

Отбор признаков и построение нечеткого классификатора на основе алгоритма прыгающих лягушек¹

Аннотация. В работе рассматривается подход к проектированию классификаторов на основе нечетких правил. Для оптимизации параметров классификаторов применен непрерывный алгоритм прыгающих лягушек. На множестве построенных классификаторов производится выбор оптимального классификатора относительно точности и количества используемых признаков с помощью статистического информационного критерия Акаике. Эффективность предложенного подхода проверяется тестированием на наборах данных KEEL. Полученные данные сравниваются с результатами работы алгоритмов-аналогов. Предложенный в статье новый подход к проектированию нечетких классификаторов позволяет уменьшить число правил и признаков, тем самым повысить интерпретируемость результатов классификации.

Ключевые слова: нечеткий классификатор, оптимизация параметров, отбор признаков, алгоритм прыгающих лягушек.

Введение

Продукционные системы классификации успешно применяются в различных областях, включая медицину, биоинформатику, финансовые и технические приложения. Продукционное правило представляется в виде ЕСЛИ-ТО структуры, в которой ЕСЛИ-часть содержит описание объекта классификации, а в ТО-части указан класс, к которому должен быть отнесен объект. Такой формат легко поддается интерпретации, поскольку он использует язык, близкий к естественному языку. Поэтому, даже если другие методы классификации могут давать хорошие результаты, продукционные правила предпочтительнее других методов, при условии, что база правил не слишком велика и точность классификации достаточна. В реальных задачах области описания классификации объектов перекрываются, и их можно отнести к тому или иному классу с определенной степе-

ню уверенности. Если описание выполнено с использованием функций принадлежности, тогда речь идет о нечетком классификаторе.

Нечеткий классификатор является эффективным инструментом для изучения структур типа «вход-выход» [1, 2]. Кроме того, в областях исследования, подобных медицине, полученные результаты требуют объяснения. С этой точки зрения роль продукционных систем, использующих нечеткую логику, имеет особое значение из-за их способности представить модель интерпретируемым образом. Более того, использование нечеткой логики дает возможность моделирования неопределенности, связанной как с входными данными, так и с отношением между входными признаками и классом выходных данных [3].

Эффективность работы нечеткого классификатора зависит от алгоритма, используемого для его обучения. Цель любого обучения с учителем (supervised training) – минимизировать число неправильных ответов нечеткого классификатора.

¹ Работа выполнена в рамках базовой части государственного задания Минобрнауки России, проект 8.9628.2017/8.9.

Решение этой задачи будет выполнено только на оптимальном наборе параметров классификатора, для получения которых часто применяются метаэвристические методы [4-10].

Алгоритм прыгающих лягушек (АПР) – метаэвристика, основанная на эволюции мемов, сотрудничестве и конкуренции в популяции лягушек. Вся популяция лягушек делится на несколько групп или мемплексов. После поиска в каждом мемплексе лягушки из всех мемплексов перераспределяются, формируя новые мемплексы, что уменьшает шансы попадания в локальный оптимум [11, 12]. Алгоритм прост для понимания и реализации, имеет небольшое количество параметров, успешно применялся для решения задач комбинаторной и непрерывной оптимизации [13]. АПР использовался и для настройки параметров нечетких систем. Так, в работе [14] описано применение алгоритма АПР для оптимальной настройки параметров нечеткого контроллера. АПР успешно применен для выбора параметров модифицированных классификаторов на основе метода опорных векторов с двумя разделяющими гиперплоскостями [15, 16]. В работе [17] алгоритм используется для оптимизации весов и пороговых значений нейронной сети обратного распространения, которая применяется для ранней диагностики неисправностей подшипников качения.

Целью работы является описание построения нечеткого классификатора с помощью алгоритма прыгающих лягушек.

1. Постановка задачи

Пусть имеется универсум $U = (A, C)$, где $A = \{x_1, x_2, \dots, x_n\}$ – множество входных признаков, $C = \{c_1, c_2, \dots, c_m\}$ – множество классов. Пусть $\mathbf{X} = x_1 \times x_2 \times \dots \times x_n \in \mathfrak{R}^n$ – n -мерное про-

странство признаков. Объект u в заданном универсуме характеризуются своим вектором значений признаков. Задача классификации заключается в предсказании класса объекта u по его вектору значений признаков. Задача отбора признаков заключается в поиске на заданном множестве признаков A такого их подмножества, которое при уменьшении числа признаков не приводило бы к существенному уменьшению точности классификации. Решение представляется в виде вектора $\mathbf{S} = (s_1, s_2, \dots, s_n)$, где $s_i = 0$ свидетельствует о том, что i -й признак не участвует в классификации, $s_i = 1$ означает, что i -й признак используется классификатором. Для каждого подмножества признаков оценивается точность классификации.

