

# Feature selection for fuzzy classifier using the spider monkey algorithm

**Ilya A. Hodashinsky**

E-mail: hodashn@rambler.ru

**Mikhail M. Nemirovich-Danchenko**

E-mail: nemdan2011@gmail.com

**Sergey S. Samsonov**

E-mail: samsonicx@mail.ru

Tomsk State University of Control Systems and Radioelectronics (TUSUR)

Address: 40, Prospect Lenina, Tomsk 634050, Russia

## Abstract

In this paper, we discuss the construction of fuzzy classifiers by dividing the task into the three following stages: the generation of a fuzzy rule base, the selection of relevant features, and the parameter optimization of membership functions for fuzzy rules. The structure of the fuzzy classifier is generated by forming the fuzzy rule base with use of the minimum and maximum feature values in each class. This allows us to generate the rule base with the minimum number of rules, which corresponds to the number of class labels in the dataset to be classified. Feature selection is carried out by a binary spider monkey optimization (BSMO) algorithm, which is a wrapper method. As a data preprocessing procedure, feature selection not only improves the efficiency of training algorithms but also enhances their generalization capability. In the process of feature selection, we investigate the dynamics of changes in classification accuracy, iteration by iteration, for various parameter values of the binary algorithm and analyze the effect of its parameters on its convergence rate. The parameter optimization of fuzzy rule antecedents uses another spider monkey optimization (SMO) algorithm that processes continuous numerical data. The performance of the fuzzy classifiers based on the rules and features selected by these algorithms is tested on some datasets from the KEEL repository. Comparison with two competitor algorithms on the same datasets is carried out. It is shown that fuzzy classifiers with the minimum number of rules and a significantly reduced number of features can be developed with their accuracy being statistically similar to that of the competitor classifiers.

**Key words:** feature selection; wrapper method; binary spider monkey algorithm; fuzzy classifier; binary metaheuristics; fuzzy rule.

**Citation:** Hodashinsky I.A., Nemirovich-Danchenko M.M., Samsonov S.S. (2019) Feature selection for fuzzy classifier using the spider monkey algorithm. *Business Informatics*, vol. 13, no 2, pp. 29–42. DOI: 10.17323/1998-0663.2019.2.29.42

## Introduction

Classification is a pattern recognition and machine-learning problem. Presently there are many different classification algorithms (classifiers) available. When selecting a suitable algorithm, it is necessary to take into account the following criteria: classification accuracy, interpretability of the result, training time, classification time, etc. To classify an object is to identify the name (label) of a class to which this object belongs. The traditional classifier assigns one class label to an object under investigation. In turn, the fuzzy classifier can assign degrees of membership, or soft labels, to all classes. The advantage of fuzzy classifiers is their high interpretability and accuracy [1–3]. The resulting interpretable model is easy to use in the process of decision-making [4].

Selection of relevant features is an important problem of data mining and pattern recognition. A feature is an individual measurable property of an observable object, process, or phenomenon. Using a set of features, a machine-learning algorithm can perform classification. The feature selection problem can be formulated as a search for the optimal subset of features with the minimum redundancy and maximum predictive capability. A compact subset selected from an original set of features makes it possible to reduce computational costs and improve classification accuracy [5]. Feature selection methods are classified based on various criteria. Depending on the type of data in the training set (labeled, unlabeled, or partially labeled data), supervised, non-supervised, and semi-supervised methods are distinguished. Depending on the way of interaction between feature selection algorithms and classifiers, feature selection approaches are divided into three groups: filters, wrappers, and embedded methods. Filters do not depend on the classifier construction algorithm and have the following advantages: low computational complexity, sufficient generalization capability and independence from

the classifier. Their main disadvantage is that features are selected independently. The wrapper method is integrated into the classifier construction process and uses a measure of classification accuracy to assess the selected subset of features. This interaction with the classifier generally yields better results as compared to filters; however, it increases the computational complexity of the method and there is a risk of overfitting [6, 7]. With embedded methods, feature selection is carried out in the process of training and is integrated into the classifier construction algorithm.

Wrapper-based feature selection belongs to the class of NP-hard problems, which is why, in this case, it is reasonable to use heuristic and metaheuristic methods [8–10]. Metaheuristics can be divided into two classes: discrete and continuous ones. Discrete metaheuristics, e.g., the genetic algorithm [11–15], ant colony optimization [16–18], and harmony search [19, 20], have long been successfully employed to solve feature selection problems.

Feature selection is a binary optimization problem, which is why the use of most continuous metaheuristics is preceded by their binarization. Binarization generally uses transfer functions [21], which determine the probability for the elements of the solution vector to change their values from 0 to 1 (and vice versa). This feature selection principle is inherent in particle swarm optimization [22, 23], gravitational search [24, 25], brainstorming [26], teaching–learning-based optimization [27], black hole algorithm [8], grasshopper optimization [28], the salp swarm algorithm [29, 30] and the ant lion optimizer [31].

The recently proposed spider monkey optimization (SMO) algorithm, which mimics the foraging behavior of these social animals, belongs to swarm intelligence methods [32]. This algorithm showed good results in optimizing unimodal and multimodal test functions [32], as well as in solving various optimization problems in the electric power industry, elec-

tronics, telecommunications, biology, medicine and image processing. [33].

The No Free Lunch theorem suggests that no particular algorithm yields the best results for all numerical optimization problems [34, 35]. Hence, new optimization algorithms are being developed. In [6], it was noted that there is no single best method for feature selection, and the researcher should focus on finding a good method for each particular problem.

The purpose of this work is to investigate a binary modification of the SMO algorithm for the wrapper-based selection of the optimal number of relevant features when constructing fuzzy classifiers.

## 1. Methods

### 1.1. Feature selection

Suppose that we have a dataset with  $Z$  instances, each being characterized by  $D$  features  $X = \{x_1, x_2, \dots, x_D\}$ . Feature selection consists in choosing  $d$  features from  $D$  ( $d \leq D$ ) that optimize the objective function. In other words, the feature selection problem can be formulated as follows: on the given set of features  $X$ , find a feature subset that does not cause a significant decrease in classification accuracy as the number of features decreases. A feature selection solution is represented as a binary vector  $\mathbf{S} = (s_1, s_2, \dots, s_D)^T$ , where  $s_i = 0$  means that the  $i$ -th feature is excluded from classification, while  $s_i = 1$  means that the classifier uses the  $i$ -th feature. Classification accuracy is estimated for each feature subset.

