

# Brain storm optimization algorithm for data classification

Zhifeng Zhang<sup>1</sup>, Shaolin Zhu<sup>1</sup>, Tianqi Li<sup>1</sup> and Baohuan Li<sup>1</sup>

<sup>1</sup> Zhengzhou University of Light Industry, Zhengzhou 453300, China

1156166091@qq.com

**Abstract:** With the increasing of the number of dimensions or variables in the search space, the inductive learning of fuzzy rule classifier will be influenced by the generation and optimization of rules. Thus, the extensibility and accuracy of fuzzy systems will be affected. In this paper, the brain storm optimization algorithm was used. A new fuzzy system was designed by modifying the rules definition process in traditional fuzzy system. In the derivation of rules, the exponential model was introduced to improve the traditional brain storming algorithm. On the basis, this new fuzzy system was used for the research on data classification. The experimental results show that this new fuzzy system can improve the accuracy of data classification.

**Key words:** Brain storm optimization; Fuzzy system; Data classification

## 1 Introduction

At present, computational intelligence has become a popular technology to solve complex problems [1]. The most commonly used standard method is called hybrid computational intelligence method [1], including neural networks, decision trees and reasoning systems based on fuzzy rules. In the field of machine learning, the use of a classification system based on fuzzy rules can provide end users with a more reasonable semantic representation. Classification systems based on fuzzy rules are widely used to solve various practical problems, such as abnormal intrusion detection, image processing, and medical applications. Most classification systems based on fuzzy rules need to involve high dimensions or multiple variables. This situation leads to a rapid increase in the search space of fuzzy rules, thereby affecting the performance and efficiency of the classification system.

At present, the methods for generating and learning fuzzy classification rules based on data space mainly include simple heuristic procedures, fuzzy neural technology, clustering methods and genetic algorithms, among which heuristic and meta-heuristic algorithms include particle swarm optimization, simulated annealing, and firefly, Artificial bee colony optimization and a series of biological and physical systems formed by behaviour. On this basis, literature [11] proposed a hybrid fuzzy method to extract a compact rule base. Since this method only represents the rules in the genetic population, it does not explicitly model the fuzzy system. In addition, when the traditional fuzzy system rule generation method increases rapidly in the search space or variables, the convergence speed of the model will be extremely slow.

Although the brainstorming algorithm has been effectively applied in the fuzzy system, the traditional brainstorming algorithm cannot effectively solve the problem of poor accuracy and slow convergence caused by the rapid increase of spatial dimensions or variables. When the dimensions of the search space or variables are the rapid increase, traditional methods are difficult to effectively solve this type of problem. This article mainly focuses on the problem caused by the sharp increase in the search space and the dimensionality of the variables when the training samples are large, and a brainstorming optimization algorithm is proposed. To realize a fuzzy system for data classification, the fuzzy system is divided into two stages. In the first stage, the uniform distribution is used to design the membership function; in the second stage, the brainstorming algorithm is optimized and improved by using the exponential average model. On this basis, by introducing different constraints (such as rule length, number of rules and class distribution), generate optimal inference rules, and use these rules to classify the data. By generating the optimal rules, the search space can be effectively reduced and efficiency can be improved.

## 2 Related work

In the fuzzy expert system of this paper, a set of if-then rules and membership functions are usually used to construct a rule base. Since membership functions and rule sets are interdependent in fuzzy systems, both should be constructed and set at the same time [2]. Based on if-then rules, create a fuzzy relationship between input and output [3]. If the rules are obtained from the training data space, then the construction of the rule base is transformed into a search problem in a high-dimensional space.

There are several methods for generating and learning fuzzy classification rules from the data space, including simple heuristic procedures, fuzzy neural technology [4], clustering methods [5] and genetic algorithms [6], among which heuristic and meta-heuristic Algorithms include particle swarm optimization [7], simulated annealing [8], firefly [9], artificial bee colony optimization [10] and a series of biological and physical systems formed by behaviour. In addition, in the above algorithms, the fine-tuning of the parameters will affect the convergence speed of the optimization process [12]. The purpose and characteristics of current research on fuzzy methods are shown in Table 1.

