УДК 004.89

Р.О. Остапенко, И.А. Ходашинский

Формирование базы правил нечёткого классификатора с помощью метаэвристического алгоритма «саранчи»

Приведено описание гибридного алгоритма формирования нечетких правил нечеткого классификатора с использованием метаэвристического алгоритма «саранчи» и алгоритма кластеризации данных K-средние. Качество кластеризации оценивалось тремя функциями пригодности: суммарная дисперсия, индекс Дэвиса–Боулдина и индекс Калински–Харабаса. Были исследованы треугольные и гауссовы функции принадлежности. Эффективность сгенерированных баз нечетких правил проверена на реальных наборах данных. Лучшей комбинацией является использование суммарной дисперсии в качестве функции пригодности и гауссовой функции в качестве функции принадлежности.

Ключевые слова: кластеризация, нечеткий классификатор, K-средние, алгоритм «саранчи».
DOI: 10.21293/1818-0442-2022-25-2-31-36

Классификация и кластеризация

Классификация – важная составляющая научного направления, получившего название «машинное обучение». Однако само понятие «классификация» неоднозначно, оно содержит несколько толкований:

1) процесс «построение классификатора – разделение множества объектов (наблюдений) на группы (классы), на основе анализа их признакового описания»;

2) процесс «применение классификатора»;

3) результат выделения классов.

Признаки характеризуют какой-либо наблюдаемый феномен, признаки можно измерить, используя различные шкалы.

В статье классификация – это обучение с учителем, обучение на помеченных данных ($\mathbf{x}_i$, c_j).

Ниже приведена постановка задачи классификации по первому толкованию.

Пусть <**X**, **A**, **C**> – набор данных; $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{|X|}\}$ – непустое конечное множество объектов (наблюдений); $\mathbf{A} = \{a_1, \ldots, a_n\}$ – непустое конечное множество признаков (атрибутов); $C = \{c_1, c_2, \ldots, c_m\}$ – непустое конечное множество меток классов; $c_j \in C$ – значение метки класса для j-го наблюдения.

На множестве объектов **X** и множестве меток классов C построить алгоритм (функцию или классификатор) c: $\mathbf{X} \rightarrow C$, способный указать метку класса для произвольного объекта из исходного множества; $c = c(\mathbf{a}; \boldsymbol{\theta})$ – метка, соответствующая вектору признаков **a**; **θ** – вектор параметров классификатора.

Среди множества различных типов классификаторов нечеткий классификатор (НК) выделяется возможностью интерпретации как собственно полученного результата, так и процесса его получения [1, 2]. Процесс построения НК включает три этапа: отбор признаков [3, 4], формирование базы нечетких правил [5, 6], оптимизация параметров НК [7, 8].

Наиболее часто используемым подходом к формированию базы нечетких правил является кластеризация исходных данных. Центроидные методы кластеризации, такие как K-средние [9], связаны с поиском групп данных на основе их сходства путем определения центров кластеров и их радиусов. Количество найденных кластеров в наборе данных определяет количество возможных нечетких «ЕСЛИ-ТО» правил. Из каждого кластера формируется нечеткое правило вида

R_j: ЕСЛИ **x** равно $\mathbf{A}_j$, ТО класс c_j,

где R_j – метка j-го правила, j = 1, 2, …, K; **x** – n-мерный вектор входных признаков, т.е. $\mathbf{x} = (x_1, \ldots, x_n)$; $\mathbf{A}_j$ – вектор имен функций принадлежности, в нашем случае треугольного либо гауссового типа.

Задачей кластеризации является группировка множества объектов таким образом, чтобы объекты внутри одного кластера были подобны по заданной метрике. Функция расстояния между объектами $\rho(\mathbf{x}_i, \mathbf{x}_j)$ задана на множестве **X**. Необходимо сопоставить метку $c_i \in C$ объекту $\mathbf{x}_i \in \mathbf{X}$ таким образом, чтобы объекты внутри каждого кластера были близки относительно метрики ρ, но при этом объекты разных кластеров значительно различались. В отличие от классификации данные в наборе не размечены, т.е. метки c_i изначально не заданы.

Невозможно получить однозначное решение задачи кластеризации: заранее неизвестно число кластеров, не существует однозначного критерия качества (функции пригодности) кластеризации, присутствует существенная зависимость от метрики ρ [10].

Так как кластеризация данных может быть сведена к задаче оптимизации, для ее решения часто применяют метаэвристические алгоритмы с последующим построением базы нечетких правил [11–15]. Согласно теореме об «отсутствии бесплатных завтраков» [16], не существует универсального алгоритма, пригодного для решения любых задач оптимизации. Указанный факт заставляет исследователей искать новые методы решения задач оптимизации.