Основная идея нечеткого классификатора состоит в описании предполагаемого кластера нечетким прототипом, размерность которого определена размерностью пространства исследуемых данных [3]. Таким образом, i -й кластер определяется нечетким правилом вида:

$$\text{ЕСЛИ } s_1 \wedge a_1 = A_{1i} \text{ И } s_2 \wedge a_2 = A_{2i} \text{ И } \dots \text{ И } s_n \wedge a_n = A_{ni} \\ \text{ТО class} = c_j,$$

где A_{ki} – нечеткий терм, характеризующий k -й признак в i -м правиле, $\{A_{ki} \mid k = 1, 2, \dots, n, i = 1, 2, \dots, R\}$ – нечеткое разбиение входного пространства $X_k = [x_{\min}, x_{\max}]$. Пример разбиения приведен на Рис. 1. Запись $s_i \wedge a_i$ указывает на наличие ($s_i = 1$) или отсутствие ($s_i = 0$) признака в классификаторе; R – число правил.

В процессе нечеткой классификации объект относится к каждому классу с определенной степенью принадлежности, вычисленной следующим образом:

$$\beta_j(x) = \sum_{R_{ij}} \prod_{k=1}^n A_{ki}(x_k), j = 1, 2, \dots, m.$$

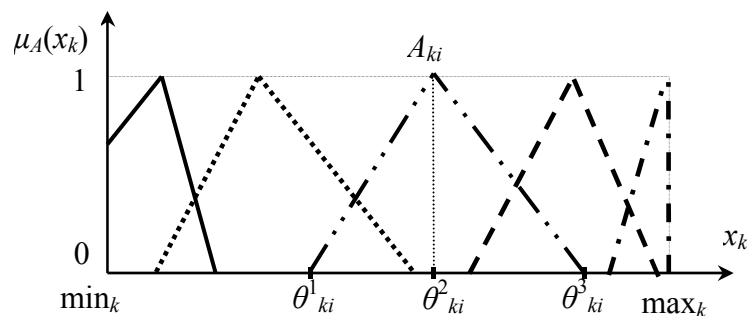


Рис. 1. Нечеткое разбиение переменной x_k

Выходом классификатора является метка класса, определяемая следующим образом:

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

Нечеткий классификатор может быть представлен функцией $c = f(\mathbf{x}, \boldsymbol{\theta})$, где $\boldsymbol{\theta}$ – вектор параметров, описывающий базу правил. На множестве обучающих данных (таблице наблюдений) $\{(\mathbf{x}_p; c_p), p = 1, \dots, z\}$ определим численный критерий качества классификации:

$$E(\boldsymbol{\theta}) = \frac{1}{z} \sum_{p=1}^z \begin{cases} 1, & \text{если } c_p = f(\mathbf{x}_p, \boldsymbol{\theta}) \\ 0, & \text{иначе} \end{cases}. \quad (1)$$

Проблема построения нечеткого классификатора сводится к поиску максимума функции качества. Для оптимизации вектора параметров $\boldsymbol{\theta}$ предлагается использовать непрерывный алгоритм прыгающих лягушек (FLA_R), а отбор признаков осуществлять дискретным алгоритмом прыгающих лягушек (FLA_D).

2. Алгоритм прыгающих лягушек. Оптимизации параметров классификатора

Алгоритм прыгающих лягушек (FLA) имитирует их поведение в процессе поиска пищи [11, 12]. Каждая виртуальная лягушка представляет собой решение $\boldsymbol{\theta}$ и включена в один из мемплексов. Процесс распределения виртуальных лягушек по мемплексам является особенностью алгоритма. Он проходит следующим образом: первая виртуальная лягушка определяется в первый мемплекс, вторая – во второй. Последующая лягушка переходит к третьему мемплексу. Так продолжается до тех пор, пока определенное количество мемплексов не будет

равномерно заполнено виртуальными лягушками. Пример распределения N виртуальных лягушек по трем группам представлен на Рис. 2.

Основой алгоритма является комбинирование локального поиска в пределах каждого из мемплексов и глобального поиска путем обмена информацией о положении лучших лягушек этих мемплексов и определения на этой основе глобально лучшего решения [11].

В непрерывном алгоритме прыгающих лягушек для решения задач оптимизации параметров нечетких систем используются следующие данные: F – популяция; UB, LB – верхняя и нижняя границы области определения признака; N – размер популяции; q – количество мемплексов; p – количество агентов в каждом мемплексе; T – количество глобальных итераций; $Iter$ – количество локальных итераций; $\boldsymbol{\theta}^{best}$, $\boldsymbol{\theta}^{worst}$ – лучшее и худшее решение; $\boldsymbol{\theta}_{global}$ – глобально лучшее решение; $rand$ – равномерно распределенное случайное число в интервале $[0,1]$. Ниже приведено пошаговое представление алгоритма оптимизации параметров нечеткого классификатора:

Шаг 1. Инициализировать популяцию $F = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dots, \boldsymbol{\theta}_N\}$.