Table 1 Comparison of related research on fuzzy methods

References	Algorithm	Purpose	Disadvantages
[13]	Adaptive genetic rule extraction algorithm	Rule generation and optimization	The design of the member function is relatively simple
[14]	Genetic algorithm	Rule optimization	Not suitable for multi-class problems
[15]	Hybrid ant bee algorithm	membership function design	not suitable for continuous search
[16]	Genetic group algorithm	Adjust the optimal rule set and membership function	Genetic algorithm is too sensitive to global search
[17]	Genetic algorithm	Rule extraction	Need to optimize the design of member functions
[18]	Genetic algorithm	Rule extraction	Need to weight the rules
[19]	Ant colony optimization	Rule extraction	Must require multiple goals in fuzzy design volume_up content_copy share

### 3 Problem analysis

The rule base is the core of the fuzzy system. It has a set of rules defined for data classification. Two issues should be paid attention to when defining rules.

a) How to get the rules? The rule generation process is a kind of search problem. In the big data space, accurate rules suitable for fuzzy systems should be generated, and should conform to the format of fuzzy rules.

b) How to optimize the rules? Two aspects need to be considered here, namely the length of the rule and the number of rules. The number of attributes contained in the rules and the number of rules in the rule base will directly affect the accuracy of inference.

Currently, fuzzy systems are widely used in classification, because compared with other classifiers (such as neural networks and support vector machines, it has the advantages of providing flexibility and avoiding learning time. Although fuzzy systems have good applications, but in the design process It is found that even if historical data is available, it is difficult to cover all domain knowledge in the design of the rule base. In view of this situation, this article hopes to design a new method to solve the excessive dependence on the manual intervention of domain experts in the design and construction of the rule base The status quo.

### 4 Fuzzy system for data classification

In view of the description of the fuzzy system in the previous section, this paper designs a data classification-oriented fuzzy system based on the brainstorming optimization algorithm by modifying the definition process of the rules in the traditional fuzzy system.

#### 4.1 Use uniform distribution to design membership functions

This section mainly constructs fuzzy and non-fuzzy membership functions. The reason for using uniform distribution is that the probability that a uniformly distributed random variable falls within any interval of a fixed length has nothing to do with the position of the interval itself (but depends on the size of the interval), as long as the interval is included in the support of the distribution, which makes the design The membership function can be easily defined for the rule length and the number of rules and is not affected by the position. The membership function bar indicates the possibility of mapping the data value to a membership value (or membership) between 0 and 1. It should be noted that the membership function here is used to determine the probability value used in the rule. The membership function contains three important steps: i) select the type of membership function, ii) determine the number of membership functions for each attribute, and iii) define The interval of each membership function. Assuming that the input database can be expressed as:

$$D = \{X_{ij}, 0 < i, j < m, n\} \quad (1)$$

Where  $m$  is the number of attributes and  $n$  is the number of data objects. This attribute is represented as:

$$A_i = 0 < i < m \quad (2)$$

Each attribute should be defined with  $p$  member functions and each member function should be defined with three interval variables, such as  $a$ ,  $b$ ,  $c$ . Among them,  $a$  refers to the lower boundary,  $c$  defines the upper boundary, and the degree of membership is zero.  $b$  refers to a center with a membership degree.

In this article, it is assumed that the type of membership function is ternary membership function, which is a very simple and easy to define interval. The membership function ( $p$ ) of fuzzy system design is 3. Next, define interval variables ( $a$ ,  $b$ , and  $c$ ) for each member to set up automatic functions. For example, for an attribute  $A_i$ , the number of member function designs ( $p$ ) is three that have been taken.

The following equation (3) shows that the attribute vector is provided to the membership function to define the number of  $p$ .

$$A_i \leftrightarrow \mu_{ik}; 0 < k < p \quad (3)$$

Here, each membership function is defined as three interval variables, such as  $a$ ,  $b$ , and  $c$ . Therefore, the input data vector of the attribute  $A_i$  is divided into  $p$  intervals, and then each interval is divided into  $p$  intervals again. Then, from each interval, the minimum, center, and maximum boundaries are calculated based on (4)(5).

$$\mu_{ik} = f(X_i; a, b, c) \quad (4)$$

$$f(X_i; a, b, c) = \left[ X_i^{max} \frac{X_i^{max} + X_i^{min}}{2} X_i^{min} \right] \quad (5)$$

Sample ternary membership functions are given in Figure 1, where two membership functions are used.