### 1.2. Fuzzy classifier

The traditional classifier is defined by the following function:

$$f : \mathfrak{R}^D \rightarrow \{0, 1\}^m,$$

where  $f(\mathbf{x}; \boldsymbol{\theta}) = (h_1, h_2, \dots, h_m)^T$ ,  $h_i = 1$ ,  $h_j = 0$ ,  $j = 1, m$ ,  $i \neq j$ , when the object represented by the vector  $\mathbf{x}$  belongs to the class  $c_i \in C$ ;

$C = \{c_1, c_2, \dots, c_m\}$  is the set of classes; and  $\boldsymbol{\theta}$  is the vector of the classifier's parameters.

The fuzzy classifier determines the class of the object with some degree of confidence:

$$f : \mathfrak{R}^D \rightarrow [0, g]^m.$$

The if-then rule of the fuzzy classifier with feature selection is written as follows:

$$R_j : \text{IF } s_1 \wedge x_1 = A_{1j} \text{ AND } s_2 \wedge x_2 = A_{2j} \text{ AND } \dots \text{ AND } s_D \wedge x_D = A_{Dj} \text{ THEN class} = c_j, j = 1, \dots, R,$$

where  $A_{kj}$  is the fuzzy term that characterizes the  $k$ -th feature in the  $j$ -th fuzzy rule ( $k = 1, \dots, D$ ),  $R$  is the number of fuzzy rules, and  $s_i \wedge x_i$  indicates the presence ( $s_i = 1$ ) or absence ( $s_i = 0$ ) of a feature in the classifier.

Below are the formulas that provide the final solution:

$$\begin{aligned} \text{class} &= c_t, \quad t = \arg \max_{1 \leq j \leq m} \{\beta_j\}, \\ \mu_j(\mathbf{x}_p) &= \mu_{A_{j1}}(x_{p1}) \cdot \dots \cdot \mu_{A_{jn}}(x_{pn}) = \prod_{k=1}^n \mu_{A_{jk}}(x_{pk}), \\ \beta_t(\mathbf{x}_p) &= \sum_{\substack{R_j \\ C_j = \text{class } t}} \mu_j(\mathbf{x}_p) = \sum_{\substack{R_j \\ C_j = \text{class } t}} \prod_{k=1}^n \mu_{A_{jk}}(x_{pk}), \end{aligned}$$

where  $\mu_{A_{jk}}(x_{pk})$  is the membership function for the fuzzy term  $A_{jk}$  at the point  $x_{pk}$ .

On the given dataset  $\{(\mathbf{x}_p; c_p), p = 1, \dots, Z\}$ , the measure of classification accuracy is defined as follows [20, 24]:

$$E(\boldsymbol{\theta}, \mathbf{S}) = \frac{\sum_{p=1}^z \begin{cases} 1, & \text{if } c_p = \arg \max_{1 \leq j \leq m} f_j(\mathbf{x}_p; \boldsymbol{\theta}, \mathbf{S}) \\ 0, & \text{otherwise} \end{cases}}{z},$$

where  $f(\mathbf{x}_p; \boldsymbol{\theta}, \mathbf{S})$  is the output of the fuzzy classifier with the parameter vector  $\boldsymbol{\theta}$  and feature vector  $\mathbf{S}$ .

Wrapper-based feature selection is an NP-hard problem. For its solution, we propose a meta-heuristic called the binary spider monkey optimization (BSMO) algorithm.

### 1.3. Binary spider monkey optimization

The population-based SMO algorithm proposed in [32] operates in the continuous search space and, therefore, is not suitable for solving binary optimization problems. In [36], this algorithm was modified to work in the binary search space. In [33, 36], it was stated that the proposed algorithm is extremely effective for binary optimization as it converges quickly and is less likely to stuck at local optima.

In the process of BSMO, the population is divided into several groups, each having its own leader with the best solution in the group. The population also includes a global leader associated with the best solution in the population. The operation of the BSMO algorithm can be described in terms of its following main stages: initialization, local leader phase, global leader phase and decision-making phase. This algorithm has the following parameters:  $M$  is the number of monkeys in the population; the position of each  $i$ -th monkey represents a solution given by the vector  $\mathbf{S}_i = (s_{i,1}, s_{i,2}, \dots, s_{i,D})$ ;  $L$  is the maximum number of local leaders;  $T$  is the number of iterations; and  $p, pr \in [0, 1]$  are constants.

The  $j$ -th element of the  $i$ -th solution is initialized as follows:

$$s_{i,j} = \begin{cases} 0 & \text{if } rand(0, 1) < p \\ 1 & \text{otherwise} . \end{cases} \quad (1)$$

In BSMO, the basic equations of the continuous algorithm are modified using the logical operators AND ( $\otimes$ ), OR ( $+$ ), and XOR ( $\oplus$ ).

The local leader phase updates the positions of monkeys taking into account the position of the local leader by using the following formula:

$$s_{i,j}^{new} = \begin{cases} s_{i,j} \oplus (b \otimes (LL_{k,j} \oplus s_{i,j})) + (d \otimes (s_{r,j} \oplus s_{i,j})), & \text{if } rand > pr \\ s_{i,j}, & \text{otherwise,} \end{cases} \quad (2)$$

where  $b$  and  $d$  take random values from the set  $\{0; 1\}$ ,  $\mathbf{LL}_k$  is the vector that specifies the coordinates of the local leader in the  $k$ -th group, and  $\mathbf{S}_r$  is the vector that specifies the coordinates of a random monkey from the same group.

The global leader phase sets the coordinates of monkeys in accordance with the position of the global leader:

$$s_{i,j}^{new} = \begin{cases} s_{i,j} \oplus (b \otimes (GL_{k,j} \oplus s_{i,j})) + (d \otimes (s_{r,j} \oplus s_{i,j})), & \text{if } rand > P_i \\ s_{i,j}, & \text{otherwise} , \end{cases} \quad (3)$$

where  $\mathbf{GL}$  is the vector that specifies the coordinates of the global leader and

$$P_i = 0.9 \cdot \frac{E(\boldsymbol{\theta}, \mathbf{S}_i)}{\max E(\boldsymbol{\theta}, \mathbf{S})} + 0.1. \quad (4)$$

The decision-making phase sets the coordinates of monkeys in accordance with the positions of the global and local leaders:

$$s_{i,j}^{new} = \begin{cases} s_{i,j} \oplus (b \otimes (LL_{k,j} \oplus s_{i,j})) + (d \otimes (GL_j \oplus s_{i,j})), & \text{if } rand > pr \\ s_{i,j}, & \text{otherwise} . \end{cases} \quad (5)$$

Below is the pseudocode of the BSMO algorithm.

