УДК 004.85

ФОРМИРОВАНИЕ СТРУКТУРЫ НЕЧЁТКОГО КЛАССИФИКАТОРА КОМБИНАЦИЕЙ АЛГОРИТМА ЭКСТРЕМУМОВ КЛАССОВ И АЛГОРИТМА «ПРЫГАЮЩИХ ЛЯГУШЕК» ДЛЯ НЕСБАЛАНСИРОВАННЫХ ДАННЫХ С ДВУМЯ КЛАССАМИ

© **М. Б. Бардамова, И. А. Ходашинский**

Томский государственный университет систем управления и радиоэлектроники, 634050, г. Томск, просп. Ленина, 40
E-mail: 722bmb@gmail.com

Предложен способ применения метаэвристического алгоритма «прыгающих лягушек» в качестве инструмента для расширения первичной базы правил нечёткого классификатора. Такой алгоритм актуален в случае, когда имеющихся правил недостаточно для качественного распознавания всех классов, например при наличии дисбаланса в данных. Дополнительные правила, генерируемые метаэвристикой, способны не только улучшить качество классификации, но и предоставить более полное описание исследуемой предметной области. Для генерации первичной структуры классификатора был использован алгоритм, основанный на экстремальных значениях признаков классов. Исследуемая комбинация была проверена на 36 несбалансированных наборах данных из репозитория Knowledge Extraction based on Evolutionary Learning и показала увеличение средней геометрической точности на 34 наборах, а также удовлетворительные результаты по сравнению с аналогичными алгоритмами. Достоинства предложенного способа формирования структуры заключаются в отсутствии необходимости дополнения данных синтетическими образцами, низком разбросе результатов на отдельных запусках и возможности улучшить качество классификации при добавлении небольшого количества правил.

Ключевые слова: нечёткий классификатор, алгоритм «прыгающих лягушек», структура нечёткого классификатора, несбалансированные данные.

DOI: 10.15372/AUT20210407

Введение. Во многих реальных данных в задачах классификации наблюдается различие в количестве экземпляров классов. Такие данные называются несбалансированными. Стандартные алгоритмы классификации имеют тенденцию переобучаться на экземплярах классов большинства (отрицательные классы), что приводит к снижению качества определения образцов классов меньшинства (положительные классы) [1]. Однако во многих случаях именно определение редких классов является более приоритетной задачей [2, 3]. Например, системам обнаружения вторжений важно уметь выделять сетевые атаки среди большого объёма нормального трафика.

Нечёткие классификаторы также подвержены риску переобучения. В [4, 5] предлагалось использовать отбор признаков и настройку параметров термов с учётом дисбаланса классов. Эти способы помогают увеличить процент распознавания классов меньшинства, но их возможности ограничены заранее заданным количеством правил и термов. В приведённых работах используется минимальное количество правил, что в некоторых случаях ограничивает возможность достижения высокого качества классификации. В данном исследовании предлагается новый способ усовершенствования процесса формирования структуры нечёткого классификатора в целях увеличения качества классификации без этапа предобработки данных, традиционно применяемого для решения проблемы несбалансированности. Отличительными особенностями этого подхода являются применение

метаэвристического алгоритма для итерационного добавления правил к первично сгенерированной базе правил и использование фитнес-функции, направленной на достижение сбалансированной точности классификации. В представленной работе ограничимся проблемой классификации данных, включающих только два класса, так как любая задача классификации может быть сведена именно к такому виду.

Связанные работы. Под структурой нечёткого классификатора понимают совокупность двух элементов — базы нечётких правил и лингвистических термов. Качество созданной структуры, например полнота и достаточность правил, во многом влияет на эффективность обучения. Задачу создания структуры классификатора можно разбить на два этапа. На первом осуществляется нечёткое разбиение входного признакового пространства. На втором этапе на основе полученного разбиения формируются правила. Термы могут быть заданы экспертами, сгенерированы случайным или равномерным образом, созданы с помощью методов кластеризации или иных методов, анализирующих свойства данных. Рассмотрим далее наиболее известные алгоритмы построения структуры нечётких классификаторов.

Алгоритм Chi [6] осуществляет попытку восстановить зависимость между пространством признаков и множеством классов. На первом шаге данного алгоритма область определения признаков равномерно заполняется заданным количеством симметричных термов треугольного типа. На втором шаге для каждого объекта происходит составление нового правила из перечня тех термов, к которым объект имеет максимальную степень принадлежности. В консеквент записывается метка класса, принадлежащая данному объекту, затем определяется вес правила. В случае создания повторяющегося правила, оно будет удалено. Однако, если два правила будут иметь одинаковые антецеденты, но разные консеквенты, в наборе останется правило с наибольшим весом.

Алгоритм Ishibuchi, предложенный в [7], формирует антецедентную часть правила на основе всех возможных комбинаций нечётких термов, используя четыре заданных варианта разбиения признака (два, три, четыре и пять термов, равномерно распределённых между 0 и 1), а также терм «всё равно». Консеквент правила определяется по максимальному значению меры доверия. При расчёте доверия между i-м правилом и консеквентом c_i вычисляется отношение между суммой степеней принадлежности к этому правилу экземпляров с меткой класса c_i и суммой степеней принадлежности к этому правилу всех экземпляров набора данных. На основе доверия также рассчитываются веса правил. Для итоговой структуры отбирается заданное количество правил по наилучшим значениям мер доверия и поддержки методом проб и ошибок. Поддержка между i-м правилом и консеквентом c_i есть отношение между суммой степеней принадлежности всех экземпляров с классом c_i к этому правилу и количеством экземпляров в наборе данных.

