Перевод статьи:
Hodashinsky I.A., Samsonov S.S. Design of fuzzy rule based classifier using the monkey algorithm
Business Informatics. 2017. No. 1 (39). P. 61–67.

Построение нечеткого классификатора на основе алгоритма обезьян[1]

И.А. Ходашинский
доктор технических наук
профессор кафедры комплексной информационной безопасности электронно-вычислительных систем
Томский государственный университет систем управления и радиоэлектроники (ТУСУР)
Адрес: 634050, г. Томск, пр-т Ленина, д. 40
E-mail: hodashn@rambler.ru

С.С. Самсонов
студент кафедры комплексной информационной безопасности электронно-вычислительных систем
Томский государственный университет систем управления и радиоэлектроники (ТУСУР)
Адрес: 634050, г. Томск, пр-т Ленина, д. 40
E-mail: samsonicx@mail.ru

Аннотация

В статье представлен подход к построению классификаторов на основе нечетких правил. Нечеткий классификатор состоит из ЕСЛИ-ТО правил с нечеткими антецедентами (ЕСЛИ-часть) и метками класса в консеквентах (ТО-часть). Антецедентные части правил разбивают входное пространство признаков на множество нечетких областей, а консеквенты задают выход классификатора, помечая эти области меткой класса. Выделены два основных этапа построения классификатора: генерация базы нечетких правил и оптимизация параметров антецедентов правил. Формирование структуры классификатора выполнялась алгоритмом генерации базы правил по экстремальным значениям признаков, найденным в обучающей выборке. Особенность данного алгоритма заключается в том, что он генерирует по одному классифицирующему правилу на каждый класс. База правил, сформированная данным алгоритмом, имеет минимально возможный размер при классификации заданного набора данных. Оптимизация параметров антецедентов нечетких правил выполнена с помощью адаптированного для этих целей алгоритма обезьян, основанного на наблюдениях за передвижением обезьян в горной местности. В процессе работы алгоритма выполняются три оператора: движение вверх, локальный прыжок и глобальный прыжок. Одним из достоинств алгоритма при решении задач оптимизации большой размерности является вычисление псевдо-градиента целевой функции, причем вне зависимости от размерности на каждой итерации выполнения алгоритма требуется вычислить только два значения целевой функции.

Эффективность нечетких классификаторов, построенных с помощью предложенных алгоритмов, проверена на реальных данных из хранилища KEEL. Проведен сравнительный анализ с известными алгоритмами-аналогами «D-MOFARC» и «FARC-HD». Число правил, используемых классификаторами, построенными с помощью разработанных алгоритмов, значительно меньше числа правил в классификаторах-аналогах при сопоставимой точности классификации, что указывает на возможно более высокую интерпретируемость классификаторов, построенных с использованием предлагаемого подхода.

Ключевые слова: нечеткий классификатор, оптимизация параметров, алгоритм обезьян, формирование базы правил.

Цитирование: Hodashinsky I.A., Samsonov S.S. Design of fuzzy rule based classifier using the monkey algorithm // Business Informatics. 2017. No. 1 (39). P. 61–67. DOI: 10.17323/1998-0663.2017.1.61.67.

[1] Исследование выполнено в рамках базовой части государственного задания министерства образования и науки Российской Федерации на 2017-2019 гг. Номер 8.9628.2017/БЧ

Перевод статьи:
Hodashinsky I.A., Samsonov S.S. Design of fuzzy rule based classifier using the monkey algorithm
Business Informatics. 2017. No. 1 (39). P. 61–67.

Введение

Нечеткие классификаторы относятся к классу нечетких систем, основанных на правилах. Классификаторы этого типа широко используются в современных бизнес-приложениях из-за их способности справляться с неопределенностью, неточностью и неполнотой информации [1], например, в таких областях как оценка кредитных рисков [2, 3], маркетинг [4, 5], электронный бизнес и электронная коммерция [6]. К достоинствам нечетких классификаторов можно отнести их хорошую интерпретируемость [7] и отсутствие допущений, необходимых для статистической классификации [8].

Построение нечетких классификаторов предполагает решение двух основных задач: генерации базы нечетких правил и оптимизации параметров антецедентов (ЕСЛИ-частей) правил. Для генерации базы нечетких правил чаще всего используются алгоритмы кластеризации, в результате чего формируется начальное, «грубое» приближение нечеткого классификатора. Процедура оптимизации параметров антецедентов правил или «тонкой» настройки выполняется, как правило, методами, основанными на производных, алгоритмами роевого интеллекта или эволюционных вычислений [1, 2, 9–14]. В настоящей работе для решения указанных задач предлагается использовать алгоритм генерации базы правил по экстремальным значениям признаков и алгоритм обезьян.