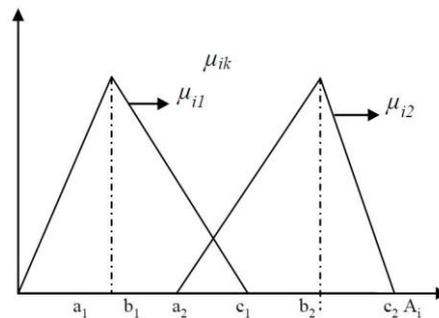


Figure 1 Sample membership function

#### 4.2 Rule generation and optimization based on storm optimization algorithm

Rule generation and optimization is the second important step of fuzzy systems. Here, the existing brainstorming optimization algorithm is improved by using the exponential moving average model (EWMA) [13]. The new algorithm aims to generate random rules and uses three constraints such as rule length, rule number and class distribution to select the optimal rule.

##### (A) Overall framework

In the improved brainstorming optimization algorithm, the solution is a set of rules defined by a fuzzy system. This rule is determined by the membership function in section 4.1. If the number of membership functions for each attribute of the data is determined, the number  $r$  of that rule is also determined, and the number of  $r$  is equal to the number of membership functions. Each scheme is

coded as shown in Figure 2. The size of the program scale is  $1+m*r$ . Among them,  $m$  is the length of each rule and  $r$  is the number of rules. In Figure 2, the rule is defined as [1 2 1 1], which means that the last column specifies whether it is an OR method or an AND method, and the penultimate element refers to the output class, and the rest refers to the input variable.

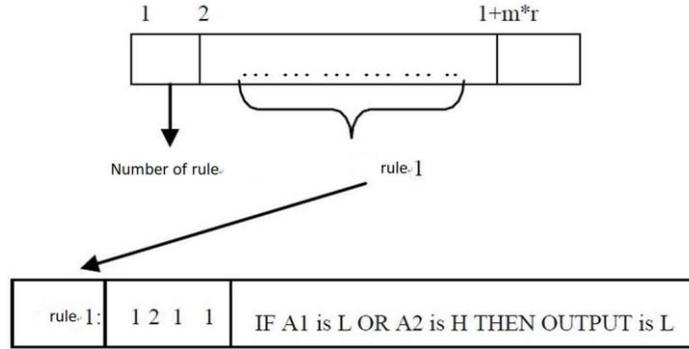


Figure 2 A sample of solving method

### (B) Fitness function

Each solution representation needs to give a method to evaluate the fitness of the solution. Through the fitness function, the number and distribution of rules can be determined globally optimally, and the optimal solution can be obtained.

According to the problem definition, three factors should be considered: the length of the rule, the distribution and scale of the rule, and the matching of the rule in the database. Therefore, the fitness function designed in this paper includes the maximization function of the above three factors. The calculation method is as follows:

$$F = \alpha f_1 + \beta f_2 + \gamma f_3 \quad (6)$$

Among them  $\alpha, \beta, \gamma$  are weighting constants,  $f_1, f_2, f_3$  are the objective functions. First calculate the objective value by looking up the length of the rule. Take the length of each rule as the summation, and separate it from the number of rules and attributes to normalize it as an absolute value. Between 0 and 1. If  $f_1$  is 1, the complexity of the rules is smaller because there are fewer rules. Therefore, the desired performance is to obtain the maximum value in the first target, which is calculated as follows:

$$f_1 = 1 - \left[ \frac{\sum_{i=1}^r L_i}{r * m} \right] \quad (7)$$

Where  $L_i$  is the length of the rule and  $r$  is the number of rules. The second purpose is to find a match between the rules and the original database. In order to use the original database to find the matching count of each rule, a member function is used to input the database discretely, and the rule is directly matched with the database to find the flexibility of the rule. If the rule has more match counts, then the rule is desirable. Therefore, considering the matching count of each rule, and dividing the number of rules and the number of data objects by the normalization of variables between 0 and 1, the calculation method can be expressed by the following formula:

$$f_2 = \frac{\sum_{i=1}^r M(r_i, D)}{r * n} \quad (8)$$

Where  $M(r_i, D)$  is the matching count of the rule  $r_i$  and the database. The third goal is to reduce changes in rules from different classes. Therefore, the number of rules for each class is used as the sample data, and the variance is calculated. The variance is separated from the number of rules, and the data value is normalized to 0 to 1. The required performance is to maximize the third objective as shown in the following formula (9), where  $V(r, c)$  is the variance function of the rule count for each class.

$$f_3 = 1 - \frac{V(r, c)}{r} \quad (9)$$

### (C) The execution of brainstorming optimization algorithm

The traditional brainstorming optimization algorithm is modified with a mathematical theory called EWMA [13] to improve the solution search in the predefined search space. In the improved brainstorming optimization algorithm, the above ideas are expressed as solutions updated in each iteration. The advantage of this improved algorithm is that it can make better use of global information and further improve evolutionary diversity. The flow chart of the improved brainstorming optimization algorithm is shown in Figure 3.