В [17] предложен популяционный алгоритм оптимизации, имитирующий поведение роя саранчи, и описано его применение для решения задач поиска оптимума сложных математических функций, а также для проектирования оптимальных форм консольной балки и ферм различной конфигурации. Алгоритм «саранчи» в [18] применен для решения определения места повреждения конструкции и степени его тяжести. В [19] описано эффективное примене-

ние алгоритма «саранчи» для решения задачи визуального отслеживания нескольких объектов в видеопотоке со сложным фоном. В [20] описано применение алгоритма «саранчи» для построения контроллера интегрированной фотоэлектрической системы выработки электроэнергии, а в [21] – для вспомогательного контроллера системы возбуждения синхронного генератора при подавлении низкочастотных колебаний в энергосистеме. Кроме того, этот алгоритм успешно применялся в решении задач обработки изображений [22, 23], анализа вибрационных сигналов [24] и сигналов ЭКГ [25], а также отбора информативных признаков [26, 27]. Результаты применения алгоритма показали его высокую эффективность в решении реальных задач с аналитически незаданными пространствами поиска.

Целью статьи является исследование возможности применения метаэвристического алгоритма «саранчи» и различных функций пригодности для формирования базы правил нечеткого классификатора.

Алгоритм «саранчи»

Миллионы личинок саранчи передвигаются как катящиеся цилиндры. На своем пути они едят почти всю растительность. Когда они становятся взрослыми, то образуют рой в воздухе. Так саранча мигрирует на большие расстояния. Основная характеристика роя в личиночной фазе – медлительность, короткие движения. С другой стороны, резкие, дальние движения – важнейшая особенность роя во взрослом возрасте. Алгоритм учитывает фазу личинок и фазу миграции роя. Рой пытается найти зону комфорта, к которой стремятся все особи (этап интенсификации). Кроме силы стремления попасть в зону комфорта, есть сила отталкивания, что позволяет каждой отдельной особи искать лучшее решение (этап диверсификации). Следующая позиция особи определяется на основе её текущего положения, лучшего решения на данный момент и положения всех других особей [17].

Ниже представлен собственно алгоритм «саранчи»:

Алгоритм «саранчи»

Вход: N – размер популяции, $iter$max – максимальное количество итераций;
Выход: **T** – лучшее решение.
1: Случайным образом сгенерировать N решений.
2: Найти лучшее решение **T** среди сгенерированных N решений.
3: **while** $k < iter$max **do**
4: Расчёт коэффициента c по формуле 1;
5: **while** $i < N$ **do**
6: Поиск i-го решения по формуле 2;
7: **end while**;
8: Найти лучшее решение **T** в новой популяции.
9: $k = k + 1$;
10: **end while**

Коэффициент c, отвечающий за соблюдение баланса диверсификация-интенсификация, определяется следующим образом:

$$c = c\max - k \cdot \frac{c\max - c\min}{iter\max}, \quad (1)$$

где $c$max, $c$min – максимальное и минимальное значение коэффициента c, itermax – максимальное количество итераций, k – текущая итерация.

$$\mathbf{x}_i(k) = c \cdot \left(\sum_{\substack{j=1 \\ j \neq i}}^{N} c \cdot \frac{ub - lb}{2} \cdot s(\rho_{ij}) \cdot \frac{\mathbf{x}_j(k-1) - \mathbf{x}_i(k-1)}{\rho_{ij}} \right) + \mathbf{T}, \quad (2)$$

где ub, lb – верхняя и нижняя границы поиска; $\rho_{ij} = |\mathbf{x}_i - \mathbf{x}_j|$– расстояние между i-м и j-м решением; $s(\rho) = f \cdot \exp(-\rho/v)$ – сила притяжения; f, v – константы; k – текущая итерация.

Гибрид алгоритмов «саранчи» и К-средние

Первоначальные координаты кластеров в гибридном алгоритме формируются с помощью алгоритма К-средние. Далее в цикле выполняется одна итерация алгоритма «саранчи», использующего заранее заданную функцию пригодности, и одна итерация алгоритма К-средние; определяется лучшее решение. Полученные в итоге координаты центроидов кластеров используются для формирования базы правил НК.

Популяция состоит из единственного решения, которое представлено в виде матрицы $\mathbf{T} = (\mathbf{q}_1, \ldots, \mathbf{q}_K)$, где $\mathbf{q}_i = (q_{i1}, \ldots, q_{in})$, q_{il} – l-я координата i-го центроида в n-мерном пространстве признаков.