**Input:**  $M, C, T, p, pr, \boldsymbol{\theta}, D$ .

**Output:**  $\mathbf{S}_{best}$

**loop** on  $i$  from 1 to  $M$

**loop** on  $j$  from 1 to  $D$

initialize the population by formula (1);

**end of loop**

**end of loop**

$C_{current} := 1$ ;

**loop** on  $i$  from 1 to  $M$

$\mathbf{GL} := \text{search\_global\_best}(E(\boldsymbol{\theta}, \mathbf{S}_i))$ ;

$\mathbf{LL} := \text{search\_group\_best}(E(\boldsymbol{\theta}, \mathbf{S}_i))$ ;

**end of loop**

**loop** on  $t$  from 1 to  $T$

```

GLold := GL;
loop on  $i$  from 1 to  $M$ 
    loop on  $j$  from 1 to  $D$ 
        use formula (2);
    end of loop
end of loop
loop on  $i$  from 1 to  $M$ 
    LL := search_group_best( $E(\theta, S_i)$ );
end of loop
loop on  $i$  from 1 to  $M$ 
    loop on  $j$  from 1 to  $D$ 
        use formulas (3) and (4);
    end of loop
end of loop
loop on  $i$  from 1 to  $M$ 
    GL := search_global_best( $E(\theta, S_i)$ );
end of loop
loop on  $i$  from 1 to  $M$ 
    loop on  $j$  from 1 to  $D$ 
        use formula (5);
    end of loop
end of loop
loop on  $i$  from 1 to  $M$ 
    GL := search_global_best( $E(\theta, S_i)$ );
    LL := search_group_best( $E(\theta, S_i)$ );
end of loop
if (GLold = GL) then
     $C_{current} := C_{current} + 1$ ;
    if ( $C_{current} > C$ ) then
        combine all agents in one group;
         $C_{current} := 1$ ;
    otherwise
        divide the largest group into two
        and generate new initial values for
        the new groups by formula (1);
    end of if
end of if
end of loop
return  $S_{best} := best\_solution(S)$ ;

```

#### 1.4. Fuzzy rule base generation algorithm

The algorithm yields a compact initial rule base where each class is represented by one fuzzy rule [20, 24]. The rule is formed based on the minimum and maximum values for each class in the observations table  $\{(\mathbf{x}_p; t_p)\}$ .

The algorithm has the following parameters:  $D$  is the number of features and  $m$  is the number of classes.

Input:  $D, m, \{(\mathbf{x}_p; t_p)\}$ .

Output: fuzzy rule base  $\Omega^*$ .

Initialize an empty rule base  $\Omega := \emptyset$ ;

loop on  $j$  from 1 to  $m$

loop on  $k$  from 1 to  $D$

find the minimum and maximum values of the feature  $k$  in the class  $j$   $\min_{class_{jk}} = \max_p(x_{pk})$  and  $\min_{class_{jk}} := \min_p(x_{pk})$ ;

generate the fuzzy term  $A_{jk}$  that covers the interval

$[\min_{class_{jk}}, \max_{class_{jk}}]$ ;

end of loop

generate the rule  $R_{ij}$  based on the terms  $A_{jk}$ , which assigns a class label  $c_j$  to an observation;

$\Omega^* := \Omega \cup \{R_{ij}\}$ ;

end of loop

return  $\Omega^*$ .

#### 1.5. Datasets

The performance of the BSMO-based feature selection algorithm was tested on 38 datasets from the KEEL repository (<https://sci2s.ugr.es/keel/datasets.php> repository). Table 1 describes the datasets used, where #Fall is the number of features in a dataset, #I is the number of instances, and #C is the number of classes.

Table 1.

Description of the datasets

Dataset	#Fall	#I	#C	Dataset	#Fall	#I	#C
appendicitis	7	106	2	phoneme	5	5404	2
balance	4	625	3	pima	8	768	2
banana	2	5300	2	ring	20	7400	2
bupa	6	345	2	satimage	36	6435	7
cleveland	13	297	5	segment	19	2310	7
coil2000	85	9822	2	shuttle	9	58000	7
contraceptive	9	1473	3	sonar	60	208	2
dermatology	34	358	6	spambase	57	4597	2
ecoli	7	336	8	spectfheart	44	267	2
glass	9	214	7	texture	40	5500	11
haberman	3	306	2	thyroid	21	7200	3
heart	13	270	2	titanic	3	2201	2
hepatitis	19	80	2	twonorm	20	7400	2
ionosphere	33	351	2	vehicle	18	846	4
iris	4	150	3	vowel	13	990	11
monk-2	6	432	2	wdbc	30	569	2
newthyroid	5	215	3	wine	13	178	3
page-blocks	10	5472	5	wisconsin	9	683	2
penbased	16	10992	10	yeast	8	1484	10

## 2. Experimental results

The experiments were carried out based on the 10-fold cross validation scheme while dividing each dataset into the training and test samples in a ratio of 9:1. The classifier was constructed on the training samples, while its accuracy was estimated on the test samples. The average accuracy on the test and training data was defined as follows:

$$Accuracy = \frac{1}{V} \sum_{j=1}^V E(\theta, S),$$

where  $V$  is the number of BSMO runs on the same sample (in our experiments,  $V = 10$ ).

The number of selected features was determined as a mean on all training samples.

The BSMO algorithm has four tunable param-

eters: the number of leaders  $C$ , the thresholds  $p$  and  $pr$ , and the size of the population  $M$ . The default values of the parameters are as follows: the population size is 40, the number of leaders is 8, the number of iterations is 100,  $p = 0.4$ , and  $pr = 0.5$ . For the datasets with 16 and more features, the population size was set to 70.

In the process of feature selection, we investigated the dynamics of change in the classification accuracy, iteration by iteration, for various values of the BSMO parameters. The convergence curves for four datasets – Ionosphere, Hepatitis, Spectfheart, and Wine – are shown in *Figures 1–4*. Here, p00 corresponds to  $p = 0.0$ ; similar designation is used for  $pr$ ; pop10 corresponds to the population size of 10; and L2 corresponds to the number of leaders  $C = 2$ .