Применение алгоритма генерации базы правил по экстремальным значениям признаков позволяет уменьшить количество правил до минимума, равного числу классов, и тем самым повысить интерпретируемость полученного результата.

Алгоритм обезьян основан на наблюдениях за передвижением обезьян в горной местности [15]. В процессе выполнения алгоритма выполняются три оператора: движение вверх, локальный прыжок, глобальный прыжок. Одним из достоинств алгоритма при решении задач оптимизации большой размерности является вычисление псевдо-градиента целевой функции, причем вне зависимости от размерности на каждой итерации выполнения алгоритма требуется вычислить только два значения целевой функции [16].

Целью настоящей работы является описание алгоритмов построения нечетких классификаторов: алгоритма генерации базы правил по экстремальным значениям признаков и алгоритма обезьян. Применение указанных алгоритмов направлено на повышение точности при решении задач классификации при сохранении интерпретируемости полученного решения.

1. Постановка задачи

Пусть имеется универсум $U = (A, C)$, где $A = \{x_1, x_2, ..., x_n\}$ – множество входных признаков, $C = \{c_1, c_2, ..., c_m\}$ – множество классов. Пусть $\mathbf{X} = x_1 \cdot x_2 \cdot ... \cdot x_n \in \Re^n$ – n-мерное пространство значений признаков. Объект u в заданном универсуме характеризуются своим вектором значений признаков. Задача классификации заключается в предсказании класса объекта u по его вектору значений признаков.

Традиционный классификатор может быть определен как функция

$$f : \Re^n \to \{0,1\}^m,$$

где $f(\mathbf{x};\boldsymbol{\theta}) = (c_1, c_2, ..., c_m)$, причем $c_i = 1$, а $c_j = 0$ ($j \in [1, m]$, $i \neq j$), когда объект, заданный вектором $\mathbf{x}$, принадлежит к классу c_i;

$\boldsymbol{\theta}$ – вектор параметров классификатора.

Нечеткий классификатор может быть представлен в виде функции, которая присваивает точке в пространстве входных признаков метку класса с вычисляемой степенью уверенности:

$$f : \Re^n \to [0,1]^m.$$

Основой нечеткого классификатора является продукционное правило следующего вида:

$$R_{ij}: \text{ЕСЛИ } x_1=A_{1i} \text{ И } x_2=A_{2i} \text{ И } x_3=A_{3i} \text{ И ... И } x_n=A_{ni}$$
$$\text{ТО class}=c_j,$$

где A_{ki} – нечеткий терм, характеризующий k-й признак в i-м правиле ($i \in [1, R]$);

R – число правил.

В настоящей работе класс определяется по принципу «команда победителей получает все»:

$$\text{class} = c_{j^*},\ j^* = \arg\max_{1 \le j \le m} \beta_j,$$

где $\beta_j(\mathbf{x}) = \sum_{R_{ij}} \prod_{k=1}^{n} \mu_{A_{jik}}(x_k),\ j = 1,2,...,m;$

$\mu_A(\cdot)$ – функция принадлежности нечеткого терма A.

Пусть имеется таблица наблюдений $\{(x_p; c_p), p = 1, ..., Z\}$. Определим следующую единичную функцию:

$$delta(p,\boldsymbol{\theta}) = \begin{cases} 1, \text{ if } c_p = f(c_p,\boldsymbol{\theta}) \\ 0, \quad \text{otherwise} \end{cases},\ p = 1,2,...,Z.$$

Перевод статьи:
Hodashinsky I.A., Samsonov S.S. Design of fuzzy rule based classifier using the monkey algorithm
Business Informatics. 2017. No. 1 (39). P. 61–67.

Тогда функция пригодности или мера точности классификации может быть выражена следующим образом:

$$E(\boldsymbol{\theta}) = \frac{\sum_{p=1}^{Z} delta(p, \boldsymbol{\theta})}{Z}.$$

Проблема построения нечеткого классификатора сводится к поиску максимума указанной функции в пространстве $\boldsymbol{\theta} = (\theta_1, \theta_2, ..., \theta_D)$:

$$\max(E(\boldsymbol{\theta})), \theta_i \in \{\Theta_i : \Theta_{i,\min} < \Theta_i < \Theta_{i,\max}, i = 1,2, ..., D\},$$

где Θ_i – значение параметра θ_i из интервала $[\Theta_{i,\min}, \Theta_{i,\max}]$;

$\Theta_{i,\min}$, $\Theta_{i,\max}$ – соответственно нижняя и верхняя границы каждого параметра.