Гибридный алгоритм приведен ниже:

Алгоритм кластеризации

Вход: K – количество кластеров, $iter$max – максимальное количество итераций, $f(\mathbf{X})$ –функция пригодности, $\mathbf{X}$ – текущее решение;
Выход: **T** – координаты центроидов кластеров.
1: Сгенерировать **T** с помощью алгоритма K-средние.
2: $fit_T = f(\mathbf{T})$.
3: **while** $k < iter$max **do**
4: Формирование $\mathbf{X}_k$ с помощью последовательного выполнения одной итерации алгоритма «саранчи» и одной итерации алгоритма K-средние;
5: $fit_k = f(\mathbf{X}_k)$.
6: **if** $fit_T > fit_k$ **then** $fit_T = fit_k$, $\mathbf{T} = \mathbf{X}_k$.
7: $k = k + 1$;
8: **end while**

В качестве первой исследуемой функции пригодности выбрана суммарная дисперсия F_1:

$$F_1 = \sum_{k=1}^{K} \sum_{x_i \in C_k} \|\mathbf{x}_i - \mathbf{c}_k\|^2 \to \min, \; i = \overline{1,N}, \; k = \overline{1,K}, \quad (3)$$

где $\mathbf{x}_i$ – i-й экземпляр таблицы наблюдений; $\mathbf{c}_k$ – координаты центроида k-го кластера; N – количество экземпляров в наборе данных; K – количество кластеров.

Недостаток данной функции в том, что в ней не учитывается межкластерное расстояние.

В качестве второй функции пригодности выбран индекс Дэвиса–Боулдина (DB) [28]:

$$DB = \frac{1}{K}\sum_{i=1}^{K}\max_{j\neq i}\left(\frac{\sigma_i+\sigma_j}{\rho(\mathbf{c}_i,\mathbf{c}_j)}\right), \qquad (4)$$

где $\mathbf{c}_k$ – координаты центроида k-го кластера; σ_i – среднее расстояние всех элементов в i-м кластере до i-го центроида; K – количество кластеров.

В качестве третьей исследуемой функции пригодности использован индекс Калински–Харабаса (*CH*) [29]:

$$CH = \frac{\sum_{k=1}^{K} n_k \cdot \|\mathbf{c}_k - \mathbf{c}\|}{K-1} \Bigg/ \frac{\sum_{k=1}^{K}\sum_{i=1}^{n_k}\|\mathbf{c}_i - \mathbf{c}_k\|}{N-K}, \qquad (5)$$

где n_k – объем k-го кластера; $\mathbf{c}$ – координаты центра всего набора данных.

Индексы *CH* и *DB* учитывают как внутрикластерное расстояние между точками, так и межкластерное расстояние.

Эксперимент

Проведён эксперимент по исследованию влияния функций пригодности на формирование базы правил нечёткого классификатора. Эксперимент проводился на наборах данных из репозитория KEEL [30]. Были использованы треугольные и гауссовы функции принадлежности.

Таблица 1

Значения точности классификации. Обучение

Набор данных	CHS		DBS		Экст		F1	
	Трг	Гсс	Трг	Гсс	Трг	Гсс	Трг	Гсс
iris	94,67	**96,15**	93,63	96	94,44	95,19	94,44	95,56
newthyroid	93,44	94,57	91,99	93,39	95,81	**96,49**	91,53	95,61
magic	77,69	77,28	77,61	77,4	56,95	59,68	**79,23**	77,44
page-blocks	91,87	92,1	91,63	91,78	50,4	5,28	**92,53**	92,11
wine-red	54,74	53,92	54,05	54,81	19,86	17,83	**55,14**	54,53
wine-white	49,53	49,19	49,33	49,58	26,45	26,04	**49,89**	49,02
marketing	22,48	25,24	22,55	26,22	9,57	9,62	25,33	**26,82**
wine	92,26	94,01	93,7	94,82	88,26	90,82	91,7	**95,69**
cleveland	54,39	56,79	55,18	**57,91**	44,67	47,03	53,8	57,24
heart	73,54	**81,03**	73,58	76,58	67,33	68,64	75,64	80,16
penbased	77,69	79,34	74,8	81,07	56,83	61,46	**85,92**	79,58
vehicle	54,81	55,71	54,32	56,79	29,87	24,6	**59,81**	57,18
hepatitis	84,57	85,13	86,11	**86,79**	27,01	69,63	84,89	85,97
bands	64,12	65,35	63,23	**68,62**	52,84	56,47	64,02	67,89
ring	**76,19**	70,06	70,36	71,32	49,55	49,52	68,57	73,51
twonorm	95,84	96,64	95,78	96,52	96,04	**96,98**	95,94	96,37
thyroid	92,58	92,58	92,58	92,58	7,05	22,77	**92,74**	92,58
wdbc	93,07	93,38	93,46	**94,57**	92,58	91,17	93,63	93,89
ionosphere	80,78	83,73	82,97	85,53	79,61	**90,66**	78,79	85,95
dermatology	78,83	**95,06**	79,52	92,83	75,45	93,95	81,4	94,88
satimage	82,13	82,07	82,02	81,55	60,26	59,02	**82,85**	82,27
texture	71,08	71,25	70,78	73,16	69,78	68,14	**91,64**	72,01
spectfheart	78,86	79,24	79,65	79,61	**80,44**	68,75	79,82	79,94
sonar	64,31	**70,67**	63,35	68,91	57,37	58,92	68,69	70,46
optdigits	21,59	56,71	21,84	54,9	10	27,33	47,52	**56,75**
movement	41,23	50,68	40,4	50,74	47,62	49,97	41,64	**52,87**
Среднее	71,63	74,92	71,32	75,15	55,62	57,92	74,12	**75,63**

