rules, then forms an offspring from these rules that have the largest number of correctly classified objects. An example is shown in Figure 3.

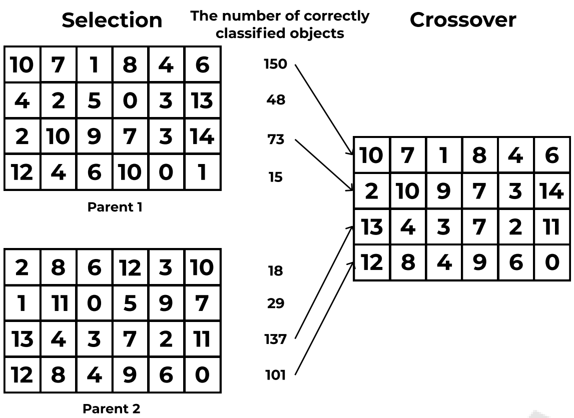

Figure 3: Visualization of crossover modification.

We also made additional changes in the algorithm. For instance, we used F-score in fitness of the main genetic algorithm. As we said before, one of the features in (Ishibuchi et al., 2013) is the Michigan-style part. Depending on the number of active rules of an individual, the number of rules that will be formed heuristically or genetically was calculated using formula with one addition: if the value is 1, then the number of rules generated by the genetic algorithm is 2, heuristic is 2. Heuristically, a rule was formed for an object that was not correctly classified by any rule. In addition, we set the minimum threshold parameter for confidence equal to 0.6. In our experiment we checked the results of weak, average and strong mutation and we came to the conclusion that average mutation has better effect on accuracy.

5 RESULTS

Testing of the basic and the modified algorithms was carried on several tasks taken from the UCI repository (Asuncion, 2007). The algorithm was run with the number of individuals set to 100.

In the first experiment with selecting the best number of objects we used three tasks and 500 generations. In the second experiment we used four tasks and we decided to increase the number of generations to 1000 in order to check whether improvements are noticeable when the number of generations is doubled.

The 10-fold cross-validation procedure was iterated three times using different data partitions into ten subsets. Average results over 30 runs are summarized in Table 1, Table 3 and Table 5, *NR* is the number of rules. Since the average values are presented, in order to evaluate the efficiency of the modification, it is necessary to use the Mann-Whitney U test (Mann and Whitney, 1947), popular nonparametric test to compare outcomes between two independent groups. Table 2, Table 4 and Table 6 also show the results of a statistical test accordingly, where the symbol "=" shows that differences are insignificant, the symbol "+" shows that differences are significant and the modification is more efficient and "–" means that differences are significant and the modification is worse than the original algorithm.

Table 1: Results on phoneme dataset, first experiment.

Number of objects	Values		
	NR	*F-score*	*Accuracy*
1	10.267	0.722	0.792
2	12.060	0.705	0.783
2 of 5	6.200	0.745	0.801
3	10.367	0.721	0.782
3 of 6	4.367	0.774	0.807
4 of 6	4.900	0.775	0.811
5 of 7	7.300	0.776	0.808
6	10.833	0.709	0.789

Table 2: Results Mann-Whitney U-test on phoneme dataset.

Number of objects	Values		
	NR	*F-score*	*Accuracy*
2	–	=	=
2 of 5	+	=	=
3	=	=	=
3 of 6	+	+	+
4 of 6	+	+	+
5 of 7	+	+	+
6	=	=	=

Table 3: Results on ring dataset, first experiment.

Number of objects	Values		
	NR	*F-score*	*Accuracy*
1	20.107	0.831	0.831
2	20.300	0.825	0.825
2 of 5	20.100	0.832	0.834
3	20.000	0.829	0.829
3 of 6	18.000	0.832	0.833
4 of 6	19.900	0.828	0.829
5 of 7	19.333	0.829	0.829
6	19.533	0.830	0.831

Table 4: Results Mann-Whitney U-test on ring dataset.

Number of objects	Values		
	NR	*F-score*	*Accuracy*
2	=	=	=
2 of 5	=	=	=
3	=	=	=
3 of 6	+	=	=
4 of 6	+	=	=
5 of 7	=	=	=
6	+	=	=

Table 5: Results on segment dataset, first experiment.

