

© 2019 г. **И.А. ХОДАШИНСКИЙ**, д-р техн. наук (hodashn@rambler.ru),
К.С. САРИН, канд. техн. наук (sks@security.tomsk.ru)
(Томский университет систем управления и радиоэлектроники)

ОТБОР КЛАССИФИЦИРУЮЩИХ ПРИЗНАКОВ С ПОМОЩЬЮ ПОПУЛЯЦИОННОГО СЛУЧАЙНОГО ПОИСКА С ПАМЯТЬЮ¹

Предложен новый метод отбора признаков. Концепция, лежащая в основе предлагаемого метода, — сочетание стратегий случайного и эвристического поиска. Решение представляется в виде бинарного вектора, размерность которого определена количеством признаков в наборе данных. Генерация новых решений проводится случайным образом с использованием нормального и равномерного распределения. Эвристика, лежащая в основе предлагаемого подхода, формулируется следующим образом: шанс признака попасть в следующую генерацию пропорционален частоте присутствия этого признака в предыдущих лучших решениях. Предложенный метод протестирован на нескольких наборах данных из репозитория KEEL. Проведено сравнение с алгоритмами-аналогами.

Ключевые слова: отбор признаков, нечеткий классификатор, бинарная оптимизация, метаэвристики, случайный поиск с памятью.

DOI: 10.1134/S0005231019020107

1. Введение

Снижение размерности входных данных как этап предварительной обработки встречается во многих областях, таких как распознавание образов, машинное обучение, интеллектуальный анализ данных. Цель этапа — повышение эффективности хранения и обработки данных путем нахождения минимального набора признаков и/или репрезентативного подмножества входных данных меньшего размера. Снижение размерности обусловлено необходимостью убрать те данные исходного набора, которые являются избыточными или выбросами и могут ухудшить эффективность классификации [1, 2]. При классификации больших наборов данных время построения классификаторов значительно возрастает, главным образом, из-за вычисления целевой функции. Ускорение обучения возможно как за счет сокращения размера обучающей выборки, так и за счет сокращения количества признаков (атрибутов, переменных). Однако в данной работе рассматривается только проблема сокращения количества признаков.

Исследования в области отбора признаков начались в 1960-е гг. Автор [3] описал изменение точности байесовского классификатора в зависимости от количества признаков. С тех пор интерес к этому научному направлению не

¹ Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (проект № 16-07-00034).

ослабевает, число книг и статей по данной проблеме постоянно растет [4]. Выделяют два подхода к решению проблемы получения минимального набора признаков: отбор информативных признаков (feature selection) и формирование или выделение множества новых признаков (feature construction, feature extraction). Классическими методами формирования множества новых признаков являются метод главных компонент, линейный дискриминантный и корреляционный анализы. Методы отбора не предполагают создания нового множества признаков, в процессе реализации этого метода сокращается исходное пространство признаков. Применение методов отбора или формирования новых признаков зависит от области приложения и классифицируемого набора данных [5].

Методы отбора признаков можно разделить на две категории: фильтры (filter) и обертки (wrapper) [4, 6]. Фильтры основаны на определенных метриках, таких как энтропия, распределение вероятностей или взаимная информация [7], и не предполагают использования классифицирующего алгоритма в процессе работы. Обертки используют классификатор для оценки подмножества признаков, при этом сам классификатор как бы «завернут» в цикл отбора признаков. Методы фильтров и оберток имеют свои достоинства и недостатки. Достоинствами методов, основанных на фильтрах, являются их масштабируемость и более высокая скорость выполнения. Общий недостаток заключается в том, что отсутствие взаимодействия с классификатором и игнорирование учета зависимости между признаками приводят к различной для разных классификаторов невысокой точности классификации.