Шаг 2. Вычислить значение функции качества каждого решения по формуле (1).

Шаг 3. Упорядочить решения в порядке убывания значения функции качества.

Шаг 4. Разделить виртуальных лягушек (решения) на мемплексы таким образом, что первая в отсортированном списке виртуальная лягушка попадает в первый мемплекс, вторая заносится во второй мемплекс.

Шаг 5. Так продолжается пока все лягушки не будут распределены в указанное количество мемплексов.

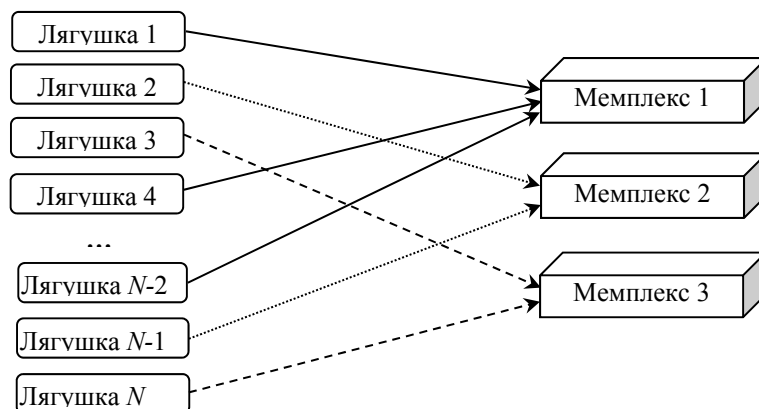


Рис. 2. Пример формирования мемплексов

Шаг 6. В каждом мемплексе найти лучшее θ^{best} и худшее θ^{worst} решение.

Шаг 7. Попытаться улучшить положение худшей виртуальной лягушки путем случайного перемещения ее в направлении лучшей лягушки, по формуле:

$$\theta_{new} = \text{rand} \cdot (\theta^{best} - \theta^{worst}) + \theta^{worst}.$$

Шаг 8. Если предыдущая операция не улучшает решение, то попытаться улучшить положение худшей виртуальной лягушки путем перемещения ее в направлении глобально лучшей лягушки, по формуле:

$$\theta_{new} = \text{rand} * (\theta^{global} - \theta^{worst}) + \theta^{worst}.$$

Шаг 9. Если и последняя операция не приводит к улучшению позиции виртуальной лягушки, то взамен ее случайным образом создать в области поиска новую лягушку (решение):

$$\theta_{new} := LB + \text{rand} * (UB - LB).$$

Шаг 10. Объединить виртуальных лягушек всех мемплексов в одну группу.

Шаг 11. Если условия завершения алгоритма не выполнены, то – переход к Шагу 3.

Шаг 12. Последняя глобально лучшая виртуальная лягушка будет являться решением задачи.

3. Отбор признаков. Дискретный алгоритм прыгающих лягушек

Отбор признаков является процессом нахождения оптимального подмножества признаков (атрибутов, свойств, характеристик, переменных) в соответствии с заданным критерием. Критерий определяется целью отбора признаков. Как правило, целью является повышение интерпретируемости и/или прогностической точности классификатора.

Методы отбора признаков делятся на три категории: фильтры (filters), обертки (wrappers) и встроенные (embedded) методы [18]. Метод фильтров основан на обобщенных свойствах обучающих данных и не включает в процесс отбора признаков собственно алгоритм построения классификатора. Достоинством данного метода является сравнительно невысокая вычислительная сложность, неплохая обобщающая способность и независимость от классификатора. Основной недостаток данного метода

заключается в том, что признаки, как правило, отбираются независимо друг от друга. Метод обертки включает построение классификатора в процесс отбора признаков и использует прогностическую точность классификатора для оценки эффективности отобранного подмножества признаков. Такое взаимодействие с классификатором, как правило, дает результаты лучшие, чем метод фильтра, однако при этом возрастает вычислительная сложность метода и существует риск переобучения. Встроенные методы выполняют отбор признаков в процессе обучения и включают алгоритм отбора в алгоритм построения классификатора. В работе [18] отмечается, что не существует лучшего метода отбора признаков, и усилия специалистов должны быть сосредоточены на поиске хорошего метода для каждой конкретной проблемы.

Поскольку задача отбора признаков может быть сформулирована как проблема оптимизации, для ее решения в нашей работе используется бинарный алгоритм прыгающих лягушек, в котором позиция каждой лягушки определяется двоичным вектором $S = (s_1, s_2, \dots, s_n)$, если $s_i = 0$, то i -й признак не участвует в классификации, а если $s_i = 1$, то он используется классификатором. Отличие непрерывного алгоритма прыгающих лягушек от дискретного заключается в цикле улучшения наихудшей позиции лягушки внутри каждого мемплекса. Указанный цикл состоит из четырех шагов. На первом применяется метод, подобный тому, что используется в дискретном алгоритме роящихся частиц [20], а на втором и третьем шаге выполняются операторы бинарного генетического алгоритма – мутация и кроссовер [19]. Ниже приведено описание описанного цикла.