Below, according to Figure 3, further description and analysis of the execution process of the algorithm:

Initialization: Assume that  $q$  schemes are randomly initialized in the search space as  $I_i = [I_{i1}, I_{i2}, \dots, I_{id}]$ , where  $i = 1, 2, \dots, q$ ,  $d$  represents the variable dimension used for optimization. During the initialization step,  $d$  is generated by the lower limit and upper limit of the variable.

Grouping and evaluation: After initialization, the schemes are grouped based on the k-means clustering algorithm, where  $k$  is the required number of clusters.

Update: Use three different probability values (such as  $a, b, c$  in section 4.2) to select clusters.

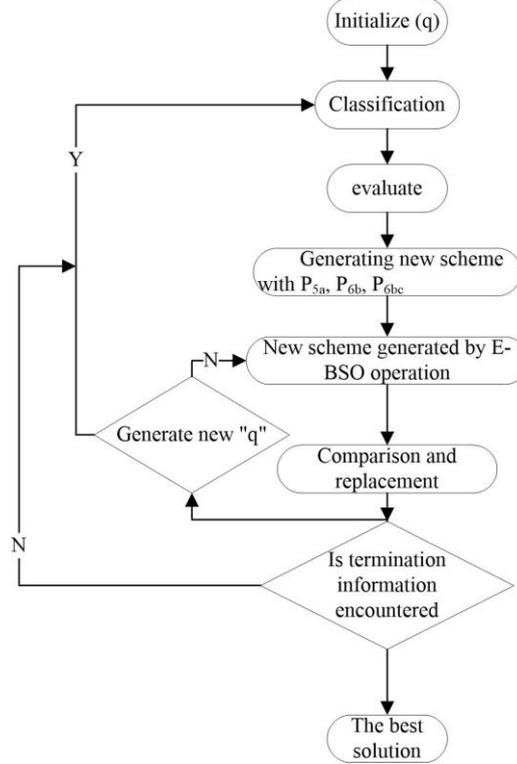


Figure 3 Improved brainstorming optimization algorithm flow chart

Among them, the candidate scheme based on 4.2.a is updated according to the center point of the cluster; the candidate scheme based on 4.2.b is updated according to the center point of the cluster and the added random value; the candidate scheme based on 4.2.b and 4.2.c is updated according to two. The average value of the center points of the clusters is updated, and the remaining plan can be updated using formula (10).

$$I_{t+1} = I_t + \xi N(\mu, \sigma) \quad (10)$$

In the algorithm proposed in this paper, the update strategy of EWMA is used to modify the scheme. The EWMA model and optimization algorithm combined with brainstorming, to further improve the search efficiency. According to the EWMA model, the project forecast can be expressed as:

$$E_{t+1} = e * I_t + (1 - e)E_t \quad (11)$$

Where  $e$  is a constant and  $E_{t+1}$  is the output scheme of the EWMA model in the next iteration.  $E_t$  is the output scheme of the previous iteration using the EWMA model. The above equation can be rewritten as:

$$I_t = \frac{1}{e}(E_{t+1} - (1 - e) * E_t) \quad (12)$$

$$I_{t+1} = \frac{1}{e}(E_{t+1} - (1 - e) * E_t) + \vartheta N(\mu, \sigma) \quad (13)$$

Where is the scheme generated from the previous iteration, and  $I_{t+1}$  is the newly generated scheme. It is a Gaussian random value with mean  $\mu$  and variance  $\sigma$ . It is an adjustment coefficient used to slow down the convergence rate as the evolution progresses.

$$\xi = s * \log \text{sig} \left( \frac{NC_{max}/2 - Ne}{K} \right) \quad (14)$$

Where  $s$  is a random value between 0 and 1. And  $NC_{max}$  indicate the maximum number of iterations and the current number of iterations. The parameter  $K$  is used to adjust the slope of the  $\text{logsig}()$  function. Conclusion: Repeat the above process until  $q$  plans are updated. At this point, the best solution is output and used as the best solution to the problem.