В табл. 1 приведены значения средней точности классификации, полученные на обучающей выборке после генерации базы правил. Здесь приняты следующие обозначения: CH – результаты, полученные с помощью гибридного алгоритма «саранчи» с индексом *CH*, DB – с индексом *DB*, F_1 – с суммарной дисперсией в качестве функции пригодности, Экст – результаты, полученные с помощью алгоритма экстремальных значений признака в классе [31], Трг – треугольная функция принадлежности, Гсс – гауссовы функции принадлежности. Жирным шрифтом выделены наибольшие значения. Наибольшая средняя точность классификации получена при использовании гауссовых функций в качестве функции принадлежности и суммарной дисперсии в качестве функции пригодности.

В табл. 2 приведены значения средней точности классификации, полученные на тестовой выборке после генерации базы правил.

Таблица 2

Значения точности классификации. Тест

Набор данных	CH		DB		Экст		F1	
	Трг	Гсс	Трг	Гсс	Трг	Гсс	Трг	Гсс
iris	94,67	97,33	93,63	97,33	94,67	94,67	**98**	97,33
newthyroid	93,05	94,46	92,1	94,96	95,41	**96,3**	93,98	94,44
magic	77,61	77,11	77,44	77,11	56,88	59,7	**79,24**	77,56
page-blocks	91,79	91,94	91,63	91,94	51,17	4,99	**92,11**	91,94
wine-red	54,65	54,79	54,91	54,79	19,7	17,51	**56,16**	54,66
wine-white	49,12	49,06	49,53	49,06	26,42	25,89	**50,41**	49,53
marketing	22,12	24,7	23,19	24,7	9,52	9,65	24,61	**26,75**
wine	94,41	95,46	92,61	95,46	87,55	89,31	93,79	**96,01**
cleveland	**56,61**	55,51	55,23	55,51	42,83	43,92	56,24	55,53
heart	76,3	75,93	73,7	75,93	67,04	67,41	**76,3**	75,93
penbased	77,57	79	74,68	79	56,69	61,43	**85,72**	79,68
vehicle	54,01	53,32	53,68	53,32	29,9	24,11	**57,93**	54,96
hepatitis	88,5	**90,69**	85,23	**90,69**	28,9	65,22	88,91	89,5
bands	63,15	65,36	64,06	65,36	52,13	55,59	65,02	**66,39**
ring	**76,39**	69,8	70,97	69,8	49,53	49,51	68,77	72,53
twonorm	95,86	96,69	95,91	96,69	96,09	**96,97**	95,86	96,41
thyroid	92,58	92,58	92,58	92,58	7,11	22,79	**92,78**	92,58
wdbc	94,55	94,37	94,72	94,37	91,91	90,67	94,37	**96,14**
ionosphere	82,05	84,62	82,9	84,62	79,76	**90,03**	79,76	86,61
dermatology	78,52	92,73	81,79	92,73	75,72	89,38	82,98	**93,52**
satimage	81,9	81,8	82,07	81,8	60,36	58,94	**82,77**	81,96
texture	70,98	71,78	70,31	71,78	69,96	68,16	**91,4**	71,49
spectfheart	79,79	80,56	79,8	80,56	**80,88**	66,71	76,44	80,16
sonar	63	**66,4**	65,38	66,4	57,24	55,81	65,43	65,93
optdigits	21,94	**56,74**	21,92	**56,74**	10,02	26,87	47,69	56,07
movement	38,61	44,17	36,67	42,22	**48,06**	42,78	36,67	43,89
Среднее	71,91	74,5	71,41	71,44	55,59	56,7	74,36	**74,9**

Наибольшая средняя точность классификации получена при использовании гауссовых функций в качестве функции принадлежности и суммарной дисперсии в качестве функции пригодности.

В табл. 3 приведены значения количества правил, сформированных гибридными алгоритмами с разными функциями пригодности, жирным шрифтом выделено наименьшее количество правил.

Сравнения результатов проводились с использованием статистического критерия Фридмана. Нулевая гипотеза (H0) сформирована следующим об-

разом: результаты разных алгоритмов генерации базы правил имеют только случайные различия. Отрицание нулевой гипотезы (H1) – результаты имеют не случайные различия. Уровень значимости выбран равным 0,05.