Шаг 1. Рассчитать скорость худшей виртуальной лягушки в каждом мемплексе по формуле:

$$vw_{i+1}^d = (\omega \cdot vw_i^d + c \cdot r_1 \cdot (pb_i^d - xw_i^d) + k \cdot c \cdot r_2 \cdot (gb_i^d - xw_i^d) + c \cdot r_3 \cdot (xb_i^d - xw_i^d)),$$

где $d=1,2, \dots, n$ – индекс, идентифицирующий признак, i – текущая итерация алгоритма; xw_i – текущая позиция худшей виртуальной лягушки; pb_i – лучшая позиция, которую занимала худшая виртуальная лягушка в мемплексе; gb_i – позиция глобально лучшей виртуальной лягушки; xb_i – позиция лучшей виртуальной лягушки в мемплексе; r_1, r_2 и r_3 – равномерно распределенные случайные числа на интервале $[0, 1]$; $k \in \{-1, 1\}$ – случайно выбранное направ-

ление движения виртуальной лягушки, при $k=1$ движение в сторону глобально лучшей виртуальной лягушки, при $k=0$ – в противоположном направлении; $c = 2$ – константа; ω – инерционный вес, который рассчитывается по формуле:

$$\omega = 0,7 + \frac{\text{rand}}{2} \cdot (1 - \exp(-i)).$$

Шаг 2. Применить оператор мутации к позиции худшей виртуальной лягушки.

Шаг 3. К позициям глобальной лучшей виртуальной лягушки и лучшей в мемплексе применить оператор кроссовера.

Шаг 4. Если все предыдущие операции не приводят к улучшению позиции виртуальной лягушки, то взамен ее случайным образом генерируется новая лягушка.

После каждого шага полученный вектор непрерывных значений трансформируется в двоичный вектор признаков $\mathbf{S}$ по формуле:

$$\mathbf{S} = (h(xw^1 + vw^1), h(xw^2 + vw^2), \dots, h(xw^n + vw^n)),$$

$$\text{где } h(x) = \begin{cases} 1, & \text{если } x \geq 0 \\ 0, & \text{иначе} \end{cases}.$$

4. Этапы построения нечеткого классификатора

Построение нечеткого классификатора включает следующие этапы: отбор информативных признаков; формирование структуры классификатора; оптимизация параметров классификатора; выбор классификатора, оптимального по точности и сложности.

На первом этапе по принципу «оберток» бинарным алгоритмом прыгающих лягушек формируются группы информативных признаков на классифицируемом наборе данных. Поскольку присутствующая в метаэвристике случайность способна привести к отбору различных информативных признаков, формируется несколько групп признаков.

На втором этапе на каждой отобранной группе признаков генерируется база правил нечетких классификаторов с использованием алгоритма формирования структуры, описание которого приведено в [8]. Сгенерированные структуры имеют минимально возможное число правил, которое ограничено количеством существующих классов в классифицируемом наборе данных. Ба-

зы правил, построенные указанным образом, просты для понимания и последующей интерпретации полученного результата.

На следующем этапе проводится оптимизация параметров каждого классификатора с помощью непрерывного алгоритма прыгающих лягушек. На множестве построенных классификаторов производится выбор оптимального классификатора относительно точности и количества используемых признаков с помощью статистического информационного критерия Акаике (AIC – Akaike Information Criterion) [21, 22]. Вычисление критерия производится по формуле, адаптированной для нахождения компромисса между точностью и сложностью (количеством используемых признаков) ниже:

$$AIC = \ln ER_{TR} + 2/|Tr| (1 + cF_S),$$

где ER_{TR} – ошибка классификатора на обучающей выборке; F_S – число признаков, используемых классификатором; $|Tr|$ – объем обучающей выборки; c – коэффициент штрафа.

Оптимальный классификатор обладает наименьшим значением AIC .

5. Эксперимент

Для оценки эффективности работы нечетких классификаторов, построенных с помощью алгоритмов прыгающих лягушек, были проведены экспериментальные исследования на 15 известных наборах реальных данных из репозитория KEEL (<http://www.keel.es>). Наборы разбиты на две группы, к первой группе относятся наборы данных с количеством признаков меньше десяти, во второй группе – наборы данных, у которых количество признаков больше десяти. Описание указанных наборов приведено в Табл. 1.

Эксперимент проводился по принципу кроссвалидации, согласно которому каждый набор данных разделен на обучающие и тестовые выборки. Для их формирования набор данных разделялся на 10 равных частей, из девяти частей формировалась обучающая выборка, из одной части – тестовая. Таким образом было сформировано 10 пар выборок.