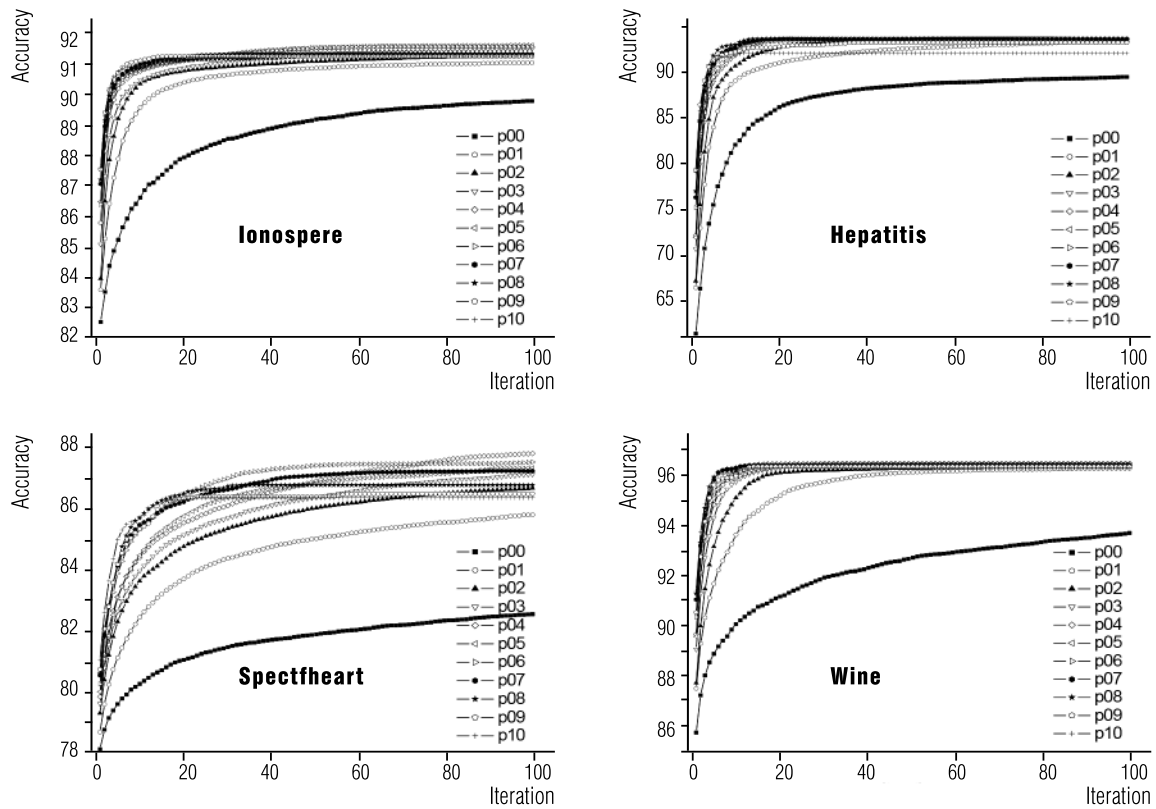


Fig. 1. BSMO convergence curves for eleven values of the parameter  $p$

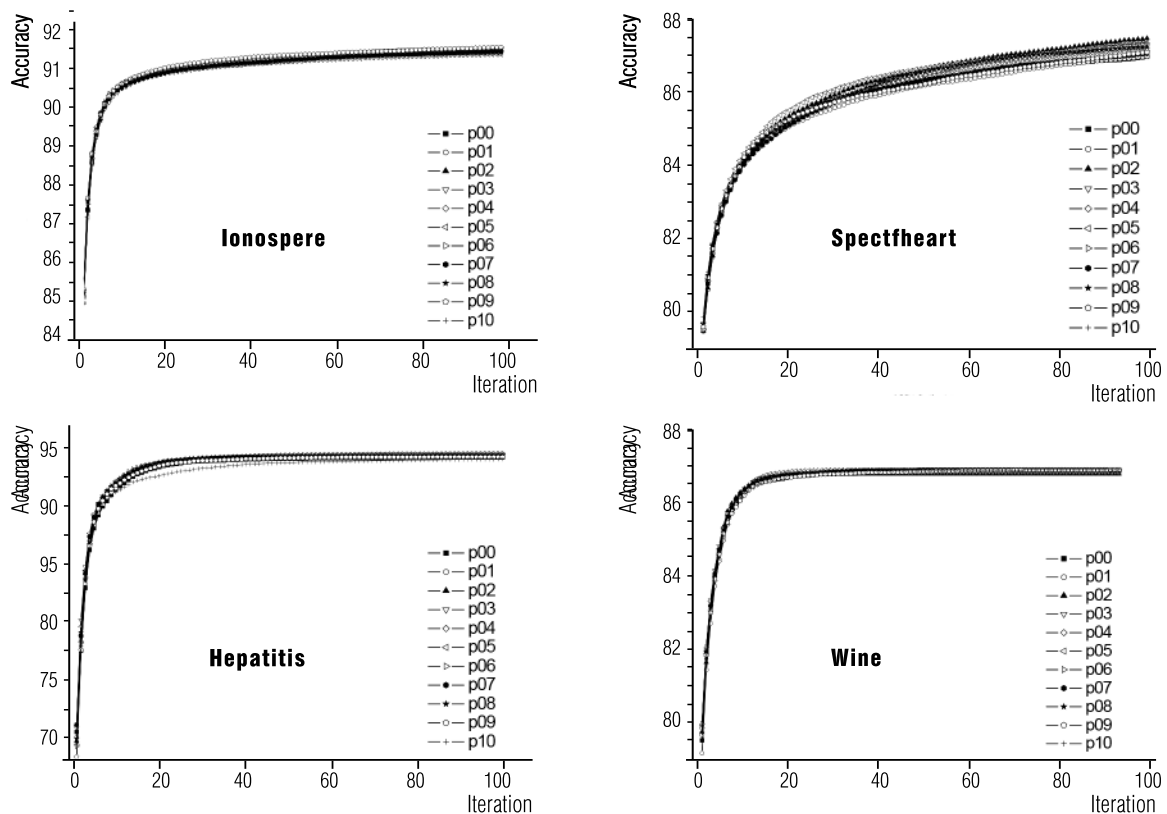


Fig. 2. BSMO convergence curves for eleven values of the parameter  $pr$