На *рисунке 1* приведен пример, поясняющий формирование вектора $\boldsymbol{\theta}$. Здесь переменная x_1 представлена тремя треугольными термами, каждый из которых задается тремя параметрами (a, b, c), входящими в вектор $\boldsymbol{\theta} = (a_{11}, b_{11}, c_{11}, a_{12}, b_{12}, c_{12}, a_{13}, b_{13}, c_{13}, ... a_{21}, b_{21}, c_{21}, ...)$.

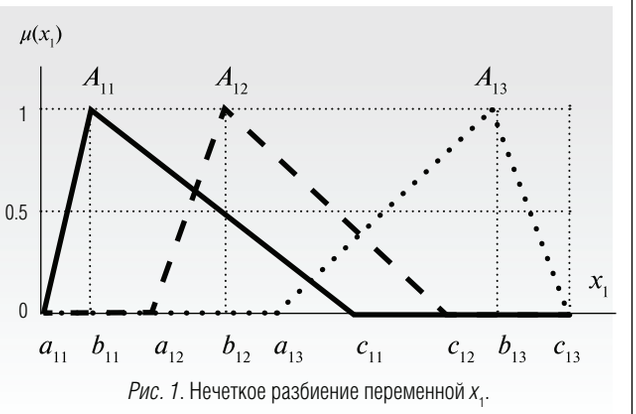

Рис. 1. Нечеткое разбиение переменной x_1.

Для нахождения оптимальных параметров $\boldsymbol{\theta}$ предлагается использовать алгоритм обезьян.

2. Алгоритм обезьян

Алгоритм обезьян (АО) – метаэвристический алгоритм оптимизации, имитирующий передвижение популяции обезьян в горной местности. Семь основных этапов алгоритма рассмотрены ниже.

1) Представление решения

Сначала определяется M – численность популяции обезьян, в которой позиция каждой i-й обезьяны представляет решение, задаваемое вектором $\boldsymbol{\theta}_i = (\theta_{i1}, \theta_{i2}, ..., \theta_{iD})$, $i = 1,..., M$.

2) Инициализация популяции

Возможные позиции обезьян в D-мерном гиперкубе генерируются случайным образом либо решения-позиции задаются пользователем. Возможна и смешанная стратегия инициализации, когда одна часть популяции задается пользователем, а другая – генерируется случайным образом.

3) Движение вверх

I. Для каждой i-й обезьяны генерируется вектор $\Delta\boldsymbol{\theta}_i = (\Delta\theta_{i1}, \Delta\theta_{i2}, ..., \Delta\theta_{iD})$,

где $\Delta\theta_{ij} = \begin{cases} a, \text{ if rand}(0;1) \geq 0,5 \\ -a, \text{if rand}(0;1) < 0,5 \end{cases}, i = 1,..., M, j = 1,..., D,$

$a > 0$ – длина шага.

II. Вычислить

$$E'_{ij}(\boldsymbol{\theta}_i) = \frac{E(\boldsymbol{\theta}_i + \Delta\boldsymbol{\theta}_i) - E(\boldsymbol{\theta}_i - \Delta\boldsymbol{\theta}_i)}{2\Delta\theta_{ij}}, \; i = 1,..., M, \; j = 1,..., D.$$

Вектор $\mathbf{E}'_i(\boldsymbol{\theta}_i) = (E'_{i1}(\boldsymbol{\theta}_i), E'_{i2}(\boldsymbol{\theta}_i), ..., E'_{iD}(\boldsymbol{\theta}_i))$ является псевдо-градиентом функции пригодности $E(\cdot)$ в точке $\boldsymbol{\theta}_i$.

III. Вычислить $z_j = \theta_{ij} + a \cdot \text{sign}(E'_{ij}(\boldsymbol{\theta}_i))$ и сформировать вектор $\mathbf{z} = (z_1, z_2, ..., z_D)$.

IV. Если полученный вектор-решение $\mathbf{z}$ не противоречит ограничениям построения нечеткого классификатора, то вектор $\boldsymbol{\theta}_i$ заменяется вектором $\mathbf{z}$, иначе вектор $\boldsymbol{\theta}_i$ остается неизменным.