Number of objects	Values		
	NR	*F-score*	*Accuracy*
1	22	0.903	0.903
2	22.567	0.898	0.899
2 of 5	22.600	0.904	0.905
3	21.767	0.903	0.904
3 of 6	17.933	0.905	0.905
4 of 6	22.700	0.907	0.908
5 of 7	21.100	0.903	0.903
6	21.567	0.907	0.907

Based on Tables 1, 3, 5, it can be seen that the modification with the design of a fuzzy rule on nearby objects works more efficiently than on randomly selected ones from the same class, regardless of their number. For the second experiment, it was decided to use 3 nearest objects selected from 6 objects of the same class to form the rule as shown in Figure 3. It can be seen from the results of the modification that the number of rules decreases, which has a positive effect on interpretation, because the fewer rules, the easier the rule base. The results are presented in Table 7 and results from Mann-Whitney U-test is in Table 8. Comparing the results of the first and second experiments, it can be seen that the changes are insignificant, which means there is no need to increase the number of generations for the data used in the experiment. The algorithm finds a high-quality rule base in 500 generations, and therefore in less time.

Table 6: Results Mann-Whitney U-test on segment dataset.

Number of objects	Values		
	NR	*F-score*	*Accuracy*
2	–	=	=
2 of 5	–	=	=
3	=	=	=
3 of 6	+	=	=
4 of 6	=	=	=
5 of 7	=	=	=
6	=	=	=

Table 7: Results of second experiment.

Number of objects	Values		
	NR	*F-score*	*Accuracy*
Phoneme			
1	10.833	0.721	0.789
3 of 6	4.933	0.766	0.825
Ring			
1	20.567	0.832	0.832
3 of 6	18.533	0.828	0.828
Satimage			
1	22.700	0.806	0.863
3 of 6	16.833	0.837	0.885
Page-blocks			
1	9.900	0.435	0.909
3 of 6	9.000	0.440	0.910

Table 8: Results Mann-Whitney U-test of second experiment.

Number of objects	Values		
	NR	*F-score*	*Accuracy*
Phoneme			
3 of 6	+	+	+
Ring			
3 of 6	+	=	=
Satimage			
3 of 6	+	+	+
Page-blocks			
3 of 6	+	=	+

Based on Table 7, it can be noted that the proposed method of generating fuzzy rules allows the modified algorithm to significantly reduce the number of rules and at the same time improve the accuracy for some data classification problems.

The results of crossover modifications and Mann-Whitney U test are presented in Table 9 and Table 10.

Table 9: Crossover modification.

Data set	Values		
	NR	*F-score*	*Accuracy*
Phoneme	11.400	0.731	0.786
Ring	23.133	0.833	0.834
Satimage	17.100	0.833	0.858

Based on the results in Tables 9-10, we can conclude that crossover modification shows higher efficiency only on one dataset “Satimage”. That means that there is a possibility that the modification may work in some cases. The difference between “Satimage” and other datasets is the number of classes and range of numbers. Dataset “Satimage” has positive numbers, others have both (positive and negative numbers). The main contributions of this paper

Table 10: Results Mann-Whitney U-test for crossover modification.

Data set	Values		
	NR	*F-score*	*Accuracy*
Phoneme	=	=	=
Ring	–	=	=
Satimage	+	+	+

are: 1) it is shown that initialization on several objects works better than on one; 2) it is shown that modified crossing can work better on some data. It is necessary to conduct additional experiments on other datasets to find out what conditions influence the efficiency of modified crossover.

We decided to test both modifications on the same data in order to check if there will be improvements. The results are presented in Table 11.

Table 11: Results of both modifications.

Data set	Values		
	NR	*F-score*	*Accuracy*
Phoneme	5.833	0.747	0.797
Ring	23.200	0.833	0.835
Satimage	14.667	0.832	0.861
Segment	21.852	0.899	0.896

As we can see from the results of both modifications, the differences are insignificant. In future experiments, it will be necessary to try changing the parameters to test the efficiency.