E-алгоритм является модификацией алгоритма Ishibuchi, способным работать с данными в их исходном виде даже при наличии дисбаланса [8]. В этом алгоритме меры доверия и поддержки определяются через взвешенные суммы степеней принадлежности. Как и в предыдущем алгоритме, здесь рекомендуется использовать по 30 правил для каждого класса [8].

В [9] описан иерархический алгоритм построения системы классификации на основе нечётких правил HFRBCS (Hierarchical fuzzy rule based classification system), насчитывающий три этапа. На первом этапе происходит генерация первичной структуры, а также оценка правил на эффективность. На втором этапе каждое правило с низкой эффективностью расширяется: термы правила разбиваются на более мелкие области, после чего на их основе формируется новое правило. Третий этап заключается в отборе и удалении избыточных и ошибочных правил генетическим алгоритмом.

Поскольку не все алгоритмы генерации структуры способны учитывать дисбаланс классов в исследуемых данных, разработчиками используются дополнительные инстру-

менты для преодоления этой проблемы. Например, изменяется расчёт весов правил. Для работы с несбалансированными данными в [10] было предложено рассчитывать вес правила с учётом затрат за ошибочную классификацию. Чтобы сделать упор на класс меньшинства, штраф за неправильную классификацию этого класса должен быть намного выше, чем стоимость неправильной классификации класса большинства.

Другим способом уменьшения влияния дисбаланса является использование подходов к предобработке данных, позволяющих или уменьшить количество экземпляров отрицательного класса (under-sampling methods), или увеличить число положительных экземпляров (over-sampling methods). Среди under-sampling methods часто используется Random under-sampling — неэвристический алгоритм, который направлен на устранение дисбаланса по классам путём случайного исключения экземпляров класса большинства. Недостатком этого алгоритма является возможность потери информации о данных класса большинства [11–13].

Среди методов группы over-sampling наиболее известным является алгоритм SMOTE (Synthetic Minority Over-sampling Technique), а также его многочисленные модификации [3, 11, 14, 15]. В исходной версии алгоритма генерация новых синтетических экземпляров положительного класса происходит на основе соседствующих объектов этого класса. Применение предобработки данных позволяет упростить задачу классификации тем алгоритмам, которые не способны напрямую обращать внимание на наличие дисбаланса в данных. Исследование [16] показало, что для алгоритмов генерации структуры Chi и HFRBCS алгоритм предобработки данных помогает существенно улучшить качество классификации.

Однако у алгоритма SMOTE есть недостатки. Добавление дополнительных экземпляров ведёт к естественному увеличению временны́х затрат на обучение и может стать причиной воспроизведения ошибочной информации, если она присутствует в данных положительного класса [17]. Кроме того, алгоритм малоэффективен при многоклассовой задаче классификации.

В данной работе предложен способ построения нечёткого классификатора, для которого не требуется обязательной предобработки данных.

Нечёткий классификатор. Нечёткий классификатор задаётся правилами вида «ЕСЛИ, ТО». В антецедентной части правила с помощью лингвистических термов описывается классифицируемая ситуация. Термы могут задаваться различными функциями принадлежности: треугольными, трапециевидными, функциями Гаусса и др. В консеквентной части правила указывается метка класса. В общем виде нечёткое правило R_i ($i \in [1, r]$, r — число правил) выглядит следующим образом:

$$\text{ЕСЛИ} \quad x_1 = T_{i1} \quad \text{И} \quad x_2 = T_{i2}, \text{И} \ \ldots \ \text{И} \quad x_n = T_{in}, \quad \text{ТО} \quad \text{class} = c_j,$$

где x_k — k-й признак классифицируемого объекта $\mathbf{x}$ ($k \in [1, n]$, n — количество признаков), T_{ik} — нечёткий терм, описывающий k-й признак в i-м правиле, c_j — метка j-го класса ($j \in [1, m]$, m — число классов) [18].

Для определения выхода классификатора необходимо оценить степень принадлежности объекта $\mathbf{x}$ каждому правилу:

$$\beta_i = \prod_{k=1}^{n} T_{ik}(x_k).$$

Согласно стратегии «победитель получает всё» выходом будет считаться метка класса правила с наибольшей степенью принадлежности:

$$\text{class} = c_{j*}, \qquad j^* = \arg \max_{1 \leqslant i \leqslant r} \beta_i.$$

На основе выхода определяется качество построенного классификатора с помощью заданной метрики. Стандартной метрикой качества классификации является общая точность

$$\text{acc} = \sum_{j=1}^{m} \text{inst}_j^* \Big/ \sum_{j=1}^{m} \text{inst}_j,$$

где inst_j^* — количество правильно определённых экземпляров j-го класса, inst_j — количество всех экземпляров j-го класса. Общая точность не будет адекватной оценкой при наличии дисбаланса: в случае если положительный класс не будет распознаваться полностью классификатором, процент точности может быть больши́м благодаря высокому качеству распознавания отрицательного класса [12]. Для преодоления данной проблемы исследователями разработан ряд альтернативных метрик. Одной из наиболее распространённых оценок является средняя геометрическая точность, достоинства которой заключаются в простом расчёте и отсутствии каких-либо параметров. Среднее геометрическое рассчитывается следующим образом:

$$\text{GM} = \Big(\prod_{j=1}^{m} \frac{\text{inst}_j^*}{\text{inst}_j} \Big)^{1/m}.$$

Таким образом, чем меньшим количеством экземпляров представлен класс, тем существеннее будет увеличиваться среднее геометрическое при увеличении числа правильно классифицированных экземпляров данного класса. В случае, когда один из классов полностью будет классифицирован неправильно, среднее геометрическое будет равно нулю.