Таблица 3

Количество правил

Набор данных	CH		DB		Экст		F1	
	Трг	Гсс	Трг	Гсс	Трг	Гсс	Трг	Гсс
iris	7	9	8	9	**3**	**3**	11	11
newthyroid	10	15	14	11	**3**	**3**	10	19
magic	20	19	19	19	**2**	**2**	20	20
page-blocks	16	20	15	20	**5**	**5**	20	20
wine-red	19	9	15	11	**11**	**11**	15	18
wine-white	20	17	20	17	**11**	**11**	18	19
marketing	16	20	14	20	**9**	**9**	19	9
wine	8	13	9	13	**3**	**3**	18	20
cleveland	4	7	5	7	**5**	**5**	6	9
heart	20	19	20	19	**2**	**2**	19	16
penbased	20	20	20	20	**10**	**10**	20	18
vehicle	18	18	14	18	**4**	**4**	20	19
hepatitis	4	6	14	6	**2**	**2**	15	7
bands	11	6	12	6	**2**	**2**	10	20
ring	4	7	7	7	**2**	**2**	12	7
twonorm	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**
thyroid	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**
wdbc	15	19	17	19	**2**	**2**	18	15
ionosphere	5	13	5	13	**2**	**2**	5	6
dermatology	13	17	18	17	**6**	**6**	18	18
satimage	19	20	18	20	**7**	**7**	20	19
texture	20	19	20	19	**11**	**11**	19	20
spectfheart	16	20	13	20	**2**	**2**	14	14
sonar	16	16	8	16	**2**	**2**	11	16
optdigits	20	20	18	20	**10**	**10**	20	19
movement	19	18	17	18	**15**	**15**	20	20
Среднее	13,27	14,31	13,27	14,23	**5,23**	**5,23**	14,73	14,77

В табл. 4 указаны значения средних рангов по критерию Фридмана, полученных при использовании треугольных и гауссовых функций принадлежности. Для всех экспериментов p-value < 0,001, нулевая гипотеза отклоняется.

Таблица 4

Средние ранги алгоритмов генерации правил

	CH		DB		Экст		F1	
	Трг	Гсс	Трг	Гсс	Трг	Гсс	Трг	Гсс
Обучение, точность	4,04	5,46	3,73	6,08	2,13	2,81	5,25	**6,50**
Тест, точность	4,27	5,60	3,88	5,52	2,33	2,65	5,54	**6,21**
Количество правил	5,00	5,42	4,75	5,42	**1,85**	**1,85**	5,87	5,84

Заключение

В работе был исследован гибридный алгоритм кластеризации для формирования базы правил нечеткого классификатора с применением трех функций пригодности и двух типов функций принадлежности.

Использование гауссовых функций принадлежности позволяет достичь большей точности классификации на всех исследованных функциях пригодности.

Использование суммарной дисперсии в качестве функции пригодности позволяет достичь большей точности классификации как на треугольных, так и на гауссовых функциях принадлежности.

Общий вывод: лучшей комбинацией является использование суммарной дисперсии в качестве функции пригодности и гауссовой функции в качестве функции принадлежности.

Работа выполнена при поддержке Российского научного фонда (проект № 22-21-00021).

Литература

1. Explainable Fuzzy Systems: Paving the Way from Interpretable Fuzzy Systems to Explainable AI Systems / J.M.A. Moral, C. Castiello, L. Magdalena, C. Mencar. – Cham: Springer, 2021. – 253 p.

2. Ходашинский И.А. Параметрическая идентификация нечетких моделей на основе гибридного алгоритма муравьиной колонии / И.А. Ходашинский, П.А. Дудин // Автометрия. – 2008. – Т. 44, № 5. – С. 24–35.

3. Ходашинский И.А. Отбор классифицирующих признаков с помощью популяционного случайного поиска с памятью / И.А. Ходашинский, К.С. Сарин // Автоматика и телемеханика. – 2019. – № 2. – С. 161–172.

4. Ходашинский И.А. Отбор классифицирующих признаков: сравнительный анализ бинарных метаэвристик и популяционного алгоритма с адаптивной памятью / И.А. Ходашинский, К.С. Сарин // Программирование. – 2019. – № 5. – С. 3–9.

5. Корышев Н.П. Алгоритм формирования базы правил нечёткого классификатора на основе алгоритма кластеризации k-средних и метаэвристического алгоритма «китов» / Н.П. Корышев, И.А. Ходашинский // Доклады ТУСУР. – 2021. – Т. 24, № 1. – С. 42–47.

6. Бардамова М.Б. Формирование структуры нечёткого классификатора комбинацией алгоритма экстремумов классов и алгоритма «прыгающих лягушек» для несбалансированных данных с двумя классами / М.Б. Бардамова, И.А. Ходашинский // Автометрия. – 2021. – Т. 57, № 4. – С. 54–64.