На каждой обучающей выборке с помощью дискретного алгоритма прыгающих лягушек были отобраны информативные признаки, на которых выполнена инициализация нечеткого классификатора алгоритмом формирования структуры, далее по 10 раз запускался непре-

Табл. 1. Описание наборов данных

Группа	Название набора данных	Условное обозначение	Число признаков	Число экземпляров	Число классов
I	1. haberman	hbr	3	306	2
	2. iris	irs	4	150	3
	3. balance	blc	4	625	3
	4. newthyroid	nth	5	215	3
	5. bupa	bp	6	345	2
	6. pima	pm	8	768	2
	7. glass	gl	9	214	7
	8. wisconsin	wsn	9	683	2
II	9. wine	wn	13	178	3
	10. cleveland	cl	13	297	5
	11. segment	sgm	19	2310	7
	12. ring	rng	20	7400	2
	13. twonorm	twm	20	7400	2
	14. thyroid	thr	21	7200	3
	15. spambase	spm	57	4597	2

рывный алгоритм для оптимизации параметров классификатора. Число признаков и усредненное по 10 запускам и 10 выборкам значение процента правильной классификации характеризуют эффективность алгоритма оптимизации на используемом наборе данных.

Для дискретного алгоритма прыгающих лягушек были определены следующие параметры: 50 глобальных итераций, 5 локальных итераций, 25 мемплексов, 10 виртуальных лягушек в каждом мемплексе. Для непрерывного алгоритма прыгающих лягушек выбраны следующие параметры: 100 глобальных итераций, 10 локальных итераций, 25 мемплексов, 15 виртуальных лягушек в каждом мемплексе.

В Табл. 2 и Табл. 3 приведены усредненные результаты экспериментального исследования алгоритмов прыгающих лягушек при построении нечетких классификаторов, а также результаты работы алгоритмов-аналогов D-MOFARC и FARC-HD [23]. Полужирным шрифтом выделены лучшие результаты. Алгоритмы FLA_R,

D-MOFARC и FARC-HD отработали на полном наборе признаков, а число признаков для алгоритма FLA_D+FLA_R приведено в последней строке таблиц.

Для оценки статистической значимости различий в точности и числе признаков, сформированных алгоритмами прыгающих лягушек, и классификаторов-аналогов использованы многовыборочный критерий Краскела-Уоллиса и критерий парных сравнений Уилкоксона.

Сравнительный анализ позволил сделать следующие выводы:

1) критерий Уилкоксона указывает на значимое отличие между числом признаков в классификаторах на основе FLA_D+FLA_R и классификаторах-аналогах ($p\text{-value} < 8,73E-4$);

2) критерий Краскела-Уоллиса указывает на отсутствие значимой разницы между точностью классификации в сравниваемых классификаторах.

Табл. 2. Точность классификации наборов данных из первой группы

Группа I		Наборы данных							
		hbr	irs	blc	nth	bp	pm	gl	wsn
FARC-HD	Обучение	79.2	98.6	92.2	99.2	78.2	82.3	79.0	98.3
	Тестирование	73.5	95.3	91.2	94.4	66.4	76.2	69.0	96.2
D-MOFARC	Обучение	81.7	98.1	89.4	99.8	82.8	82.3	95.2	98.6
	Тест	69.4	96.0	85.6	95.5	70.1	75.5	70.6	96.8
FLA _R	Обучение	79.7	99.5	94.2	99.0	78.8	79.0	72.2	97.7
	Тестирование	71.6	97.3	92.1	98.1	72.8	75.6	64.0	94.9
FLA _D + FLA _R	Обучение	79.7	99.7	94.2	99.0	81.3	78.0	89.2	98.3
	Тестирование	71.6	98.4	92.1	98.7	78.2	75.8	74.3	94.0
	Число признаков	3	2	3	3	3	6	5	3

Табл.3. Точность классификации на наборах данных из второй группы

Группа II		Наборы данных						
		wn	cl	sgm	rng	twm	thr	spm
FARC-HD	Обучение	100.0	82.2	94.8	95.1	96.6	94.3	92.4
	Тестирование	95.5	58.3	93.3	94.0	95.1	94.1	91.6
D-MOFARC	Обучение	100	90.9	98.0	94.2	94.5	99.3	91.7
	Тест	95.8	52.9	96.6	93.3	93.1	99.1	90.5
FLA _R	Обучение	99.9	67.3	82.4	97.4	96.9	95.1	76.6
	Тестирование	98.3	57.9	82.1	97.0	96.7	94.7	76.4
FLA _D +FLA _R	Обучение	99.0	68.1	82.9	77.2	97.4	95.7	77.5
	Тестирование	96.7	58.3	82.2	76.7	97.6	95.6	76.6
	Число признаков	7	7	9	3	19	9	31

3) критерий Уилкоксона указывает на отсутствие значимого отличия между точностью классификации при попарном сравнении классификаторов.