В [5] нами предложено использовать совокупность двух этих метрик:

$$\text{score} = \gamma \text{GM} + (1 - \gamma)\text{acc},$$

где γ — коэффициент приоритета. Такой подход позволяет получить более сбалансированную точность, чем их применение по отдельности.

Формирование структуры классификатора. Процесс формирования структуры — первый и ключевой этап в построении классификатора. База правил должна не только описывать предметную область в достаточной мере для обеспечения высокого качества классификации, но и обладать невысокой вычислительной сложностью. Кроме того, база правил должна поддаваться интерпретации, что может быть достигнуто только при небольшом количестве правил.

Алгоритм экстремумов классов (ЭК) позволяет создать структуру с минимально возможным количеством правил, равным количеству классов [4]. Классификатор с минимальным количеством правил будет обладать низкой вычислительной сложностью, что ускорит процесс его обучения. Однако при наличии существенного дисбаланса классов или сложного распределения данных минимального количества правил может быть недостаточно. Поэтому мы предлагаем дополнить алгоритм экстремальных значений признаков классов механизмом итерационного добавления правил, который даст возможность улучшить качество распознавания наименьших классов.

Алгоритм экстремумов классов. Алгоритм экстремальных значений признаков классов был подробно описан в [4]. Результатом применения этого алгоритма является база правил, где каждый класс описывается одним правилом. Каждое правило содержит по одному терму для каждого признака; терм равномерно распределён между минимальным и максимальным значениями признака согласно обучающей таблице наблюдений.

Добавление правил алгоритмом «прыгающих лягушек». Предлагается применять алгоритм «прыгающих лягушек» (АПЛ) [19] для итеративного добавления одного правила ко множеству правил. После генерации первичной структуры классификатора и после добавления каждого нового правила оценивается точность каждого класса и выбирается класс с наименьшей точностью. Далее для этого класса генерируется случайный набор термов — по одному для каждого признака. При использовании термов гауссова типа параметры a и b определяются следующим образом:

$$a_k = lb_k + (rb_k - lb_k)\mathrm{rand}_1, \qquad b_k = (rb_k - lb_k)\mathrm{rand}_2/2,$$

где a_k и b_k — координаты вершины терма для k-го признака по оси абсцисс и его разброс соответственно, lb_k и rb_k — левая и правая границы области определения k-го признака, rand_1 и rand_2 — случайные числа из интервала $[0; 1]$. Задача метаэвристики — настроить параметры созданных термов так, чтобы качество классификации улучшилось.

На вход АПЛ подаются исходная база правил base, созданная первичным алгоритмом, и популяция из N входных векторов, каждый из которых представляет собой новое правило и включает параметры термов и консеквент в виде метки класса. Также задаются параметры метаэвристики: G — количество глобальных итераций, T — количество локальных итераций, const — константа для генерации новых параметров термов, N_{mem} — количество мемплексов, N_{agents} — количество векторов в мемплексе, $N = N_{\mathrm{mem}} \cdot N_{\mathrm{agents}}$.

Алгоритм состоит из последовательности действий.

Первый шаг представляет собой глобальный поиск, в котором популяция сортируется по убыванию значения фитнес-функции, после чего счётчик глобальных итераций увеличивается на единицу. Здесь используется фитнес-функция, которая отражает улучшение качества классификации при добавлении нового правила R^* к исходной базе правил по сравнению с качеством классификации, полученным только на исходной базе:

$$\mathrm{fit}\,(\mathrm{base} \cup R^*) = \mathrm{score}\,(\mathrm{base} \cup R^*) - \mathrm{score}\,(\mathrm{base}).$$

На втором шаге проводится локальный поиск, который итерационно повторяет заданное количество локальных итераций. В каждом мемплексе выбираются векторы **best** и **worst** с лучшей и худшей фитнес-функцией соответственно. Так как в данной работе не используется физическое разделение популяции на группы, то принадлежность вектора к мемплексу определяется по индексу. В вектор **best** записывается вектор с индексом n_{mem}, равным номеру текущего мемплекса ($n_{\mathrm{mem}} \in [0, N_{\mathrm{mem}})$). Вектор **worst** определяется по индексу w, который вычисляется по формуле

$$w = N - fN_{\mathrm{mem}} + n_{\mathrm{mem}},$$

где f — счётчик замены. Счётчик нужен для того, чтобы не заменять на протяжении всего локального поиска один и тот же вектор.

На основе выбранных векторов генерируется новое правило R^*, определяемое вектором

$$\mathbf{new} = \mathrm{randconst}(\mathbf{best} - \mathbf{worst}) + \mathbf{worst},$$

где rand — случайное число от 0 до 1. Если фитнес-функция созданного правила оказывается лучше, чем у **worst**, то **worst** заменяется **new**, счётчик замены f увеличивается на 1. В противном случае генерация происходит повторно, но на этот раз в **best** записывается глобально лучший вектор (первый в упорядоченной популяции). Если и в этом случае не удалось улучшить вектор **worst**, то на месте **worst** происходит создание нового случайного вектора, но счётчик замены при этом не изменяется.