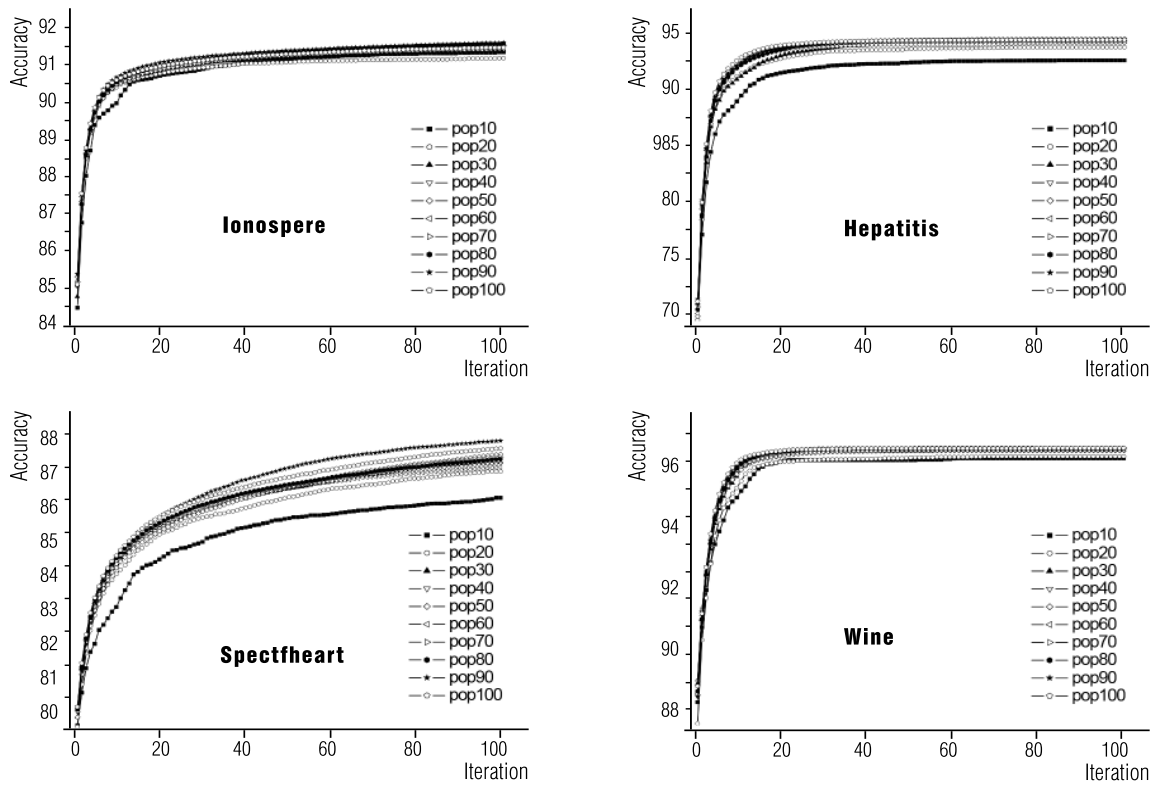


Fig. 3. BSMO convergence curves for ten values of the population size parameter

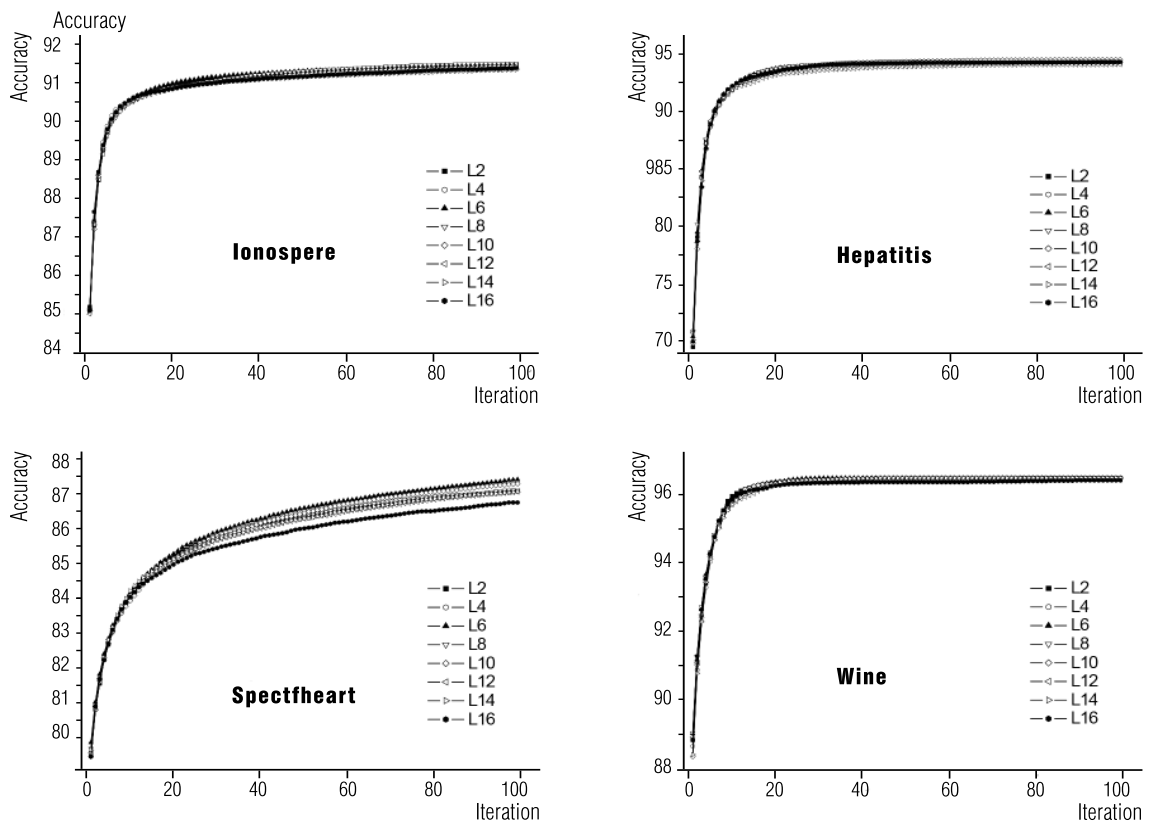


Fig. 4. BSMO convergence curves for eight values of the parameter C



Table 2.