#### (D) Rule weight

Rule weighting is an important criterion to obtain better performance of the fuzzy classifier system. That is, the system contains  $r$  rules, and different rules have different degrees of ambiguity. Therefore, the weight distribution standard of the weighting rule is very important. This article uses two criteria, namely the matching of the rule with the database and the length of the rule. The formula used to weight each rule is

$$W(R_{opt}) = \frac{1}{2} * (f_1 + f_2) \quad (15)$$

## 5 experiments

Once the membership functions and rules are defined, the fuzzy system can classify the data. First, the data of the fuzzification process is directly given, and the test data is converted into the fuzzification value according to the definition of the membership function. Then, the non-fuzzy calculation is performed to obtain the fuzzy evaluation value obtained from the language value. Finally, the fuzzy evaluation value obtained is used to determine the class label of the input data.

According to the above ideas, this section evaluates the effect of the fuzzy system proposed in the article on data classification through experiments and analysis of the results.

### 5.1 Experimental setup

Data set description: For the experiment, the medical database PID was used. The database is taken from the UCI Machine Learning Library. The database contains details of Pima Indians with diabetes. The total number of instances is 768, and each instance contains 8 attributes.

Evaluation index: When comparing the performance of the fuzzy system proposed in this paper and the existing system, three criteria are introduced: sensitivity, specificity and accuracy. The definition of each metric is as follows:

$$Sensitivity = \frac{TP}{TP + FN} \quad (16)$$

$$Specificity = \frac{TN}{TN + FP} \quad (17)$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (18)$$

Among them, correct (TP) means correctly identified, false positive (FP) means incorrectly identified, true negative (TN) means correct rejection, and false negative (FN) means incorrectly rejected.

Use MATLAB (R2014a) to implement the fuzzy system proposed in the article. The operating environment is an i5 processor with a 2.2GHz CPU clock speed and a 64-bit Windows operating system with 4 GB of memory. Among them, the parameters needed in the improved brainstorming optimization algorithm are the constant  $e$  and the slope adjustment factor  $K$ . These parameters can be determined by analyzing the performance of different assignments and selecting the best parameter values for comparison. In the experimental link, the performance of the fuzzy system proposed in this paper is compared with the AGFS algorithm given in [13], which uses an adaptive genetic algorithm to design fuzzy rules.

### 5.2 Qualitative assessment

Table 2 shows the sample rules of PID obtained by the improved brainstorming optimization algorithm. Here 1, 2, and 3 represent VL, L and M in fuzzy membership. From the table, the first rule says that if A3 is L or A4 is L or A5 is L or A6 is VL, then it is classified as L. In addition, the second rule states that if A1 is L or A2 is VL, A3 is VL, A4 is VL, A5 is VL, or A6 is M, then it is classified as H. The following four rules are also specified in Table 2. These rules are generated by uniformly distributed membership functions. It can be clearly seen from Table 2 that the fuzzy rules generated by uniformly distributed membership functions are helpful for the judgment of patients with diabetes. Analyzing the reasons, we classify them as there is no order between the characteristics of patients. The uniform distribution allows the rules of each characteristic attribute of the patient to be generated independently. This helps improve the accuracy of detecting the presence of diabetes in patients. These rules ensure that the attributes of the disease in the rules are not expressed as "don't care" attributes. In order to further verify that the membership function based on uniform distribution can be helpful to the generation and optimization of fuzzy rules when the size of the search space or the variable increases rapidly under the condition of a large amount of data, the convergence speed of the classification problem can be guaranteed under the premise of high accuracy. However, the traditional method of

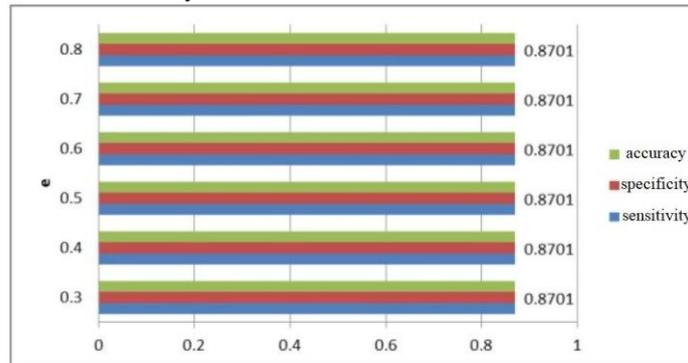
fuzzy rule generation does not take into account the problem of large convergence speed caused by the rapid increase of search space dimensions or the rapid increase of variables when the amount of data is extremely large. This article has further experiments. Figure 4 shows the impact of different  $e$  and  $K$  values on classification accuracy. Different  $e$  and  $K$  values do not affect the accuracy of classification, and the generation of rules is directly related to accuracy. The even distribution can effectively solve the influence of the position, which shows that the uniform distribution can effectively improve the accuracy of the fuzzy system.