Таблица 1

Описание наборов данных для тестирования нечёткого классификатора

№	Наборы данных	Аббревиатура	Признаки	inst_{all}	inst_{+}	inst_{-}	IR
1	glass1	gl1	9	214	76	138	1,82
2	ecoli0vs1	ecl0/1	7	220	77	143	1,86
3	wisconsin	wis	9	683	239	444	1,86
4	pima	pm	8	768	268	500	1,87
5	glass0	gl0	9	214	70	144	2,06
6	yeast1	yst1	8	1484	429	1055	2,46
7	vehicle1	vhc1	18	846	217	629	2,90
8	vehicle2	vhc2	18	846	218	628	2,88
9	vehicle3	vhc3	18	846	212	634	2,99
10	haberman	hbr	3	306	81	225	2,78
11	glass0123vs456	gl0123/456	9	214	51	163	3,20
12	vehicle0	vhc0	18	846	199	647	3,25
13	ecoli1	ecl1	7	336	77	259	3,36
14	newthyroid2	nwth2	5	215	35	180	5,14
15	newthyroid1	nwth1	5	215	35	180	5,14
16	ecoli2	ecl2	7	336	52	284	5,46
17	segment0	sgm0	19	2308	329	1979	6,02
18	glass6	gl6	9	214	29	185	6,38
19	yeast3	yst3	8	1484	163	1321	8,10
20	ecoli3	ecl3	7	336	35	301	8,60
21	page-blocks0	pb0	10	5472	559	4913	8,79
22	yeast2vs4	yst2/4	8	514	51	463	9,08
23	yeast05679vs4	yst05679/4	8	528	51	477	9,35
24	vowel0	vwl0	13	988	90	898	9,98
25	glass2	gl2	9	214	17	197	11,59
26	ecoli4	ecl4	7	336	20	316	15,80
27	glass4	gl4	9	214	13	201	15,46
28	page-blocks13vs2	pb13/2	10	472	28	444	15,86
29	abalone9-18	ab9/18	7/8	731	42	689	16,40
30	yeast1458vs7	yst1458/7	8	693	30	663	22,10
31	yeast2vs8	yst2/8	8	482	20	462	23,10
32	yeast4	yst4	8	1484	51	1433	28,10
33	yeast1289vs7	yst1289/7	8	947	30	917	30,57
34	yeast5	yst5	8	1484	44	1440	32,73
35	ecoli0137vs26	ecl0137/26	7	281	7	274	39,14
36	yeast6	yst6	8	1484	35	1449	41,40

Когда локальные итерации истекают, алгоритм возвращается на второй шаг. После достижения счётчиком глобальных итераций значения G алгоритм выдаёт вектор правила с наибольшей фитнес-функцией.

Описание эксперимента. Для эксперимента были использованы 36 несбалансированных наборов данных из открытого репозитория KEEL (Knowledge Extraction based on Evolutionary Learning) [20]. Описание наборов для первого этапа эксперимента приведено в табл. 1. Здесь inst_{all} — общее количество экземпляров в наборе, inst_{+} — количество экземпляров положительного класса, inst_{-} — число экземпляров отрицательного класса, IR — коэффициент дисбаланса, $IR = \text{inst}_{-}/\text{inst}_{+}$. Все наборы содержат только два класса.

В наборе данных abalone9-18 удалён признак «Пол», являющийся номинальным.

Эксперимент проводился по схеме пятикратной кроссвалидации. На всех выборках комбинация алгоритма экстремальных значений классов и АПЛ запускалась по пять раз

для каждого числа добавляемых правил. Были замерены результаты построения нечётких классификаторов при добавлении одного, двух, пяти и семи правил. В качестве функции принадлежности использовались функции Гаусса. Параметры АПЛ были следующими: 15 глобальных итераций, 20 локальных итераций, 5 мемплексов, 5 векторов в мемплексе; константа для генерации новых параметров термов равнялась 1,2. Все параметры были подобраны эмпирически как наиболее универсальные для рассматриваемых наборов данных. Коэффициент приоритета в фитнес-функции равнялся 0,5.

Результаты эксперимента. В табл. 2 представлены средняя геометрическая точность построенных классификаторов и её разброс на основе результатов пяти запусков. Столбец ЭК демонстрирует качество классификации до добавления правил, полученное алгоритмом ЭК. В следующих столбцах приведены результаты после добавления одного, двух, пяти и семи правил (переменная $\#R$ показывает итоговое число правил) комбинацией алгоритма экстремумов классов и АПЛ (ЭК+АПЛ). Последняя строка демонстрирует усреднённые значения по всем наборам данных.

Только на двух наборах данных (newthyroid1 и newthyroid2) нечёткий классификатор демонстрировал лучшее значение средней геометрической точности до добавления новых правил. В остальных 34 случаях точность повышалась после расширения базы правил.

Сравнение результатов экспериментов. Эксперимент продемонстрировал увеличение значения средней геометрической точности классификатора после добавления правил метаэвристическим алгоритмом «прыгающих лягушек». Это подтверждает статистическое сравнение с помощью непараметрического критерия Манна — Уитни — Уилкоксона (табл. 3). Нулевая гипотеза гласит, что между результатами нет статистических различий, уровень значимости равен 0,05. Значение стандартизированной статистики критерия (ССК) показывает степень превосходства результатов, полученных после добавления правил, над точностью классификатора на исходной базе правил.

Во всех случаях нулевая гипотеза отклоняется, а ССК является положительной. Следовательно, классификаторы с дополненными базами правил показали более высокие результаты, чем классификаторы, структура которых построена только алгоритмом экстремальных значений классов.