**Comparison of the average accuracies  
on the Wine dataset depending on the number of leaders**

Number of leaders	#L_2		#L_4		#L_6		#L_8		#L_10		#L_12		#L_14		#L_16	
	s	t	s	t	s	t	s	t	s	t	s	t	s	t	s	t
#L_2			+	0.238	+	0.000	+	0.006	-	0.058	-	0.000	-	0.227	-	0.000
#L_4	-	0.238			+	0.010	+	0.201	-	0.001	-	0.000	-	0.001	-	0.000
#L_6	-	0.000	-	0.010			-	0.009	-	0.000	-	0.000	+	0.000	-	0.000
#L_8	-	0.006	-	0.201	+	0.009			-	0.001	-	0.000	-	0.004	-	0.000
#L_10	+	0.058	+	0.001	+	0.000	+	0.001			-	0.000	+	0.974	-	0.000
#L_12	+	0.000	+	0.000	+	0.000	+	0.000	+	0.000			+	0.000	-	0.111
#L_14	+	0.227	+	0.001	+	0.000	+	0.004	-	0.974	-	0.000			-	0.000
#L_16	+	0.000	+	0.000	+	0.000	+	0.000	+	0.000	+	0.111	+	0.000		

Using the paired samples *t*-test, we analyzed the distributions of the average accuracy on the Wine dataset for different numbers of leaders. The null hypothesis was formulated at a significance level of 0.05 as follows: the compared distributions are equal. The results of the comparison (*p*-values) are shown in Table 2; here, *s* is the difference sign (#L<sub>*i*</sub> – #L<sub>*j*</sub>), where *i* is the column pointer and *j* is the row pointer.

Table 3 shows the results of feature selection without optimizing the parameters of the fuzzy classifier. Here, #F is the average number of selected features, #Rt is the ratio between the initial number of features and the number of features selected by the algorithm, #Tra is the average accuracy on the training sample, and #Tst is the average accuracy on the test sample.

### 3. Discussion

Our feature selection experiments with BSMO on 38 datasets allow us to conclude that the number of features can be reduced by

a factor of three (on average) while preserving acceptable classification accuracy. Small discrepancies between the classification accuracies on the training and test data indicate the absence of overfitting.

The threshold *p* can be used to control the convergence rate of the algorithm. Small values of *p* (0.0 or 0.1) reduce the convergence rate. For *p* close to 1, the algorithm converges quickly, but not always to the global optimum. We recommend setting the values of this parameter close to 0.5. The effect of the parameter *pr* is lower; it is recommended to set it in the range of (0.3–0.6). We recommend setting the population size close to the maximum (80–100). The number of leaders has less effect on the convergence of the algorithm, with its value from the interval (6–10) being optimal.

Table 4 compares the feature selection results of the BSMO algorithm with the parameter optimization of the fuzzy classifier by the SMO algorithm [37] and the results of two competitor classifiers, D-MOFARC and FARC-HD [38]. Here, #R is the average number of rules,

Table 3.

**Results of feature selection  
without classifier parameter optimization**

Dataset	#F	#Rt	#Tra	#Tst	Dataset	#F	#Rt	#Tra	#Tst
appendicitis	2.9	2.4	82.81	76.36	phoneme	4	1.3	76.35	76.22
balance	4	1	46.24	46.24	pima	4.1	2	73.63	72.67
banana	1	2	59.42	59.36	ring	1	20	58.51	58.39
bupa	2.7	2.2	60.61	58.17	satimage	10.2	3.5	65.48	64.69
cleveland	8.2	1.6	56.38	54.15	segment	9	2.1	81.80	81.90
coil2000	31.3	2.7	74.93	74.96	shuttle	3.9	2.3	88.82	88.81
contraceptive	2.8	3.2	44.32	44.06	sonar	27.2	2.2	85.05	73.46
dermatology	21.6	1.6	97.95	90.99	spambase	23.7	2.4	75.01	74.27
ecoli	3.9	1.8	54.20	51.22	spectfheart	20.7	2.1	91.21	84.55
glass	6.2	1.5	60.29	56.12	texture	12.1	3.3	74.01	73.69
haberman	1.1	2.7	70.44	67.96	thyroid	1.8	11.7	91.06	90.93
heart	5.4	2.4	79.14	74.21	titanic	1.4	2.1	77.60	77.60
hepatitis	8.1	2.3	94.45	85.51	twonorm	19.7	1	96.87	96.74
ionosphere	16.8	2	93.06	91.58	vehicle	5.9	3.1	49.28	47.87
iris	1.8	2.2	96.81	94.67	vowel	5.3	2.5	47.28	47.68
monk-2	1	6	63.89	63.95	wdbc	10.4	2.9	97.06	95.18
newthyroid	3.5	1.4	97.88	96.77	wine	6.2	2.1	96.5	96.08
page-blocks	2	5	80.42	80.52	wisconsin	5.3	1.7	94.45	93.59
penbased	7	2.3	51.58	51.84	yeast	5.1	1.6	43.67	41.91

#F is the number of features, #L is the average accuracy on the training samples, and #T is the average accuracy on the test samples.

The classifier was constructed on the training samples by using the 10-fold cross validation scheme for each feature set yielded by the BSMO algorithm on each individual dataset. The classifier's parameters were optimized by the SMO algorithm. The average accuracy on the test and training data was determined by computing the mean. Then, for each dataset, a classifier with the highest average accuracy on the test samples was selected.

The statistical significance of the differences

in the classification accuracies, numbers of features, and numbers of fuzzy rules generated by the SMO algorithm and competitor classifiers were estimated using the Wilcoxon signed-rank test.

1. The test indicated a significant difference between the numbers of fuzzy rules in the SMO-based classifiers and the competitor classifiers (p-value < 0.001).

2. The test indicated a significant difference between the numbers of features in the SMO-based classifiers and the competitor classifiers (p-value < 0.001).

3. The test indicated the absence of a sig-

Table 4.