V. Шаги I–IV повторять заданное число раз.

4) Локальный прыжок

I. Из случайно сгенерированных равномерно распределенных действительных чисел из диапазона $(\theta_{ij} - b, \theta_{ij} + b)$ сформировать вектор $\mathbf{z} = (z_1, z_2, ..., z_D)$, здесь b – параметр, характеризующий способность обезьяны вести наблюдения.

II. Если значение $E(\mathbf{z}) > E(\boldsymbol{\theta}_i)$ и вектор $\mathbf{z}$ не противоречит требованиям построения нечеткого классификатора, то вектор $\boldsymbol{\theta}_i$ заменяется вектором $\mathbf{z}$.

III. Шаги I–II повторять заданное число раз.

5) Глобальный прыжок

I. Сгенерировать случайное равномерно распределенное действительное число α из интервала $[c, d]$, где c, d – параметры алгоритма.

II. Вычислить $z_j = \theta_{ij} + \alpha \cdot (\rho_j - \theta_{ij})$, $j = 1,2,..., D$,

где $\rho_j = \frac{\sum_{i=1}^{M} \theta_{ij}}{M}$, $i = 1,..., M$, $j = 1,..., D$.

III. Если полученный вектор $\mathbf{z} = (z_1, z_2,..., z_D)$ не противоречит требованиям построения нечеткого классификатора и значение $E(\mathbf{z}) > E(\boldsymbol{\theta}_i)$, то вектор $\boldsymbol{\theta}_i$ заменяется вектором $\mathbf{z}$, иначе вектор $\boldsymbol{\theta}_i$ остается неизменным.

IV. Шаги I–III повторять заданное число раз.

6) Повтор N раз операторов движения вверх, локального и глобального прыжка

7) Вывод лучшего решения

3. Алгоритм генерации базы правил по экстремальным значениям признаков

Алгоритм генерации базы правил по экстремальным значениям признаков (ЕС) предназначен для формирования начальной базы правил нечеткого классификатора, содержащей по одному правилу на каждый класс. Правила формируются на основе экстремальных значений обучающей выборки $\{(\mathbf{x}_p; t_p), p = 1 ,..., Z\}$. Введем следующие обозначения: m – число классов, n – число признаков, $\boldsymbol{\Omega}^*$ – база правил классификатора.

Вход: m, {(xp; tp)}.
Выход: База правил классификатора $\boldsymbol{\Omega}^*$.
$\boldsymbol{\Omega} := \varnothing$;
цикл по j от 1 до m
  цикл по k от 1 до n
    поиск min $class_{jk} := \min(x_{pk})$;
    поиск max $class_{jk} := \max(x_{pk})$;
    формирование нечеткого терма A_{jk},
    накрывающего интервал
    $[\min class_{jk}, \max class_{jk}]$;
  конец цикла
  создание правила R_{1j} на основе термов A_{jk},
  относящего наблюдение к классу
  с идентификатором c_j;
  $\boldsymbol{\theta}^* := \boldsymbol{\theta} \cup \{R_{1j}\}$;
конец цикла
вывод $\boldsymbol{\theta}^*$.

4. Эксперимент

Для оценки эффективности работы нечетких классификаторов, оптимизированных комбинацией вышеприведенных алгоритмов (ЕС+АО), были проведены тесты на наборах данных из репозитория KEEL, приведенных в *таблице 2.* Каждый набор данных был отобран в соответствии с одной из следующих групп:

✦ «малое количество признаков – малое количество экземпляров» (ММ): наборы данных с количеством признаков меньшим 13 и количеством экземпляров меньшим 1000;

✦ «малое количество признаков – большое количество экземпляров» (МБ): наборы данных с количеством признаков меньшим 13 и количеством экземпляров большим или равным 1000;

✦ «большое количество признаков – малое количество экземпляров» (БМ): наборы данных с количеством признаков большим или равным 13 и количеством экземпляров меньшим 1000;

✦ «большое количество признаков – большое количество экземпляров» (ББ): наборы данных с количеством признаков большим или равным 13 и количеством экземпляров большим или равным 1000.

Таблица 1.