Table 2 Examples of rules on PID

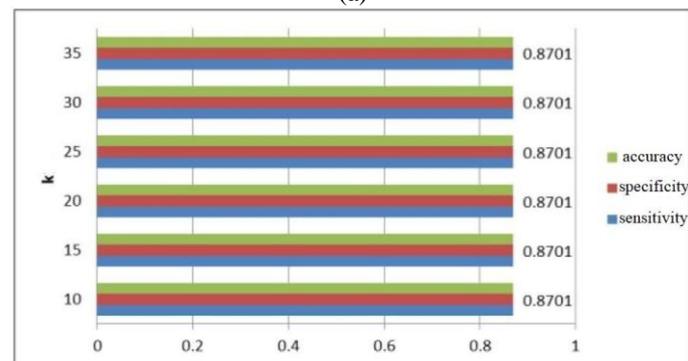
Rvalue	The plasma glucose concentration in the oral glucose tolerance test is 2 hours(A1)	Diastolic blood pressure (mm Hg) (A2)	Triceps skin fold thickness(mm) (A3)	2 hours serum insulin(mu. U/ml) (A4)	Body mass index (A5)	Age (A6)	Class variable	body weight	And/or method
R1	0	0	2	2	2	1	1	0.3333	1
R2	2	1	1	1	1	3	2	0.5000	1
R3	1	1	0	1	0	1	2	0.3889	1
R4	3	1	0	1	0	1	2	0.3333	1
R5	1	1	2	2	3	2	2	0.4444	1
R6	1	1	3	2	0	2	2	0.3889	1

### 5.3 Quantitative evaluation

This section introduces the quantitative evaluation of the proposed fuzzy system. First, the performance of the proposed system is analyzed using various parameter values of  $e$  and  $K$  constants to find accurate values suitable for better performance. Figure 4 (a) and (b) respectively show the effect of different  $e$  and  $K$  values on sensitivity, specificity and accuracy when the proportion of the training set is 80% on PID. It can be clearly seen from the results that the classification effect of the fuzzy system proposed in this paper is not affected by the  $e$  constant and  $K$ .



(a)



(b)

Figure 4 The classification effect of the system on PID under different  $e$  and  $K$  values

The convergence speed of the algorithm is an important reference index to measure the quality of the algorithm in practical applications. For this reason, this article conducts experiments on the convergence speed of the improved algorithm and the original brainstorming algorithm. The experimental results are shown in Figure 5. As can be seen from the figure, the optimization The latter algorithm has a faster convergence speed, especially as the training data increases.

The main purpose of the AGFS algorithm is to improve the accuracy of data classification through the generation and optimization of rules. The focus is not on the poor classification accuracy and low convergence speed caused by the sharp increase in spatial dimensions or variables. This article takes into account the classification accuracy while focusing on By rationally designing the membership function and introducing different constraints, the search space is effectively reduced and the convergence speed of classification is improved. Next, this article conducts experiments from the two perspectives of the algorithm's convergence speed and accuracy, and the results are shown in Figure 5 and Table 3. Through Figure 5 and Table 3, compared with AGFS, our method can effectively improve the convergence speed of classification. The reason is that we not only optimize the generation of rules, but also consider the convergence speed while optimizing the rules, mainly through Design a uniform membership function. Since the optimal rule is selected through three constraints such as rule length, rule number and class distribution, when the amount of data increases sharply, the problem of increasing rules can be alleviated to a certain extent, making the generated rules irregular It will increase exponentially with the rapid increase of data, thereby effectively improving the problem of slow convergence caused by the algorithm when the amount of data increases sharply.

Since the AGFS algorithm focuses on solving the accuracy of fuzzy system classification, although our optimization method can effectively solve the problem of slow convergence, we still need to verify the impact of the algorithm on the accuracy of the classification problem. Due to the improvement of the predefined search, the solution in the space searches for solutions updated in each iteration. The advantage of this improved algorithm is that it can make better use of global information and further improve the evolutionary diversity. The AGFS algorithm uses an adaptive genetic algorithm and can also obtain a global optimal solution. Through the experimental results shown in Table 3, it can be seen that our method can obtain an accuracy rate that is not weaker than AGFS when the amount of data increases sharply.