## Performance comparison of the algorithms

№	Dataset	Algorithms											
		BSMO+SMO				D-MOFARC				FARC-HD			
		#R	#F	#L	#T	#R	#F	#L	#T	#R	#F	#L	#T
1	balance	3	4	87.5	86.7	20.1	4	89.4	85.6	18.8	4	92.2	91.2
2	banana	2	2	78.9	78.4	8.7	2	90.3	89	12.9	2	86	85.5
3	bupa	2	3	74.2	70.4	7.7	6	82.8	70.1	10.6	6	78.2	66.4
4	cleveland	5	8	60.7	57.1	45.6	13	90.9	52.9	42.1	13	82.2	58.3
5	ecoli	7	4	72.6	64.0	26.2	7	94	82.7	32.2	7	91.6	81.2
6	glass	7	7	69	63.3	27.4	9	95.2	70.6	18.2	9	79	69
7	haberman	2	1	79.2	74.8	9.2	3	81.7	69.4	5.7	3	79.2	73.5
8	heart	2	6	81.6	80.2	18.7	13	94.4	84.4	27.8	13	93.1	83.7
9	hepatitis	2	10	96.8	91.1	11.4	19	100	90	10.4	19	99.4	88.7
10	iris	3	2	97.8	96.3	5.6	4	98.1	96	4.4	4	98.6	95.3
12	newthyroid	3	4	98.7	96.8	9.5	5	99.8	95.5	9.6	5	99.2	94.4
13	page-blocks	5	2	95.5	95.3	21.5	10	97.8	97	18.4	10	95.5	95
14	penbased	10	6	73.7	72.6	119.2	16	97.4	96.2	152.7	16	97	96
15	phoneme	2	4	79.9	79.2	9.3	5	84.8	83.5	17.2	5	83.9	82.4
16	pima	2	4	75.5	74.3	10.4	8	82.3	75.5	20.2	8	82.3	76.2
17	segment	7	8	88.7	82.3	26.2	19	98	96.6	41.1	19	94.8	93.3
23	spambase	2	25	77.9	76.3	24.3	57	91.7	90.5	30.5	57	92.4	91.6
18	thyroid	3	2	99.6	99.3	5.9	21	99.3	99.1	4.9	21	94.3	94.1
19	titanic	2	2	79.5	79.0	10.4	3	78.9	78.7	4.1	3	79.1	78.8
20	twonorm	2	20	97.5	97.1	10.2	20	94.5	93.1	60.4	20	96.6	95.1
21	wine	3	8	98.9	96.6	8.6	13	100	95.8	8.3	13	100	95.5
22	wisconsin	2	5	96.8	96.4	9	9	98.6	96.8	13.6	9	98.3	96.2

nificant difference between the classification accuracies on the test samples for the classifiers compared:  $p$ -value = 0.153 for the pair (BSMO+SMO – D-MOFARC) and  $p$ -value = 0.148 for the pair (BSMO+SMO – FARC-HD).

Based on the results of the statistical comparison, we can conclude that the reduction in the number of rules and features did not cause a

significant decrease in classification accuracy; the classifiers with a smaller number of features and rules are preferable due to their higher interpretability.

### Conclusion

In this paper, we have described an approach to the construction of fuzzy classifiers. The rule base of the fuzzy classifier has been generated

using the minimum and maximum feature values in each class. Feature selection has been carried out by the wrapper method, namely, the binary spider monkey optimization (BSMO) algorithm. The parameters of the membership functions used in the fuzzy rules have been optimized using the SMO algorithm operating in the continuous search space. The performance of the proposed approach has been tested experimentally on a number of datasets. The first experiment on 38 datasets consisted in reducing the initial set, finding relevant features, and analyzing the effect of the

BSMO parameters on the convergence rate. In the second experiment, we compared BSMO with two well-known metaheuristic methods by using the Wilcoxon signed-rank test. Both experiments have confirmed the effectiveness and competitive advantages of the proposed approach. ■

### Acknowledgements

This work was supported by the Ministry of Education and Science of the Russian Federation, project no. 2.3583.2017/4.6.

### References

1. Alonso J.M., Castiello C., Mencar C. (2015) Interpretability of fuzzy systems: Current research trends and prospects. *Springer Handbook of Computational Intelligence*. Berlin: Springer, pp. 219–237.
2. Mekh M.A., Hodashinsky I.A. (2017) Comparative analysis of differential evolution methods to optimize parameters of fuzzy classifiers. *Journal of Computer and Systems Sciences International*, no 56, pp. 616–626.
3. Zhang Y., Ishibuchi H., Wang S. (2018) Deep Takagi–Sugeno–Kang fuzzy classifier with shared linguistic fuzzy rules. *IEEE Transactions on Fuzzy Systems*, vol. 26, no 3, pp. 1535–1549.
4. Lucca G., Dimuro G.P., Fernández J., Bustince H., Bedregal B.R.C., Sanz J.A. (2019) Improving the performance of fuzzy rule-based classification systems based on a nonaveraging generalization of CC-integrals named  $C_{FI, I2}$ -integrals. *IEEE Transactions on Fuzzy Systems*, vol. 27, no 1, pp. 124–134.
5. Zhang P., Gao W., Liu G. (2018) Feature selection considering weighted relevancy. *Applied Intelligence*, no 48, pp. 4615–4625.
6. Bolon-Canedo V., Sanchez-Marono N., Alonso-Betanzos A. (2015) *Feature Selection for High-Dimensional Data*. London: Springer.
7. Cai J., Luo J., Wang S., Yang S. (2018) Feature selection in machine learning: A new perspective. *Neurocomputing*, no 300, pp. 70–79.
8. Pashaei E., Aydin N. (2017) Binary black hole algorithm for feature selection and classification on biological data. *Applied Soft Computing*, no 56, pp. 94–106.
9. Yusta S.C. (2009) Different metaheuristic strategies to solve the feature selection problem. *Pattern Recognition Letters*, no 30, pp. 525–534.
10. Kohavi R., John G.H. (1997) Wrappers for feature subset selection. *Artificial Intelligence*, no 97, pp. 273–324.
11. Dong H., Li T., Ding R., Sun J. (2018) A novel hybrid genetic algorithm with granular information for feature selection and optimization. *Applied Soft Computing*, no 65, pp. 33–46.
12. Sayed S., Nassef M., Badr A., Farag I. (2019) A nested genetic algorithm for feature selection in high-dimensional cancer microarray datasets. *Expert Systems with Applications*, no 121, pp. 233–243.
13. Ma B., Xia Y. (2017) A tribe competition-based genetic algorithm for feature selection in pattern classification. *Applied Soft Computing*, no 58, pp. 328–338.
14. Wu Y.-L., Tang C.-Y., Hor M.-K., Wu P.-F. (2011) Feature selection using genetic algorithm and cluster validation. *Expert Systems with Applications*, no 38, pp. 2727–2732.