Преимущество методов оберток заключается в том, что они работают совместно с конкретным алгоритмом классификации и учитывают синергетический эффект от совместного использования отобранных признаков. Недостатками методов оберток являются более высокий риск переобучения и большие временные затраты по причине необходимости вычисления точности классификации [5]. Метод оберток предполагает использование конкретного классификатора. Выполняется метод оберток в два этапа: 1) поиск подмножества признаков; 2) оценка качества подмножества признаков на основе точности классификатора, построенного на отобранном подмножестве. Указанные этапы повторяются до тех пор, пока не будут выполнены заданные условия останова. Поиск подмножества признаков может быть выполнен полным перебором, вычислительная сложность которого равна $O(2^{|A|})$, где A — исходное множество признаков. Если $|A|$ велико, полный перебор нецелесообразен. Поэтому для поиска подмножества признаков применяют иные стратегии, которые можно разделить на две категории: детерминированные и стохастические (или метаэвристические) методы. Типичным примером детерминированного метода является жадный алгоритм, который итеративно добавляет или удаляет признаки, пока не будет достигнут необходимый эффект. Алгоритм обладает полиномиальной сложностью, прост в реализации и эффективен для наборов данных с небольшим количеством признаков. Однако у данного алгоритма есть существенный недостаток, он не учитывает взаимозависимости между признаками. Стохастические методы способны найти субоптимальные решения как с точки зрения точности, так и размера подмножества признаков, примером является Las Vegas Filter алгоритм, быстро со-

кращающий количество признаков на ранних стадиях и способный находить оптимальные решения, если позволяют вычислительные ресурсы [8]. В реальных приложениях специалисты больше заинтересованы в получении приемлемых решений в разумные сроки, и необязательно эти решения должны быть гарантированно оптимальными. Использование метаэвристических методов позволяет получить такие решения без исследования всего пространства поиска [9]. Для решения проблемы отбора признаков давно и успешно применяются роевые и эволюционные алгоритмы, такие как генетический алгоритм [10, 11], алгоритм роящихся частиц [12, 13], алгоритмы муравьиной [14] и пчелиной колонии [15, 16], гармонический поиск [17, 18], дифференциальная эволюция [19].

Цель работы — описание нового метода обертки, где в качестве классифицирующего алгоритма используется нечеткий классификатор. Достоинство нечетких классификаторов — понятность пользователю полученных правил и их интерпретируемость. 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.

2. Постановка задачи

Пусть имеется универсум $U = (A, G)$, где $A = \{x_1, \dots, x_n\}$ — множество входных признаков, $G = \{1, \dots, m\}$ — множество меток классов. Пусть $\mathbf{X} = x_1 \times \dots \times x_n \in \mathbb{R}^n$ — n мерное пространство признаков. Объект в заданном универсуме характеризуется своим вектором значений признаков.

Задача классификации заключается в предсказании класса объекта по вектору значений признаков. Задача отбора признаков заключается в поиске на заданном множестве признаков \mathbf{X} такого их подмножества, которое при уменьшении числа признаков не приводило бы к существенному уменьшению точности классификации или увеличивало бы ее. Решение представляется в виде вектора $\mathbf{S} = (s_1, \dots, s_n)^T$, где $s_i = 0$ означает, 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. Традиционный классификатор может быть определен как функция

$$f : \mathbb{R}^n \rightarrow \{0, 1\}^m,$$

где $f(\mathbf{x}; \Theta) = (\beta_1, \dots, \beta_m)^T$, причем $\beta_i = 1$, а $\beta_j = 0$, $j = \overline{1, m}$, $i \neq j$, когда объект, заданный вектором \mathbf{x} , принадлежит к классу i ; Θ — вектор параметров классификатора.

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

$$f : \mathbb{R}^n \rightarrow [0, g]^m,$$



Рис. 1. Схема построения классификатора.

где $g \in \mathfrak{R}$ и определяется индивидуальными особенностями классификатора. Основой нечеткого классификатора является продукционное правило следующего вида:

$$R_i : \text{ЕСЛИ } s_1 \wedge x_1 = A_{1i} \text{ И } \dots \text{ И } s_n \wedge x_n = A_{ni} \text{ ТО } y = L_i,$$

где $L_i \in \{1, \dots, m\}$ — выход i -го правила ($i = \overline{1, r}$), r — число правил, A_{ki} — нечеткий терм, характеризующий k -й признак в i -м правиле ($k = \overline{1, n}$); запись $s_k \wedge x_k$ указывает на наличие ($s_k = 1$) или отсутствие ($s_k = 0$) признака в классификаторе.