Описание наборов данных

Группа	Название	Признаки	Экземпляры	Классы
ММ	1. haberman	3	306	2
	2. iris	4	150	3
	3. balance	4	625	3
	4. newthyroid	5	215	3
	5. bupa	6	345	2
	6. pima	8	768	2
	7. glass	9	214	7
	8. Wisconsin	9	683	2
МБ	9. banana	2	5300	2
	10. titanic	3	2201	2
	11. phoneme	5	5404	2
	12. magic	10	19020	2
	13. page-blocks	10	5472	5
БМ	14. wine	13	178	3
	15. Cleveland	13	297	5
	16. heart	13	270	2
	17. hepatitis	19	80	2
ББ	18. segment	19	2310	7
	19. twonorm	20	7400	2
	20. thyroid	21	7200	3

Эксперименты были реализованы в соответствии с принципом кросс-валидации, который предполагает разделение набора данных на обучающие и тестовые дынные. Согласно описанному принципу, каждый набор данных представлен группой файлов, являющихся тестовыми и обучающими выборками. В соответствии с этим в ходе экспериментов

Таблица 2.

Сравнение классификаторов на наборах данных KEEL

Набор данных		EC+MA			D-MOFARC			FARC-HD		
Группа	Название	#R	#L	#T	#R	#L	#T	#R	#L	#T
ММ	1. haberman	2	79.2	**73.8**	9.2	81.7	69.4	5.7	79.2	73.5
	2. iris	3	97.8	95.3	5.6	98.1	**96.0**	4.4	98.6	95.3
	3. balance	3	87.5	86.7	20.1	89.4	85.6	18.8	92.2	**91.2**
	4. newthyroid	3	97.5	90.7	9.5	99.8	**95.5**	9.6	99.2	94.4
	5. bupa	2	74.2	68.4	7.7	82.8	**70.1**	10.6	78.2	66.4
	6. pima	2	75.5	71.3	10.4	82.3	75.5	20.2	82.3	76.2
	7. glass	7	69.0	61.3	27.4	95.2	**70.6**	18.2	79.0	69.0
	8. wisconsin	2	96.8	96.4	9.0	98.6	**96.8**	13.6	98.3	96.2
МБ	9. banana	2	78.9	78.4	8.7	90.3	**89.0**	12.9	86.0	85.5
	10. titanic	2	78.5	78.0	10.4	78.9	78.7	4.1	79.1	**78.8**
	11. phoneme	2	79.9	79.3	9.3	84.8	**83.5**	17.2	83.9	82.4
	12. magic	2	81.2	81.0	32.2	86.3	**85.4**	43.8	85.4	84.8
	13. page-blocks	5	95.5	95.3	21.5	97.8	**97.0**	18.4	95.5	95.0
БМ	14. wine	3	99.0	**96.6**	8.6	100.0	95.8	8.3	100.0	95.5
	15. cleveland	5	60.7	57.1	45.6	90.9	52.9	42.1	82.2	**58.3**
	16. heart	2	75.8	74.4	18.7	94.4	**84.4**	27.8	93.1	83.7
	17. hepatitis	2	94.1	77.8	11.4	100.0	**90.0**	10.4	99.4	88.7
ББ	18. segment	7	85.8	84.0	26.2	98.0	**96.6**	41.1	94.8	93.3
	19. twonorm	2	97.5	**97.1**	10.2	94.5	93.1	60.4	96.6	95.1
	20. thyroid	3	99.6	**99.3**	5.9	99.3	99.1	4.9	94.3	94.1

построение классификатора осуществлялось на обучающих выборках, после чего производилась оценка точности на тестовых выборках. Итоговое значение точности на тестовых и обучающих данных определялось путем вычисления среднего значения.

Для всех наборов данных использовались треугольные функции принадлежности. В качестве параметров алгоритма было выбраны следующие: количество особей – 30, итераций движения вверх – 5, итераций прыжка – 5, итераций кувырка – 15, интервал локального прыжка – 0,5, границы для глобального прыжка – –0,5 и 0,5 для левой и правой границы соответственно.

В *таблице 2* приведены усредненные результаты экспериментального исследования алгоритма обезьян при построении нечетких классификаторов на полном наборе признаков, а также результаты работы алгоритмов-аналогов «D-MOFARC» и «FARC-HD» [11]; здесь #R – число правил, #L – процент правильной классификации на обучающей выборке, #T – процент правильной классификации на тестовой выборке, полужирным шрифтом выделены лучшие результаты.

Для оценки статистической значимости различий в точности и числе правил классификаторов, сформированных комбинацией алгоритмов EC+AO и классификаторов-аналогов, использован критерий парных сравнений Уилкоксона–Манна–Уитни.

Сравнительный анализ позволил сделать следующие выводы:

1) критерий Уилкоксона–Манна–Уитни указывает на значимое различие между числом правил в классификаторах на основе EC+AO и классификаторах-аналогах (*p-value* < 5E-8);

2) критерий Уилкоксона–Манна–Уитни указывает на отсутствие значимого отличия между точностью классификации в сравниваемых классификаторах.