7. Ходашинский И.А. Идентификация нечетких систем на базе алгоритма имитации отжига и методов, основанных на производных // Информационные технологии. – 2012. – № 3. – С. 14–20.

8. Ходашинский И.А. Оптимизация параметров нечетких систем на основе модифицированного алгоритма пчелиной колонии / И.А. Ходашинский, И.В. Горбунов // Мехатроника, автоматизация, управление. – 2012. – № 10. – С. 15–20.

9. Xu R. Clustering / R. Xu, D.C. Wunsch // New Jersey. – Hoboken: John Wiley & Sons, Inc., 2009. – 357 p.

10. Abraham A. Swarm intelligence algorithms for data clustering / A. Abraham, S. Das, S. Roy // Soft Computing for Knowledge Discovery and Data Mining. – 2007. – P. 279–313.

11. Nanda S.J. A survey on nature inspired metaheuristic algorithms for partitional clustering / S.J. Nanda, G. Panda // Swarm and Evolutionary Computation. – 2014. – Vol. 16. – P. 1–18.

12. Accelerated Two-Stage Particle Swarm Optimization for Clustering Not-Well-Separated Data / X. Xu, J. Li, M.C. Zhou, J. Xu, J. Cao // IEEE Transactions on Systems,

Man, and Cybernetics: Systems. – 2020. – Vol. 50, No. 11. – P. 4212–4223.

13. Sharma M. An efficient hybrid PSO polygamous crossover based clustering algorithm / M. Sharma, J.K. Chhabra // Evolutionary Intelligence. – 2021. – Vol. 14. – P. 1213–1231.

14. Elephant search algorithm applied to data clustering / S. Deb, Z. Tian, S. Fong, R. Wong, R. Millham, K.K.L. Wong // Soft Computing. – 2018. – Vol. 22. – P. 6035–6046.

15. Воронин Д.А. Алгоритм «ворон» для оптимизации параметров нечеткого классификатора / Д.А. Воронин, И.А. Ходашинский // Электронные средства и системы управления: матер. докл. междунар. науч.-практ. конф. – 2021. – № 1-2. – С. 320–323.

16. Wolpert D. No Free Lunch Theorems for Optimization / D. Wolpert, W. Macready // EEE Transactions on Evolutionary Computation. – 1997. – Vol. 1, No. 1. – P. 67–82.

17. Saremi S. Grasshopper Optimisation Algorithm: Theory and application / S. Saremi, S. Mirjalilia, A. Lewis, S. Saremietal // Advances in Engineering Software. – 2017. – Vol. 105. – P. 30–47.

18. Nabavi S. Damage detection in frame elements using Grasshopper Optimization Algorithm (GOA) and time-domain responses of the structure / S. Nabavi, S. Gholampour, M.S. Haji // Evolving Systems. – 2022. – Vol. 13. – P. 307–318.

19. Reddy K.N. A novel method to solve visual tracking problem: hybrid algorithm of grasshopper optimization algorithm and differential evolution / K.N. Reddy, P. Bojja // Evolutionary Intelligence. – 2022. – Vol. 15. – P. 785–822.

20. Ram B.V.K. Grasshopper optimization algorithm utilized Xilinx controller for maximum power generation in photovoltaic system / B.V.K. Ram, N. Chidambararaj // Evolving Systems. – 2021. – Vol. 12. – P. 885–898.

21. Falehi A.D. Optimal robust disturbance observer based sliding mode controller using multi-objective grasshopper optimization algorithm to enhance power system stability // Journal of Ambient Intelligence and Humanized Computing. – 2020. – Vol. 11. – P. 5045–5063.

22. Dinh P.-H. A novel approach based on Grasshopper optimization algorithm for medical image fusion // Expert Systems with Applications. – 2021. – Vol. 171. – P. 114576.

23. Bhandari A.K. A novel local contrast fusion-based fuzzy model for color image multilevel thresholding using grasshopper optimization / A.K. Bhandari, K. Rahul // Applied Soft Computing. – 2019. – Vol. 81. – P. 105515.

24. Zhang X. A parameter-adaptive VMD method based on grasshopper optimization algorithm to analyze vibration signals from rotating machinery / X. Zhang, Q. Miao, H. Zhang, L. Wang // Mechanical Systems and Signal Processing. – 2018. – Vol. 108. – P. 58–72.

25. Malghan P.G. Grasshopper optimization algorithm based improved variational mode decomposition technique for muscle artifact removal in ECG using dynamic time warping / P.G. Malghan, M.K. Hota // Biomedical Signal Processing and Control. – 2022. – Vol. 73. – P. 103437.

26. Zakeri A. Efficient feature selection method using real-valued grasshopper optimization algorithm / A. Zakeri, A. Hokmabadi // Expert Systems with Applications. – 2019. – Vol. 119. – P. 61–72.