15. Sikora R., Piramuthu S. (2007) Framework for efficient feature selection in genetic algorithm based data mining. *European Journal of Operational Research*, no 180, pp. 723–737.
16. Vieira S.M., Sousa J.M.C., Runkler T.A. (2007) Ant colony optimization applied to feature selection in fuzzy classifiers. *IFSA 2007, LNAI 4529*. Berlin: Springer, pp. 778–788.
17. Ghimatgar H., Kazemi K., Helfroush M.S., Aarabi A. (2018) An improved feature selection algorithm based on graph clustering and ant colony optimization. *Knowledge-Based Systems*, no 159, pp. 270–285.
18. Dadaneh B.Z., Markid H.Y., Zakerolhosseini A. (2016) Unsupervised probabilistic feature selection using ant colony optimization. *Expert Systems with Applications*, no 53, pp. 27–42.
19. Diao R., Shen Q. (2012) Feature selection with harmony search. *IEEE Transactions on Systems, Man, and Cybernetics – PART B: Cybernetics*, no 42, pp. 1509–1523.
20. Hodashinsky I.A., Mekh M.A. (2017) Fuzzy classifier design using harmonic search methods. *Programming and Computer Software*, no 43, pp. 37–46.
21. Mirjalili S., Lewis A. (2013) S-shaped versus V-shaped transfer functions for binary particle swarm optimization. *Swarm and Evolutionary Computation*, no 9, pp. 1–14.
22. Yadav S., Ekbal A., Saha S. (2018) Feature selection for entity extraction from multiple biomedical corpora: A PSO-based approach. *Soft Computing*, no 22, pp. 6881–6904.
23. Ajit Krishna N.L., Deepak V.K., Manikantan K., Ramachandran S. (2014) Face recognition using transform domain feature extraction and PSO-based feature selection. *Applied Soft Computing*, no 22, pp. 141–161.
24. Bardamova M., Konev A., Hodashinsky I., Shelupanov A. (2018) A fuzzy classifier with feature selection based on the gravitational search algorithm. *Symmetry*, no 10 (11), 609.
25. Rashedi E., Nezamabadi-pour H. (2014) Feature subset selection using improved binary gravitational search algorithm. *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology*, no 26, pp. 1211–1221.
26. Papa J.P., Rosa G.H., de Souza A.N., Afonso L.C.S. (2018) Feature selection through binary brain storm optimization. *Computers and Electrical Engineering*, no 72, pp. 468–481.
27. Kiziloz H.E., Deniz A., Dokeroglu T., Cosar A. (2018) Novel multiobjective TLBO algorithms for the feature subset selection problem. *Neurocomputing*, no 306, pp. 94–107.
28. Mafarja M., Aljarah I., Faris H., Hammouri A.I., Ala'M A.Z., Mirjalili S. (2019) Binary grasshopper optimisation algorithm approaches for feature selection problems. *Expert Systems with Applications*, no 117, pp. 267–286.
29. Sayed G.I., Khoriba G., Haggag M.H. (2018) A novel chaotic salp swarm algorithm for global optimization and feature selection. *Applied Intelligence*, no 48, pp. 3462–3481.
30. Faris H., Mafarja M.M., Heidari A.A., Aljarah I., Al-Zoubi A.M., Mirjalili S., Fujita H. (2018) An efficient binary salp swarm algorithm with crossover scheme for feature selection problems. *Knowledge-Based Systems*, no 154, pp. 43–67.
31. Emary E., Zawbaa H.M., Hassanien A.E. (2016) Binary ant lion approaches for feature selection. *Neurocomputing*, no 213, pp. 54–65.
32. Bansal J.C., Sharma H., Jadon S.S., Clerc M. (2014) Spider monkey optimization algorithm for numerical optimization. *Memetic Computing*, no 6, pp. 31–47.
33. Agrawal V., Rastogi R., Tiwari D.C. (2018) Spider monkey optimization: a survey. *International Journal of System Assurance Engineering and Management*, no 9, pp. 929–941.
34. Wolpert D.H. (1996) The existence of a priori distinctions between learning algorithms. *Neural Computation*, no 8, pp. 1341–1390.
35. Wolpert D.H. (1996) The lack of a priori distinctions between learning algorithms. *Neural Computation*, no 8, pp. 1391–1420.

36. Singh U., Salgotra R., Rattan M. (2016) A novel binary spider monkey optimization algorithm for thinning of concentric circular antenna arrays. *IETE Journal of Research*, no 62, pp. 736–744.
37. Hodashinsky I.A., Samsonov S.S. (2017) Design of fuzzy rule based classifier using the monkey algorithm. *Business Informatics*, no 1, pp. 61–67.
38. Fazzolari F., Alcalá R., Herrera F. (2014) A multi-objective evolutionary method for learning granularities based on fuzzy discretization to improve the accuracy-complexity trade-off of fuzzy rule-based classification systems: D-MOFARC algorithm. *Applied Soft Computing*, no 24, pp. 470–481.

### About the authors

#### **Ilya A. Hodashinsky**

Dr. Sci. (Tech.), Professor;

Professor, Department of Complex Information Security of Computer Systems,  
Tomsk State University of Control Systems and Radioelectronics (TUSUR),  
40, Prospect Lenina, Tomsk 634050, Russia;

E-mail: hodashn@rambler.ru

#### **Mikhail M. Nemirovich-Danchenko**

Dr. Sci. (Phys.-Math.);

Professor, Department of Complex Information Security of Computer Systems,  
Tomsk State University of Control Systems and Radioelectronics (TUSUR),  
40, Prospect Lenina, Tomsk 634050, Russia;

E-mail: nemdan2011@gmail.com

#### **Sergey S. Samsonov**

Doctoral Student, Department of Complex Information Security of Computer Systems,  
Tomsk State University of Control Systems and Radioelectronics (TUSUR),  
40, Prospect Lenina, Tomsk 634050, Russia;

E-mail: samsonicx@mail.ru