The modified method was also compared with alternative approaches: Decision Tree (DT), Support Vector Machine (SVM), Logistic Regression (LR), Neural Networks (NN). These methods were taken from sklearn library, the standard parameters were used. The accuracy are presented at Table 12.

Table 12: Results of alternative approaches.

Data set	Methods			
	DT	*SVM*	*LR*	*NN*
Phoneme	0.770	0.774	0.774	0.738
Ring	0.737	0.726	0.721	0.762
Satimage	0.772	0.869	0.867	0.789
Page-blocks	0.960	0.967	0.967	0.944
Segment	0.541	0.961	0.962	0.913

As can be seen from the table, the proposed modification with initialization is comparable classification quality to other known methods. For example, on the “Satimage” and “Ring” tasks, it proved to be better than neural networks and on the “Phoneme” task the proposed modification showed better results comparing to others.

6 CONCLUSIONS

In this paper, the search for better way to find accurate and compact rule bases was investigated. In this regard, the modification with initialization was proposed in this paper, and showed efficiency on several datasets, decreasing number of rules and increasing accuracy and F-score. The design of a rule on several nearby objects from the same class showed higher efficiency compared to using a single object. As for the modification with crossover, the improved efficiency of this approach was observed only on one dataset. The proposed method shows similar accuracy compared to alternative methods. In future studies, it is possible to consider other modifications to improve the quality of the algorithm. For instance, we want to implement multi-objective optimization.

ACKNOWLEDGEMENTS

The work was carried out within the framework of the state support program for leading scientific schools (grant of the President of the Russian Federation NSh-421.2022.4).

REFERENCES

Alcalá, R., Gacto, M. J., Herrera, F., and Alcalá-Fdez, J. (2007). A multi-objective genetic algorithm for tuning and rule selection to obtain accurate and compact linguistic fuzzy rule-based systems. *Int. J. Uncertain. Fuzziness Knowl. Based Syst.*, 15:539–557.

Asuncion, A. U. (2007). Uci machine learning repository, university of california, irvine, school of information and computer sciences.

Banzhaf, W., Francone, F. D., Keller, R. E., and Nordin, P. (1998). Genetic programming - an introduction: On the automatic evolution of computer programs and its applications.

Bleckmann, C. A. (2006). Evolution and creationism in science: 1880–2000.

Bodenhofer, U. and Herrera, F. (1997). Ten lectures on genetic fuzzy systems.

Fernández, A., García, S., Luengo, J., Bernadó-Mansilla, E., and Herrera, F. (2010). Genetics-based machine learning for rule induction: State of the art, taxonomy, and comparative study. *IEEE Transactions on Evolutionary Computation*, 14:913–941.

Herrera, A. and Magdalena, L. (1997). Genetic fuzzy systems : A tutorial.

Ishibuchi, H., Mihara, S., and Nojima, Y. (2013). Parallel distributed hybrid fuzzy gbml models with rule set migration and training data rotation. *IEEE Transactions on Fuzzy Systems*, 21:355–368.

Ishibuchi, H. and Nakashima, T. (2000). Effect of rule weights in fuzzy rule-based classification systems. *Ninth IEEE International Conference on Fuzzy Systems. FUZZ- IEEE 2000 (Cat. No.00CH37063)*, 1:59–64 vol.1.

Komargodski, I., Naor, M., and Yogev, E. (2017). White-box vs. black-box complexity of search problems: Ramsey and graph property testing. *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 622–632.

Mamdani, E. H. (1974). Applications of fuzzy algorithms for control of a simple dynamic plant. *Proceedings of the IEEE*.

Mann, H. B. and Whitney, D. R. (1947). On a test of whether one of two random variables is stochastically larger than the other. *Annals of Mathematical Statistics*, 18:50–60.

Mitchell, M. (1996). An introduction to genetic algorithms.

Poli, R., Langdon, W. B., and McPhee, N. F. (2008). A field guide to genetic programming.

Ştefan, R.-M. (2012). A comparison of data classification methods. *Procedia. Economics and finance*, 3:420–425.

Zadeh, L. A. (1965). Fuzzy sets. *Inf. Control.*, 8:338–353.