В табл. 4 приведено сравнение полученных результатов с аналогичными алгоритмами формирования нечётких классификаторов и алгоритмом построения решающих деревьев C4.5 из [9]. Все алгоритмы, кроме E-алгоритма, используют предварительно процедуру предобработки данных с помощью алгоритма SMOTE, поэтому находятся в более привилегированном положении, чем E-алгоритм или предлагаемая нами комбинация ЭК + АПЛ.

При сравнении использованы лучшие усреднённые результаты нечётких классификаторов (представлены в последнем столбце табл. 4). В скобках указано, на каком количестве правил получено это значение. Для экономии места наборы данных были обозначены порядковым номером согласно табл. 1.

Для попарного сравнения результатов был использован критерий Манна — Уитни — Уилкоксона. Полученные значения критерия показаны в табл. 5.

По значению среднего геометрического результаты нечётких классификаторов, построенных с помощью алгоритма экстремумов классов и АПЛ, статистически неразличимы с результатами таких алгоритмов, как Chi-3 и Chi-5 (преимущество в пользу представленного алгоритма), а также HFRBCS и C4.5 (преимущество в пользу аналогов). Сравнение показывает, что применение предлагаемой комбинации алгоритмов существенно превосходит результаты Ishibuchi и E-алгоритма, последний из которых позиционируется как алгоритм построения структуры при наличии дисбаланса в данных.

Нужно отметить, что разброс результатов, полученных с помощью комбинации ЭК+АПЛ меньше, чем у аналогов. Это подтверждается статистическим сравнением (табл. 6).

Таблица 2

Результаты построения структуры классификатора при добавлении правил алгоритмом «прыгающих лягушек»

Алгоритм	ЭК	ЭК + АПЛ			
$\#R$	2	3	4	7	9
gl1	40,53	60,92 ± 3,82	62,75 ± 5,98	68,89 ± 5,08	68,50 ± 5,19
ecl0/1	88,78	95,05 ± 1,80	96,44 ± 1,22	95,84 ± 1,90	95,36 ± 2,41
wis	73,39	93,47 ± 1,04	95,09 ± 1,10	94,95 ± 1,62	94,53 ± 1,01
pm	55,62	68,08 ± 1,90	70,57 ± 1,55	71,63 ± 1,77	72,16 ± 2,32
gl0	60,07	74,41 ± 3,55	73,91 ± 3,41	75,94 ± 4,43	78,42 ± 3,78
yst1	39,61	65,24 ± 2,61	66,18 ± 1,34	69,01 ± 1,24	69,85 ± 1,25
vhc1	41,93	50,10 ± 2,70	61,75 ± 3,37	66,14 ± 2,86	68,37 ± 1,88
vhc2	39,95	59,21 ± 2,68	67,07 ± 4,28	80,21 ± 2,97	82,78 ± 3,98
vhc3	39,12	43,28 ± 2,43	52,86 ± 3,06	64,61 ± 2,35	66,39 ± 2,60
hbr	44,28	46,05 ± 3,43	50,54 ± 3,81	54,98 ± 4,38	56,45 ± 4,76
gl0123/456	87,64	93,17 ± 1,49	93,07 ± 2,12	90,76 ± 3,07	89,66 ± 2,67
vhc0	55,50	71,81 ± 4,19	75,50 ± 3,34	81,63 ± 1,84	86,36 ± 2,18
ecl1	80,77	86,77 ± 1,75	88,43 ± 1,57	88,09 ± 1,96	88,34 ± 2,20
nwth2	99,16	97,94 ± 0,47	98,18 ± 0,09	96,77 ± 2,54	96,83 ± 2,08
nwth1	99,16	97,16 ± 1,57	96,08 ± 1,99	94,94 ± 3,55	94,05 ± 3,85
ecl2	34,24	81,23 ± 4,61	84,89 ± 4,07	89,31 ± 2,65	90,54 ± 2,77
sgm0	88,06	92,65 ± 1,09	94,02 ± 0,98	88,95 ± 1,04	97,58 ± 0,82
gl6	22,77	83,72 ± 5,07	86,45 ± 4,87	86,53 ± 4,42	88,25 ± 4,66
yst3	85,48	88,47 ± 1,41	90,08 ± 0,81	90,35 ± 1,76	89,96 ± 1,25
ecl3	50,75	84,03 ± 3,05	87,08 ± 2,95	85,52 ± 3,30	81,72 ± 4,66
pb0	63,62	74,13 ± 1,78	81,85 ± 1,57	79,84 ± 2,26	88,21 ± 0,85
yst2/4	67,31	82,87 ± 3,93	83,99 ± 3,22	86,72 ± 4,88	85,48 ± 3,82
yst05679/4	61,91	69,23 ± 3,20	74,22 ± 3,56	74,35 ± 3,63	73,71 ± 3,94
vwl0	83,87	84,38 ± 1,02	86,62 ± 1,23	88,91 ± 3,10	92,11 ± 3,37
gl2	10,78	57,80 ± 9,98	61,71 ± 10,63	61,11 ± 8,73	53,37 ± 14,21
ecl4	68,70	87,98 ± 4,74	89,44 ± 3,10	89,92 ± 3,11	84,32 ± 2,99
gl4	23,09	76,13 ± 8,75	74,82 ± 14,35	79,43 ± 10,45	75,86 ± 12,49
pb13/2	75,12	85,72 ± 3,69	89,22 ± 3,93	90,07 ± 5,12	86,76 ± 5,14
ab9/18	58,73	67,44 ± 3,25	71,39 ± 3,72	75,02 ± 2,30	72,24 ± 7,25
yst1458/7	45,81	59,59 ± 3,24	56,60 ± 6,24	54,67 ± 6,71	52,74 ± 9,89
yst2/8	68,82	68,45 ± 3,25	69,66 ± 3,96	70,21 ± 4,51	66,17 ± 8,01
yst4	65,74	70,16 ± 2,50	75,65 ± 2,74	79,87 ± 2,13	77,26 ± 3,91
yst1289/7	54,07	62,35 ± 3,13	63,07 ± 4,28	61,30 ± 6,04	63,25 ± 7,21
yst5	72,94	92,48 ± 1,59	91,75 ± 1,98	93,46 ± 2,06	92,54 ± 2,89
ecl0137/26	0,00	71,34 ± 3,80	68,47 ± 8,76	61,18 ± 16,33	65,83 ± 9,45
yst6	51,15	82,34 ± 3,46	83,97 ± 2,53	83,41 ± 2,06	83,01 ± 3,97
Среднее	**58,29**	**75,70 ± 3,11**	**78,15 ± 3,55**	**79,57 ± 3,84**	**79,69 ± 4,33**