27. Kamel S.R. Feature selection using grasshopper optimization algorithm in diagnosis of diabetes disease / S.R. Kamel, R.Yaghoubzadeh // Informatics in Medicine Unlocked. – 2021. – Vol. 26. – P. 100707.

28. Davies D.L. A Cluster Separation Measure / D.L. Davies, D.W. Bouldin // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 1979. – Vol. PAMI-1(2). – P. 224–227.

29. Calinski T. A Dendrite method for cluster analysis / T. Calinski, J. Harabasz // Communications in Statistics. – 1974. – Vol. 3. – P. 1–27.

30. KEEL – Knowledge Extraction based on Evolutionary Learning. – Access mode: http://www.keel.es (accessed: 29.04.2022).

31. Мех М.А. Сравнительный анализ применения методов дифференциальной эволюции для оптимизации параметров нечетких классификаторов / М.А. Мех, И.А. Ходашинский // Изв. Российской академии наук. Теория и системы управления. – 2017. – № 4. – С. 65–75.

Остапенко Роман Олегович
Студент каф. комплексной информационной безопасности электронно-вычислительных систем (КИБЭВС) ТУСУРа
Ленина пр-т, 40, г. Томск, Россия, 634050
Тел.: +7-923-408-34-21
Эл. почта: romanOstpub@mail.ru

Ходашинский Илья Александрович
Д-р техн. наук, профессор каф. КИБЭВС ТУСУРа
Ленина пр-т, 40, г. Томск, Россия, 634050
ORCID: 0000-0002-9355-7638
Тел.: +7 (382-2) 70-15-29
Эл. почта: hodashn@gmail.com

Ostapenko R.O., Hodashinsky I.A.

Setting a rule base for a fuzzy classifier using the grasshopper optimization algorithm and the clustering algorithm

The article presents a description of a hybrid algorithm for generating fuzzy rules for a fuzzy classifier using grasshopper optimization algorithm and the K-means data clustering algorithm. The performance of clustering was evaluated by three fitness functions: total variance, Davis–Bouldin index, and Calinski–Harabasz index. Triangular and Gaussian membership functions have been investigated. The efficiency of the generated fuzzy rule bases has been tested on real datasets. The best combination is to use the total variance as the fitness function and the Gaussian function as the membership function.

Keywords: fuzzy classifier, clustering, *K*-means, grasshopper optimization algorithm.
DOI: 10.21293/1818-0442-2022-25-2-31-36

References

1. Moral J.M.A., Castiello C., Magdalena L., Mencar C. *Explainable Fuzzy Systems: Paving the Way from Interpretable Fuzzy Systems to Explainable AI Systems.* Cham, Springer, 2021, 253 p.

2. Khodashinsky I.A., Dudin P.A. Parametric fuzzy model identification based on a hybrid ant colony algorithm. *Optoelectronics, Instrumentation and Data Processing*, 2008, vol. 44, no. 5, pp. 402–411.

3. Hodashinsky I.A., Sarin K.S. Feature selection for classification through population random search with memory. *Automation and Remote Control*, 2019, vol. 80, no. 2, pp. 324–333.

4. Hodashinsky I.A., Sarin K.S. Feature selection: comparative analysis of binary metaheuristics and population based algorithm with adaptive memory. *Programming and Computer Software*, 2019, vol. 45, no. 5, pp. 221–227.

5. Koryshev N.P., Hodashinsky I.A. [Algorithm to forming a rule base for a fuzzy classifier de-signed on the basis of the K-means clustering algorithm and the whale optimization algorithm]. *Proceedings of TUSUR University*, 2021, vol. 24, no. 1, pp. 42–47 (in Russ.).

6. Bardamova M.B., Hodashinsky I.A. Algorithm and the shuffled frog leaping algorithm for imbalanced data with two classes. *Optoelectronics, Instrumentation and Data Processing*, 2021, vol. 57, no. 4, pp. 378–387.

7. Hodashinsky I.A. [Identification of fuzzy systems based on the annealing simulation algorithm and methods based on derivatives]. *Information Technologies*, 2012, no. 3, pp. 14–20 (in Russ.).

8. Hodashinsky I.A., Gorbunov I.V. [Optimization of parameters of fuzzy systems based on the modified bee colony algorithm]. *Mechatronics, Automation, Control*, 2012, no. 10, pp. 15–20 (in Russ.).

9. Xu R., Wunsch D.C. *Clustering*. Hoboken, John Wiley & Sons, Inc., 2009, 357 p.

10. Abraham A., Das S., Roy S. *Swarm intelligence algorithms for data clustering*. Soft Computing for Knowledge Discovery and Data Mining. 2007, pp. 279–313.

11. Nanda S.J., Panda G. A survey on nature inspired metaheuristic algorithms for partitional clustering. *Swarm and Evolutionary Computation*, 2014, vol. 16, pp. 1–18.