Эти выводы приводят к следующему заключению: при статистически неразличимой точности сравниваемых классификаторов классификаторы, сформированные алгоритмом FLA_D+FLA_R, являются предпочтительными в силу меньшего числа признаков, что в конечном итоге указывает на их меньшую вычислительную сложность и более высокую интерпретируемость.

Заключение

В работе рассмотрены методы построения нечетких классификаторов с отбором признаков. Отбор признаков выполнен с помощью дискретного алгоритма прыгающих лягушек. Формирование структуры классификатора выполнялась алгоритмом генерации базы правил по экстремальным значениям признаков. Для оптимизации параметров классификаторов был применен непрерывный алгоритм прыгающих лягушек.

Работоспособность нечетких классификаторов, настроенных приведенными алгоритмами, проверена на 15 наборах данных из репозитория KEEL. Полученные классификаторы имеют хорошие способности к обучению, подтвержденными высоким процентом правильной классификации на обучающих выборках, и не менее хорошие прогностические способности, о чем свидетельствует высокий процент правильной классификации на тестовых выборках.

Количество признаков, используемых классификаторами, построенными с помощью разработанных алгоритмов, значительно меньше полного количества признаков наборов данных.

Таким образом, разработанные алгоритмы построения классификаторов на основе комбинаций алгоритмов FLA_D+FLA_R позволяют построить нечеткие классификаторы, использующие меньшее количество признаков, но обладающие точностью на сокращенной группе признаков статистически неразличимой с точностью классификаторов, построенных на полной группе признаков.

В дальнейшем предполагается исследовать другие способы бинаризации алгоритма прыгающих лягушек и увеличить количество тестовых наборов данных.

Литература

- Guillaume S. Designing fuzzy inference systems from data: An interpretability-oriented review / S. Guillaume // IEEE Transactions on Fuzzy Systems. - 2001. - Vol. 9(3). - P. 426-443.
- Ducange P. Multi-objective genetic fuzzy classifiers for imbalanced and cost-sensitive datasets / P. Ducange, B. Lazzerini, F. Marcelloni // Soft. Comput. - 2010. - Vol. 14. - P. 713-728.
- Pota M. Designing rule-based fuzzy systems for classification in medicine / M. Pota, M. Esposito, G. D. Pietro // Knowl.-Based Syst. - 2017. - Vol. 124. - P. 105-132.
- Jamalabadi H. Competitive interaction reasoning: A bio-inspired reasoning method for fuzzy rule based classification systems / H. Jamalabadi, H. Nasrollahi, S. Alizadeh, B.N. Araabi, M.N. Ahamadabadi // Information Sciences. - 2016. - Vol. 352-353. - P. 35-47.
- Salcedo-Sanz S. Modern meta-heuristics based on nonlinear physics processes: A review of models and design procedures / S. Salcedo-Sanz // Physics Reports. - 2016. - Vol. 655. - P. 1-70.
- Meli P. Optimal design of fuzzy classification systems using PSO with dynamic parameter adaptation through fuzzy logic / P. Melin, F. Olivas, O. Castillo, F. Valdez, J. Soria, M. Valdez // Expert Systems with Applications. - 2013. - Vol. 40(8). - P. 3196-3206.
- Lahtasna A. An improved genetic-fuzzy system for classification and data analysis / A. Lahtasna, W. C. Seng // Expert Systems With Applications. - 2017. - Vol. 83. - P. 49-62.