Таблица 3

Статистическое сравнение средней геометрической точности классификаторов до и после дополнения базы правил

Количество добавленных правил	p-value	CCK	Нулевая гипотеза
1	$< 0{,}001$	5,090	Отклоняется
2	$< 0{,}001$	5,137	Отклоняется
5	$< 0{,}001$	5,106	Отклоняется
7	$< 0{,}001$	5,075	Отклоняется

Таблица 4

Сопоставление средней геометрической точности с аналогичными алгоритмами построения нечётких классификаторов и с алгоритмом C4.5

Данные	Chi-3	Chi-5	Ishibuchi	E-алгоритм	HFRBCS	C4.5	ЭК + АПЛ
1	64,9 ± 6,9	64,9 ± 6,9	59,3 ± 10,3	0,0 ± 0,0	73,7 ± 4,7	75,1 ± 3,7	68,9 ± 5,1 (7)
2	92,3 ± 5,9	95,6 ± 5,2	96,7 ± 2,4	95,3 ± 4,8	93,6 ± 6,5	98,0 ± 2,2	96,4 ± 1,2 (4)
3	88,9 ± 2,1	43,6 ± 5,9	95,8 ± 1,4	96,0 ± 1,6	88,2 ± 1,6	95,4 ± 2,0	95,1 ± 1,1 (4)
4	66,8 ± 5,9	66,8 ± 2,3	71,1 ± 4,5	55,0 ± 4,6	68,7 ± 5,3	71,3 ± 4,1	72,2 ± 2,3 (9)
5	64,1 ± 3,5	63,7 ± 1,8	69,4 ± 7,7	0,0 ± 0,0	76,6 ± 8,1	78,1 ± 2,2	78,4 ± 3,8 (9)
6	67,7 ± 1,9	69,7 ± 1,5	51,4 ± 12,2	0,0 ± 0,0	71,7 ± 2,4	70,9 ± 3,0	69,9 ± 1,2 (9)
7	70,9 ± 4,3	71,9 ± 1,3	64,9 ± 4,4	3,1 ± 6,9	71,8 ± 2,6	69,3 ± 3,4	68,4 ± 1,9 (9)
8	85,5 ± 3,4	87,2 ± 3,0	67,8 ± 5,0	43,8 ± 13,2	90,6 ± 2,2	94,9 ± 1,7	82,8 ± 4,0 (9)
9	69,2 ± 4,9	63,1 ± 2,0	63,1 ± 4,1	0,0 ± 0,0	66,8 ± 3,3	74,3 ± 1,1	66,4 ± 2,6 (9)
10	58,9 ± 6,0	60,4 ± 2,4	62,7 ± 2,8	4,9 ± 11,1	57,1 ± 4,1	61,3 ± 3,9	56,4 ± 4,8 (9)
11	85,8 ± 3,0	85,9 ± 1,7	88,6 ± 5,2	82,1 ± 7,0	88,4 ± 4,0	90,1 ± 3,2	93,2 ± 1,5 (3)
12	86,4 ± 3,1	84,9 ± 1,6	75,9 ± 1,4	39,1 ± 16,5	88,9 ± 2,0	91,1 ± 2,7	86,4 ± 2,2 (9)
13	85,3 ± 9,8	86,1 ± 8,6	85,7 ± 2,9	77,8 ± 7,9	84,2 ± 12,7	76,1 ± 9,6	88,4 ± 1,6 (4)
14	89,8 ± 10,8	96,3 ± 6,7	94,2 ± 4,2	88,6 ± 3,8	99,7 ± 0,6	96,5 ± 4,9	98,2 ± 0,1 (4)
15	87,4 ± 8,1	95,4 ± 8,8	89,0 ± 13,5	88,5 ± 8,8	98,6 ± 2,5	98,0 ± 3,8	97,2 ± 1,6 (3)
16	88,0 ± 5,5	87,6 ± 5,0	87,0 ± 4,4	70,4 ± 15,4	87,6 ± 8,2	91,6 ± 4,9	90,5 ± 2,8 (9)
17	95,0 ± 0,5	95,9 ± 1,2	42,5 ± 2,8	95,3 ± 1,1	97,5 ± 1,1	99,3 ± 0,6	97,6 ± 0,8 (9)
18	83,9 ± 9,8	78,1 ± 7,8	86,3 ± 8,2	90,2 ± 3,8	87,0 ± 10,8	83,0 ± 9,1	88,2 ± 4,7 (9)
19	90,1 ± 4,1	89,3 ± 3,3	77,1 ± 17,7	82,0 ± 2,3	90,4 ± 2,3	88,5 ± 3,7	90,4 ± 1,8 (7)
20	87,6 ± 4,1	91,6 ± 5,0	85,4 ± 3,7	75,5 ± 8,7	90,8 ± 4,4	88,8 ± 7,7	87,1 ± 3,0 (4)
21	79,9 ± 4,3	87,3 ± 1,9	32,2 ± 9,6	64,5 ± 2,8	91,4 ± 0,7	94,8 ± 1,5	88,2 ± 0,9 (9)
22	86,8 ± 5,5	86,4 ± 7,4	70,9 ± 23,5	80,9 ± 9,1	89,3 ± 4,2	85,1 ± 10,2	86,7 ± 4,9 (7)
23	78,9 ± 6,0	76,0 ± 6,4	79,5 ± 9,5	60,0 ± 16,4	73,2 ± 7,5	74,9 ± 10,9	74,3 ± 3,6 (7)