12. Xu X., Li J., Zhou M.C., Xu J., Cao J. Accelerated Two-Stage Particle Swarm Optimization for Clustering Not-Well-Separated Data. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2020, vol. 50, no. 11, pp. 4212–4223.

13. Sharma M., Chhabra J.K. An efficient hybrid PSO polygamous crossover based clustering algorithm. *Evolutionary Intelligence*, 2021, vol. 14, pp. 1213–1231.

14. Deb S., Tian Z., Fong S., Wong R., Millham R., Wong K.K.L. Elephant search algorithm applied to data clustering. *Soft Computing*, 2018, vol. 22, pp. 6035–6046.

15. Voronin D.A., Hodashinsky I.A. [*The Crow Algorithm for Optimizing the Parameters of a Fuzzy Classifier*. Electronic means and control systems. Materials of reports of the International scientific-practical conference], 2021. no 1-2, pp. 320–323.

16. Wolpert D., Macready W. No Free Lunch Theorems for Optimization. *IEEE Transactions on Evolutionary Computation*, 1997, vol. 1, no. 1, pp. 67–82.

17. Saremi S., Mirjalilia S., Lewis A., Saremieta S. Grasshopper Optimisation Algorithm: Theory and application. *Advances in Engineering Software*, 2017, vol. 105, pp. 30–47.

18. Nabavi S., Gholampour S., Haji M.S. Damage detection in frame elements using Grasshopper Optimization Algorithm (GOA) and time-domain responses of the structure. *Evolving Systems*, 2022, vol. 13, pp. 307–318.

19. Reddy K.N., Bojja P. A novel method to solve visual tracking problem: hybrid algorithm of grasshopper optimization algorithm and differential evolution. *Evolutionary Intelligence*, 2022, vol. 15, pp. 785–822.

20. Ram B.V.K., Chidambararaj N. Grasshopper optimization algorithm utilized Xilinx controller for maximum power generation in photovoltaic system. *Evolving Systems*, 2021, vol. 12, pp. 885–898.

21. Falehi A.D. Optimal robust disturbance observer-based sliding mode controller using multi-objective grasshopper optimization algorithm to enhance power system stability. *Journal of Ambient Intelligence and Humanized Computing*, 2020, vol. 11, pp. 5045–5063.

22. Dinh P.-H. A novel approach based on Grasshopper optimization algorithm for medical image fusion. *Expert Systems with Applications*, 2021, vol. 171, p. 114576.

23. Bhandari A.K., Rahul K. A novel local contrast fusion-based fuzzy model for color image multilevel thresholding using grasshopper optimization. *Applied Soft Computing*, 2019, vol. 81, p. 105515.

24. Zhang X., Miao Q., Zhang H., Wang L. A parameter-adaptive VMD method based on grasshopper optimization algorithm to analyze vibration signals from rotating machinery. *Mechanical Systems and Signal Processing*, 2018, vol. 108, pp. 58–72.

25. Malghan P.G., Hota M.K. Grasshopper optimization algorithm based improved variational mode decomposition technique for muscle artifact removal in ECG using dynamic time warping. *Biomedical Signal Processing and Control*, 2022, vol. 73, p. 103437.

26. Zakeri A., Hokmabadi A. Efficient feature selection method using real-valued grasshopper optimization algorithm. *Expert Systems with Applications*, 2019, vol. 119, pp. 61–72.

27. Kamel S.R., Yaghoubzadeh R. Feature selection using grasshopper optimization algorithm in diagnosis of diabetes disease. *Informatics in Medicine Unlocked*, 2021, vol. 26, p. 100707.

28. Davies D.L., Bouldin D.W. A Cluster Separation Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1979, vol. PAMI-1(2), pp. 224–227.

29. Calinski T. A, Harabasz J. Dendrite method for cluster analysis. *Communications in Statistics*, 1974, vol. 3, pp. 1–27.

30. KEEL – Knowledge Extraction based on Evolutionary Learning. – Access mode: http://www.keel.es. (Accessed: 29.04.2022).

31. Mekh M.A., Hodashinsky I.A. Comparative analysis of differential evolution methods to optimize parameters of fuzzy classifiers. *Journal of Computer and Systems Sciences International*, 2017, vol. 56, no 4, pp. 616–626.

Roman O. Ostapenko
Student, Department of Complex Information Security
of Computer Systems Tomsk State University
of Control Systems and Radioelectronics (TUSUR)
40, Lenin pr., Tomsk, Russia, 634050
Phone: +7-923-408-34-21
Email: romanOstpub@mail.ru

Ilya A. Hodashinsky
Doctor of Science in Engineering, Professor,
Department of Complex Information Security
of Computer Systems, TUSUR
40, Lenin pr., Tomsk, Russia, 634050
ORCID: https://orcid.org/0000-0002-9355-7638
Phone: +7 (382-2) 70-15-29
Email: hodashn@gmail.com