8. Мех М.А. Сравнительный анализ применения методов дифференциальной эволюции для оптимизации параметров нечетких классификаторов // Известия Российской академии наук. Теория и системы управления. - 2017. - № 4. - С. 65-75.
9. Ходашинский И.А. Построение нечеткого классификатора на основе методов гармонического поиска // Программирование. - 2017. - № 1. - С. 54-65.
10. Ходашинский И.А. Алгоритмы муравьиной и пчелиной колонии для обучения нечетких систем // Доклады ТУСУР. - 2009. - № 2 (20). - С. 157-161.
11. Eusuff M. M. Optimization of water distribution network design using the shuffled frog leaping algorithm / M. M. Eusuff, K.E. Lansey // J. Water Resour. Plann. Manag. - 2003. - Vol. 129. - P. 210-225.
12. Eusuff M. M. Shuffled frog-leaping algorithm: a memetic meta heuristic for discrete optimization // Engineering Optimization. - 2006. - Vol. 38(2). - P. 129-154.
13. Wang L. Diversity Analysis of Population in Shuffled Frog Leaping Algorithm / L. Wang, Y. Gong // Y. Tan, Y. Shi, and H. Mo (Eds.): ICSI 2013, Part I, LNCS 7928. - Springer, 2013. - P. 24-31.
14. Nguyen D.-H. Comparing Convergence of PSO and SFLA Optimization Algorithms in Tuning Parameters of Fuzzy Logic Controller / D.-H. Nguyen, M.-D. Ngo // Lecture Notes in Electrical Engineering, 371. - Springer, 2016. - P. 457-467.
15. Wu F.L. Mixed kernel twin support vector machines based on the shuffled frog leaping algorithm / F.L. Wu, S.F. Ding, H.J. Huang, Z.B. Zhu // J. Comput. - 2014. - Vol. 9(4). P. 947-955.
16. Zhang X. Multi-class LSTMSVM based on optimal directed acyclic graph and shuffled frog leaping algorithm / X. Zhang, S. Ding, T. Sun // Int. J. Mach. Learn. & Cyber. - 2016. - Vol. 7. - P. 241-251.
17. Zhao Z. Improved shuffled frog leaping algorithm-based BP neural network and its application in bearing early fault diagnosis / Z. Zhao, Q. Xu, M. Jia // Neural Comput. & Applic. - 2016. - Vol. 27. - P. 375-385.
18. Bolon-Canedo V. Feature Selection for High-Dimensional Data / V. Bolon-Canedo, N. Sanchez-Marono, A. Alonso-Betanzos // Springer, 2015.
19. Vakil Baghmisheh M.T. A discrete shuffled frog optimization algorithm/ M.T. Vakil Baghmisheh, K. Madani, A. Navarbah // Artif. Intell. Rev. - 2011. Vol. 36. - P. 267-284.
20. Kennedy J. A discrete binary version of the particle swarm algorithm / J. Kennedy, R. C. Eberhart // Proceeding of the IEEE International Conference on System, Man, and Cybernetics. - IEEE, 1997. - P. 4104-4109.
21. Yen J. Application of Statistical Information Criteria for Optimal Fuzzy Model Construction / J. Yen // IEEE Transaction on Fuzzy Systems. - 1998. - Vol. 6. - P. 362-372.
22. Ходашинский, И.А. Построение компактных и точных нечетких моделей на основе статистических информационных критериев // Информатика и системы управления. - 2014. - № 1(39). - С. 99-107.
23. Fazzolari 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 / F. Fazzolari, R. Alcalá, F. Herrera // Applied Soft Computing. - 2014. - Vol. 24. - P. 470-481.

Ходашинский Илья Александрович. Профессор Томского государственного университета систем управления и радиоэлектроники (ТУСУР). Окончил Новосибирский электротехнический институт (НЭТИ) в 1975 году. Доктор технических наук. Количество печатных работ: более 150, в том числе две монографии. Область научных интересов: вычислительный интеллект, метаэвристики. E-mail: hodashn@rambler.ru

Бардамова Марина Борисовна. Младший научный сотрудник Лаборатории медико-биологических исследований ТУСУР. Окончила ТУСУР в 2017 году. Количество печатных работ: 15. Область научных интересов: вычислительный интеллект, машинное обучение, нечеткие системы. E-mail: 722bmb@gmail.com

Ковалев Виталий Сергеевич. Окончил ТУСУР в 2017 году. Количество печатных работ: 7. Область научных интересов: вычислительный интеллект, машинное обучение, нечеткие системы. E-mail: vitality_979@mail.ru

Using shuffled frog-Leaping algorithm for feature selection and fuzzy classifier design

I.A. Hodashinsky, M.B. Bardamova, I.V. Kovalev

This article, a new approach to designing fuzzy-rule-based classifier is considered. The fuzzy classifier construction includes the following stages: feature selection, generation of the classifier structure, optimization of the classifier parameters, and choice of the optimal classifier. At the first stage, according to the principle of "wrappers", binary shuffled frog-leaping algorithm are formed groups of relevant features on the classifiable data set. Since the randomness present in metaheuristics is capable of leading to the selection of various relevant features, several groups of features are formed. At the second stage, on each selected feature group, the structure of the fuzzy classifier is generated on the basis of the extrema of the observation table. The generated structures have the minimum possible number of rules, which is limited by the number of existing classes in the classified data set. The rules bases constructed in this way are simple for understanding and subsequent interpretation of the result obtained. The next step is to optimize the parameters of each classifier using a continuous shuffled frog-leaping algorithm. On a set of constructed classifiers, the optimal classifier is selected with respect to the accuracy and the number of features used, using the statistical Akaike informational criterion. The effectiveness of the proposed ap-

proach is tested on 15 KEEL data sets. The results obtained are statistically compared with the results of similar algorithms. The new approach to designing fuzzy classifiers proposed in the article allows to reduce the number of rules and attributes, thereby increasing the interpretability of classification results.

Keywords: fuzzy classifier, parameters optimization, features selection, shuffled frog-leaping algorithm.