24	98,4 ± 0,6	97,9 ± 1,8	89,0 ± 6,6	89,6 ± 6,1	98,8 ± 1,6	94,7 ± 5,2	92,1 ± 3,4 (9)
25	47,7 ± 10,2	49,2 ± 8,2	43,6 ± 15,7	9,9 ± 22,1	54,8 ± 20,6	33,9 ± 32,3	61,7 ± 10,6 (4)
26	91,3 ± 7,4	92,1 ± 8,4	86,9 ± 8,7	92,4 ± 8,2	93,0 ± 8,2	81,3 ± 11,7	89,9 ± 3,1 (7)
27	85,0 ± 13,8	81,8 ± 11,2	78,3 ± 17,7	83,4 ± 19,9	70,4 ± 40,5	83,7 ± 10,8	79,4 ± 10,4 (7)
28	91,9 ± 4,8	92,9 ± 9,5	94,5 ± 4,9	94,1 ± 10,3	98,6 ± 0,7	99,6 ± 0,5	90,1 ± 5,1 (7)
29	63,9 ± 11,0	66,5 ± 10,7	65,8 ± 9,2	32,3 ± 20,6	67,6 ± 14,0	53,2 ± 8,3	75,0 ± 2,3 (7)
30	62,4 ± 4,6	58,8 ± 8,6	40,8 ± 16,6	0,0 ± 0,0	62,5 ± 6,3	41,2 ± 6,1	59,6 ± 3,2 (3)
31	72,8 ± 15,0	78,8 ± 8,6	72,8 ± 15,0	72,8 ± 15,0	72,5 ± 15,1	78,2 ± 13,1	70,2 ± 4,5 (7)
32	83,0 ± 3,1	83,1 ± 2,6	71,4 ± 23,3	32,2 ± 20,6	82,6 ± 2,3	65,0 ± 8,9	79,9 ± 2,1 (7)
33	76,1 ± 7,2	69,3 ± 4,6	48,6 ± 16,9	50,0 ± 13,6	69,4 ± 4,4	64,1 ± 9,0	63,3 ± 7,2 (7)
34	93,4 ± 5,4	93,6 ± 2,1	94,9 ± 0,4	88,2 ± 7,0	94,2 ± 2,6	92,0 ± 5,0	93,5 ± 2,1 (7)
35	71,0 ± 41,4	49,6 ± 46,4	71,3 ± 41,7	73,7 ± 43,1	71,5 ± 41,8	71,2 ± 41,3	71,3 ± 3,8 (3)
36	87,5 ± 10,6	87,7 ± 9,3	88,4 ± 6,1	51,7 ± 13,8	84,9 ± 12,9	80,4 ± 15,5	84,0 ± 2,5 (4)
Ср.	**80,0 ± 7,1**	**78,6 ± 6,4**	**73,4 ± 9,6**	**57,3 ± 9,6**	**81,8 ± 7,6**	**80,1 ± 7,4**	**81,4 ± 3,3 (6,7)**

Таблица 5

Сравнение средней геометрической точности, полученной различными классификаторами, непараметрическим критерием Манна — Уитни — Уилкоксона

Сравниваемые алгоритмы	p-value	ССК	Нулевая гипотеза
Chi-3 — ЭК + АПЛ	0,242	–1,17	Принимается
Chi-5 — ЭК + АПЛ	0,315	–1,005	Принимается
Ishibuchi — ЭК + АПЛ	< 0,001	–3,723	Отклоняется
E-алгоритм — ЭК + АПЛ	< 0,001	–4,556	Отклоняется
HFRBCS — ЭК + АПЛ	0,396	0,848	Принимается
C4.5 — ЭК + АПЛ	0,925	0,094	Принимается

Таблица 6

Сравнение разброса критерием Манна — Уитни — Уилкоксона

Сравниваемые алгоритмы	p-value	ССК	Нулевая гипотеза
Chi-3 — ЭК + АПЛ	<0,001	4,572	Отклоняется
Chi-5 — ЭК + АПЛ	0,001	3,224	Отклоняется
Ishibuchi — ЭК + АПЛ	<0,001	4,823	Отклоняется
E-алгоритм — ЭК + АПЛ	<0,001	4,336	Отклоняется
HFRBCS — ЭК + АПЛ	<0,001	3,488	Отклоняется
C4.5 — ЭК + АПЛ	<0,001	3,998	Отклоняется

Заключение. В данной работе был предложен новый способ формирования структуры нечёткого классификатора, заключающийся в комбинации алгоритма экстремальных значений признаков классов и алгоритма «прыгающих лягушек». Благодаря использованию фитнес-функции, включающей в себя как общую точность, так и среднее геометрическое, при обучении классификатора учитывается несбалансированный характер данных.