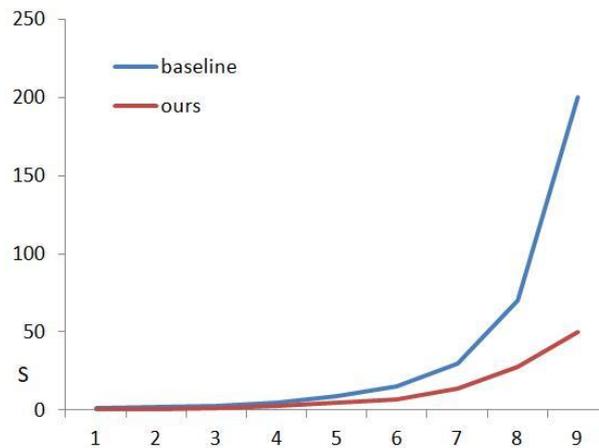


Figure 5 The convergence speed of the algorithm

Table 3 shows the comparison between the improved brainstorming optimization algorithm proposed in the PID data squadron article and AGFS in terms of sensitivity, specificity and accuracy. This article uses training data of different proportions to train the model, and then uses the remaining data for testing. In the selection of training data, in order to make the experiment more universal, this article uses a random selection method to select training data proportionally, The proportion of selected data is set to {70%; 75%; 80%; 85%} to verify that the improved brainstorming optimization algorithm is more effective than traditional methods, and the effectiveness is not accidental. It can be seen from the table that, compared with AGFS, the improved brainstorming optimization algorithm proposed in this paper can obtain a better classification effect. Taking only 70% as an example, when the training data size is 70%, the value obtained by the fuzzy system based on the improved brainstorming optimization algorithm is 86.89%, while the value obtained by AGFS is 86.5%. When the ratio of training and testing data is 75:25, the fuzzy system and AGFS based on the improved brainstorming optimization algorithm reach 86.67% and 86.59%, respectively. When the size of the

training data is 80%, the highest performance achieved by the fuzzy system based on the improved brainstorming optimization algorithm is 87.01%, while the accuracy of AGFS is about 86.22%. This shows that the improved brainstorming algorithm under different training data is more effective than ACFS.

Table 3 Comparison of classification effects of different systems on PID

	Training data ratio	AGFS	Fuzzy system based on improved brainstorming optimization algorithm
Spirit Sensitive degree	0.7	0.8652	0.8689
	0.75	0.8659	0.8667
	0.8	0.8622	0.8701
	0.85	0.8549	0.8625
Special different Sex	0.7	0.8652	0.8689
	0.75	0.8659	0.8667
	0.8	0.8622	0.8701
	0.85	0.8549	0.8625
quasi-ndeed Sex	0.7	0.8652	0.8689
	0.75	0.8659	0.8667
	0.8	0.8622	0.8701
	0.85	0.8549	0.8625

## 6 Conclusion

Aiming at the problem that the fuzzy system has less scalability and accuracy when the dimensionality or variable of the search space increases rapidly in the inductive learning of the rule classifier, this paper proposes to modify the rule definition process in the traditional fuzzy system. A new method of fuzzy system rule generation is proposed. In the rule derivation, by introducing the exponential model into the existing brainstorming algorithm, an improved brainstorming optimization algorithm is proposed. When designing the membership function, a simple method based on uniform distribution is adopted. Finally, the fuzzy score generated by this new fuzzy system is used to classify the data samples. Through the analysis of the experimental results, it can be known that compared with the existing AGFS, this fuzzy system based on the brainstorming optimization algorithm can obtain a better classification effect. Since the optimization of the membership function plays an important role in the fuzzy system, in the following research, this paper hopes to replace the distribution-based method by designing the optimization process of the membership function to improve the effect of the system on data classification.