Данные выводы приводят к следующему заключению: при статистически неразличимой точности сравниваемых классификаторов классификаторы, оптимизированные комбинацией алгоритмов EC+AO, являются предпочтительными в силу меньшего числа правил, что в конечном итоге указывает на их возможно более высокую интерпретируемость.

За рамками статьи остался важный вопрос сравнения алгоритмов по их вычислительной сложности, поскольку в статьях, в которых приводятся результаты предыдущих исследований, нет подробного описания алгоритмов и не приводятся результаты экспериментов, по которым можно было бы судить о вычислительной сложности этих подходов.

Заключение

В работе рассмотрены методы построения нечетких классификаторов. Формирование структуры классификатора выполнялось алгоритмом генерации базы правил по экстремальным значениям признаков. Для оптимизации параметров классификаторов был применен алгоритм обезьян.

Работоспособность нечетких классификаторов, настроенных приведенными алгоритмами, проверена на нескольких наборах данных из репозитария KEEL. Полученные классификаторы имеют хорошие способности к обучению (высокий процент правильной классификации на обучающих выборках) и не менее хорошие прогностические способности (высокий процент правильной классификации на тестовых выборках).

Количество правил, используемых классификаторами, построенными с помощью разработанных алгоритмов, значительно меньше количества правил в классификаторах-аналогах при сопоставимой точности классификации, что указывает на возможно более высокую интерпретируемость классификаторов, построенных на основе комбинации ЕС+АО. ■

Литература

1. Garcia-Galan S., Prado R.P., Exposito M.J.E. Rules discovery in fuzzy classifier systems with PSO for scheduling in grid computational infrastructures // Applied Soft Computing. 2015. No. 29. P. 424–435.
2. Gorzalczany M.B., Rudzinski F. A multi-objective genetic optimization for fast, fuzzy rule-based credit classification with balanced accuracy and interpretability // Applied Soft Computing. 2016. No. 40. P. 206–220.
3. Laha A. Building contextual classifiers by integrating fuzzy rule based classification technique and k-nn method for credit scoring // Advanced Engineering Informatics. 2007. No. 21. P. 281–291.
4. Zhao R., Chai C., Zhou X. Using evolving fuzzy classifiers to classify consumers with different model architectures // Physics Procedia. 2012. No. 25. P. 1627–1636.
5. Setnes M., Kaymak U. Fuzzy modeling of client preference from large data sets: An application to target selection in direct marketing // IEEE Transactions on Fuzzy Systems. 2001. No. 9. P. 153–163.
6. Meier A., Werro N. A fuzzy classification model for online customers // Informatica. 2007. No. 31. P. 175–182.
7. Горбунов И.В., Ходашинский И.А. Методы построения трехкритериальных парето-оптимальных нечетких классификаторов // Искусственный интеллект и принятие решений. 2015. № 2. С. 75–87.
8. Scherer R. Multiple fuzzy classification systems // Studies in Fuzziness and Soft Computing. Vol. 288. Berlin: Springer-Verlag, 2012.
9. Ходашинский И.А. Идентификация нечетких систем на базе алгоритма имитации отжига и методов, основанных на производных // Информационные технологии. 2012. № 3. С. 14–20.
10. Antonelli M., Ducange P., Marcelloni F. An experimental study on evolutionary fuzzy classifiers designed for managing imbalanced datasets // Neurocomputing. 2014. No. 146. P. 125–136.
11. Fazzolari F., Alcala R., Herrera F. 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. 2014. No. 24. P. 470–481.
12. Ходашинский И.А., Горбунов И.В. Оптимизация параметров нечетких систем на основе модифицированного алгоритма пчелиной колонии // Мехатроника, автоматизация, управление. 2012. № 10. С. 15–20.
13. Ходашинский И.А., Дудин П.А. Идентификация нечетких систем на основе прямого алгоритма муравьиной колонии // Искусственный интеллект и принятие решений. 2011. № 3. С. 26–33.
14. Ходашинский И.А., Дудин П.А. Параметрическая идентификация нечетких моделей на основе гибридного алгоритма муравьиной колонии // Автометрия. 2008. № 5 (44). С. 24–35.
15. Zhao R., Tang W. Monkey algorithm for global numerical optimization // Journal of Uncertain Systems. 2008. No. 2. P. 165–176.
16. Zheng L. An improved monkey algorithm with dynamic adaptation // Applied Mathematics and Computation. 2013. No. 222. P. 645–657.