Продемонстрированные результаты на наборах данных с двумя классами показывают конкурентоспособность предложенного алгоритма по сравнению с аналогичными алгоритмами формирования структуры нечётких классификаторов. При меньшем количестве правил созданные нами нечёткие классификаторы показывают превосходящее значение средней геометрической точности относительно алгоритма без предобработки данных и статистически неразличимые результаты с алгоритмами, которые были построены на предварительно дополненных данных.

В дальнейших исследованиях планируется изучить различные варианты создания термов при генерации популяции для метаэвристики, рассмотреть эффективность настройки всех термов в процессе добавления правил, проверить целесообразность дополнения правил весами, а также попробовать другие метаэвристики в данной задаче.

Финансирование. Работа выполнена при поддержке Российского фонда фундаментальных исследований (проект № 19-37-90064).

СПИСОК ЛИТЕРАТУРЫ

1. **Gil M. A., González-Rodríguez G., Kruse R.** Editorial of the special issue “Statistics with Imperfect Data” // Inf. Sci. 2013. **245**. P. 1–3.
2. **Peng L., Zhang H., Yang B., Chen Y.** A new approach for imbalanced data classification based on data gravitation // Inf. Sci. 2014. **288**. P. 347–373.

3. **Mathew J., Pang C. K., Luo M., Leong W. H.** Classification of imbalanced data by oversampling in Kernel space of support vector machines // IEEE Trans. Neural Netw. Learn. Syst. 2018. **29**. P. 4065–4076.
4. **Bardamova M., Konev A., Hodashinsky I., Shelupanov A.** A fuzzy classifier with feature selection based on the gravitational search algorithm // Symmetry. 2018. **10**. P. 609.
5. **Bardamova M., Konev A., Hodashinsky I., Shelupanov A.** Application of the gravitational search algorithm for constructing fuzzy classifiers of imbalanced data // Symmetry. 2019. **11**. P. 1458.
6. **Chi Z., Yan H., Pham T.** Fuzzy algorithms with applications to image processing and pattern recognition // Advances in Fuzzy Systems. Applications and Theory. Vol 10. Singapore: World Scientific Pub Co Inc, 1996. 232 p.
7. **Ishibuchi H., Yamamoto T.** Rule weight specification in fuzzy rule-based classification systems // IEEE Trans. Fuzzy Syst. 2005. **13**, N 4. P. 428–435.
8. **Xu L., Chow M. Y., Taylor L. S.** Power distribution fault cause identification with imbalanced data using the data mining-based fuzzy classification ealgorithm // IEEE Trans. Power Syst. 2007. **22**, N 1. P. 164–171.
9. **Fernández A., García S., del Jesus M. J., Herrera F.** A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced data-sets // Fuzzy Set. Syst. 2008. **159**, N 18. P. 2378–2398.
10. **López V., del Río S., Benítez J. M., Herrera F.** Cost-sensitive linguistic fuzzy rule based classification systems under the MapReduce framework for imbalanced big data // Fuzzy Set. Syst. 2015. **258**. P. 5–38.
11. **Haixiang G., Yijing L., Shang J. et al.** Learning from class-imbalanced data: Review of methods and application // Expert Syst. Appl. 2017. **73**. P. 220–239.
12. **Imbalanced Learning**: Foundations, Algorithms, and Applications. New Jersey: John Wiley & Sons, Inc., 2013. 216 p.
13. **D'Addabbo A., Maglietta R.** Parallel selective sampling method for imbalanced and large data classification // Patt. Recogn. Lett. 2015. **62**. P. 61–67.
14. **Chawla N. V., Bowyer K. W., Hall L. O., Kegelmeyer W. P.** SMOTE: Synthetic minority over-sampling technique // Journ. Artif. Intell. Res. 2002. **16**. P. 321–357.
15. **Liu G., Yang Y., Li B.** Fuzzy rule-based oversampling technique for imbalanced and incomplete data learning // Knowledge-Based Syst. 2018. **158**. P. 154–174.
16. **Fernández A., del Jesus M. J., Herrera F.** Hierarchical fuzzy rule based classification systems with genetic rule selection for imbalanced data-sets // Int. Journ. Approx. Reasoning. 2009. **50**, N 3. P. 561–577.
17. **Nguyen G. H., Bouzerdoum A., Phung S. L.** Learning Pattern Classification Tasks with Imbalanced Data Sets. Pattern Recognition. UK: IntechOpen, 2009. P. 193–208.
18. **Ходашинский И. А., Минина Д. Ю., Сарин К. С.** Идентификация параметров нечетких аппроксиматоров и классификаторов на основе алгоритма «кукушкин поиск» // Автометрия. 2015. **51**, № 3. С. 27–34.
19. **Elbeltagi E., Hegazy T., Grierson D.** A modified shuffled frog-leaping optimization algorithm: Applications to project management // Struct. Infrastruct. Eng. 2007. **3**, N 1. P. 53–60.
20. **Knowledge** Extraction Based on Evolutionary Learning. URL: http://www.keel.es/ (дата обращения: 11.03.2020).

Поступила в редакцию 15.12.2020
После доработки 15.04.2021
Принята к публикации 15.04.2021