## References

- [1] Akbal B. Applications of artificial intelligence and hybrid neural network methods with new bonding method to prevent electroshock risk and insulation faults in high-voltage underground cable lines[J]. *Neural Computing and Applications*, 2018, 29(8): 97-105.
- [2] Shunmugapriya P, Kanmani S. A hybrid algorithm using ant and bee colony optimization for feature selection and classification (AC-ABC Hybrid)[J]. *Swarm and evolutionary computation*, 2017: 27-36.
- [3] Lai C M , Huang H P . A gene selection algorithm using simplified swarm optimization with multi-filter ensemble technique[J]. *Applied Soft Computing*, 2021, 100:106994.
- [4] Kim E H, Oh S K, Pedrycz W. Design of double fuzzy clustering-driven context neural networks[J]. *Neural Networks*, 2018, 104: 1-14.
- [5] Gertrudes J C , Zimek A , Sander J , et al. A unified view of density-based methods for semi-supervised clustering and classification[J]. *Data Mining and Knowledge Discovery*, 2019, 33(6):1894-1952.
- [6] Wang Yijie, Wang Tao. Design of fuzzy logic system based on genetic algorithm [J]. *Fuzzy systems and mathematics*, 2018, 32 (5): 78-86.
- [7] ] Amponsah A A , F Han, Ling Q H , et al. An enhanced class topper algorithm based on particle swarm optimizer for global optimization[J]. *Applied Intelligence*, 2020:1-19..
- [8] Man Fenghuan, Chen Xiuhong, he Jiajia, et al. An improved convolutional neural network based on simulated annealing algorithm [J]. *Microelectronics and computer*, 2017, 34 (9): 58-62.
- [9] Shill P C , Paul A K , Murase K . Adaptive Fuzzy Logic Controllers Using Hybrid Genetic Algorithms[J]. *International Journal of Uncertainty Fuzziness & Knowledge Based Systems*, 2019, 27(01):41-71.
- [10] Ilango S S , Vimal S , Kaliappan M , et al. Optimization using Artificial Bee Colony based clustering approach for big data[J]. *Cluster Computing*, 2018.
- [11] Hidayah Nurul Hasanah Zen, Laila Wahyu Trimartanti, Zaenal Abidin, et al. Determining hydrocarbon prospective zone using the combination of qualitative analysis and fuzzy logic method[J]. *Journal of systems science and systems engineering*, 2017, 26(4):463-474.
- [12] Binu, D.; Selvi, M., "BFC: Bat Algorithm Based Fuzzy Classifier for Medical Data Classification", *Journal of Medical Imaging and Health Informatics*, Vol. 5, no. 3, pp. 599-606, June 2015.
- [13] Binu Dennis, S. Muthukrishnan, "AGFS: Adaptive Genetic Fuzzy System for medical data classification", *Applied Soft Computing*, vo. 25, pp. 242–252, 2014.
- [14] Chin Hooi Tan, Keem Siah Yap, Hwa Jen Yap, "Application of genetic algorithm for fuzzy rules optimization on semi expert judgment automation using Pittsburg approach", *Applied Soft Computing*, vol. 12, pp. 2168–2177, 2012.
- [15] Wu Q , Ma Z , Xu G , et al. A Novel Neural Network Classifier Using Beetle Antennae Search Algorithm for Pattern Classification[J]. *IEEE Access*, 2019, PP(99):1-1.
- [16] Lakshmi P , Ramyachitra D . An Improved Genetic with Particle Swarm Optimization Algorithm Based on Ensemble Classification to Predict Protein–Protein Interaction[J]. *Wireless Personal Communications*, 2020, 113(42).
- [17] Dan Meng, Zheng Pei, "Extracting linguistic rules from data sets using fuzzy logic and genetic algorithms", *Neurocomputing*, vol. 78, pp. 48–54, 2012.

- [18] You Y , Sun J , Chen Y W , et al. Ensemble Belief Rule-Based Model for complex system classification and prediction[J]. Expert Systems with Applications, 2020, 164:113952.
- [19] Chen T , Deng N . Research of Association Rules Based on Improved Ant Colony Optimization[C]// 2020 International Conferences on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics). IEEE, 2020.
- [20] Varniab M S, Hung C C, Sharghi V K. Classification of multiclass datasets using genetic programming[C]// Proceedings of the Conference on Research in Adaptive and Convergent Systems. 2019: 76-82.
- [21] Lin Defu, Wang Jun, Jiang Yizhang, et al. A new method of Takagi Sugeno fuzzy system modeling for high dimensional data [J]. Acta electronica Sinica, 2018, v.40 (6): 138-145.
- [22] A E T , A I S , A T B , et al. Classification and Feature Selection Method for Medical Datasets by Brain Storm Optimization Algorithm and Support Vector Machine[J]. Procedia Computer Science, 2019, 162:307-315.