References

- Guillaume S. 2001. Designing fuzzy inference systems from data: An interpretability-oriented review. *IEEE Transactions on Fuzzy Systems*. 9(3):426-443.
- Ducange P, Lazzerini B, Marcelloni F. 2010. Multi-objective genetic fuzzy classifiers for imbalanced and cost-sensitive datasets. *Soft. Comput.* 14:713-728.
- Pota M, Esposito M, Pietro GD. 2017. Designing rule-based fuzzy systems for classification in medicine. *Knowl.-Based Syst.* 124:105-132.
- Jamalabadi H, Nasrollahi H, Alizadeh S, Araabi BN, Ahamadabadi MN. 2016. Competitive interaction reasoning: A bio-inspired reasoning method for fuzzy rule based classification systems. *Information Sciences*. 352-353:35-47.
- Salcedo-Sanz S. 2016. Modern meta-heuristics based on nonlinear physics processes: A review of models and design procedures. *Physics Reports*. 655:1-70.
- Melin P, Olivas F., Castillo O., Valdez F., Soria J., Valdez M. 2013. Optimal design of fuzzy classification systems using PSO with dynamic parameter adaptation through fuzzy logic. *Expert Systems with Applications*. 40(8):3196-3206.
- Lahsasna A, Seng WC. 2017. An improved genetic-fuzzy system for classification and data analysis. *Expert Systems with Applications*. 83:49-62.
- Mekh MA, Hodashinsky IA. 2017. Comparative Analysis of differential evolution methods to optimize parameters of fuzzy classifiers. *Journal of Computer and Systems Sciences International*. 4(56):65-75.
- Hodashinsky IA, Mekh MA. 2017. Fuzzy classifier design using harmonic search methods. *Programming and computer software*. 1(43):54-65.
- Hodashinsky IA, Gorbunov I.V. & Dudin P.A. 2009. Algoritmy murav'inoi i pchelinoi kolonii dlia obucheniia nechetkikh sistem [Algorithms of ants and bee colony for training a fuzzy system]. *Doklady TUSUR [Proceedings of TUSUR]*, 20(2):157-161.
- Eusuff MM, Lansey KE. 2003. Optimization of water distribution network design using the shuffled frog leaping algorithm. *J. Water Resour. Plann. Manag.* 129:210-225.
- Eusuff MM, Lansey KE, Pasha F. 2006. Shuffled frog-leaping algorithm: a memetic meta heuristic for discrete optimization. *Engineering Optimization*. 38(2):129-154.
- Wang L, Gong Y. 2013. Diversity Analysis of Population in Shuffled Frog Leaping Algorithm. *ICSI 2013, Part I, LNCS* 7928. London. 24-31.
- Nguyen D-H, Ngo M-D. 2016. Comparing Convergence of PSO and SFLA Optimization Algorithms in Tuning Parameters of Fuzzy Logic Controller. *Lecture Notes in Electrical Engineering*, 371. London. 457-467.
- Wu FL, Ding SF, Huang HJ, Zhu ZB. 2014. Mixed kernel twin support vector machines based on the shuffled frog leaping algorithm. *J. Comput.* 9(4):947-955.
- Zhang X, Ding S, Sun T. 2016. Multi-class LSTMSVM based on optimal directed acyclic graph and shuffled frog leaping algorithm. *Int. J. Mach. Learn. & Cyber.* 7:241-251.
- Zhao Z, Xu Q, Jia M. 2016. Improved shuffled frog leaping algorithm-based BP neural network and its application in bearing early fault diagnosis. *Neural Comput. & Applic.* 27:375-385.
- Bolon-Canedo V, Sanchez-Marono N, Alonso-Betanzos A. 2015. Feature Selection for High-Dimensional Data. London: Springer. 142 p.
- Vakil Baghmisheh MT, Madani K, Navarbat A. 2011. A discrete shuffled frog optimization algorithm. *Artif. Intell. Rev.* 36:267-284.
- Kennedy J, Eberhart RC. 1997. A discrete binary version of the particle swarm algorithm. *Proceeding of the IEEE International Conference on System, Man, and Cybernetics*. IEEE. 4104-4109.
- Yen J. 1998. Application of Statistical Information Criteria for Optimal Fuzzy Model Construction. *IEEE Transaction on Fuzzy Systems*. 6:362-372.
- Hodashinsky IA. 2014. Postroenie kompaktnykh i tochnykh nechetkikh modelej na osnove statisticheskikh informacionnykh kriteriev [Design of compact and precise fuzzy models based on statistical information criteria]. *Informatika i sistemy upravleniya [Informatics and Control Systems]* 1(39):99-107.
- Fazzolari 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*. 24:470-481.

Hodashinsky I.A. Professor of the Tomsk State University of Control Systems and Radioelectronics, Department of Complex Information Security of Computer Systems, Tomsk, Russia. E-mail: hodashn@rambler.ru

Bardamova M.B. Junior research officer of Biomedical Research Laboratory in Tomsk State University of Control Systems and Radioelectronics, Tomsk, Russia. E-mail: 722bmb@gmail.com

Kovalev V.S. Tomsk State University of Control Systems and Radioelectronics, Tomsk, Russia. E-mail: vitya_979@mail.ru