Выход нечеткого классификатора представляет собой вектор $(\beta_1, \dots, \beta_m)^T$, где

$$\beta_j = \sum_{(i=\overline{1, r}); (L_i=j)} \prod_{k=1}^n \mu_{ki}(x_k),$$

$j = \overline{1, m}$, $\mu_{ki}(x_k)$ — значение функции принадлежности нечеткого термина A_{ki} в точке x_k . На основе полученного вектора класс определяется по принципу «команда победителей получает всё»:

$$\text{class} = \arg \max_{j=\overline{1, m}} \beta_j.$$

На таблице наблюдений $\{(\mathbf{x}_p, c_p), p = \overline{1, z}\}$ мера точности классификации может быть выражена следующим образом:

$$(1) \quad E(\Theta, \mathbf{S}) = \frac{\sum_{p=1}^z \begin{cases} 1, & \text{если } c_p = \arg \max_{j=\overline{1, m}} f_j(\mathbf{x}_p; \Theta, \mathbf{S}), \\ 0 & \text{иначе} \end{cases}}{z},$$

где $f(\mathbf{x}_p; \Theta, \mathbf{S})$ — выход нечеткого классификатора с параметрами Θ и признаками \mathbf{S} в точке \mathbf{x}_p . Проблема построения нечеткого классификатора сводится к поиску максимума указанной функции в пространстве \mathbf{S} и $\Theta = (\theta^1, \dots, \theta^D)$:

$$\begin{cases} E(\Theta, \mathbf{S}) \rightarrow \max, \\ \theta_{\min}^i \leq \theta^i \leq \theta_{\max}^i, \quad i = \overline{1, D}, \\ s_j \in \{0, 1\}, \quad j = \overline{1, n}, \end{cases}$$

где $\theta_{\min}^i, \theta_{\max}^i$ — нижняя и верхняя границы каждого параметра соответственно. Проблема относится к классу NP-трудных, поэтому невозможно гарантировать получение оптимального решения за исключением случаев, когда выполняется полный перебор в пространстве решений. В данной работе предлагается решать указанную проблему с помощью популяционного случайного поиска с памятью. Схема представлена на рис. 1.

3. Алгоритм популяционного случайного поиска с памятью

Алгоритм основан на сочетании стратегий случайного и эвристического поиска. Важнейшим элементом алгоритма является кодирующий признаки вектор-решение \mathbf{S} . Работа алгоритма начинается с формирования популяции — набора сгенерированных случайным или иным образом векторов \mathbf{S} . Число векторов в популяции — это наперед заданное целое число, называемое размером популяции. Для каждого вектора вычисляется мера точности классификации E (формула 1). На каждой итерации определяется вектор с максимальным значением E — лучшее решение на текущей итерации. Другим важным элементом алгоритма является вектор \mathbf{B} , в котором b_i хранит число появлений i -го признака в лучших решениях на предыдущих итерациях, размерность этого вектора совпадает с размерностью \mathbf{S} . Вектор \mathbf{B} служит для реализации следующей эвристики: шанс признака попасть в новую популяцию пропорционален частоте присутствия этого признака в предыдущих лучших решениях. Новая популяция формируется на основе указанной эвристики и случайного поиска. Нормально распределенная случайная величина $u \sim N(0, \sigma_g)$ определяет механизм сокращения или добавления признаков, а также количество сокращенных/добавленных признаков. Если значение u больше или равно нулю, то происходит добавление новых признаков путем увеличения количества единиц в векторе \mathbf{S} , иначе происходит сокращение признаков. Количество потенциально возможных изменяемых признаков определяется размерностью вектора \mathbf{S} , равной n , и числом единиц в векторе \mathbf{S} , положим, равное r . Если значение u меньше нуля, то среди r элементов вектора \mathbf{S} , содержащих единицы, случайным образом выбираются l признаков-кандидатов на сокращение по формуле $l = \text{round}(r|\tanh(u)|)$. Определение числа добавляемых признаков выполняется по формуле $l = \text{round}((n - r)|\tanh(u)|)$. Однако собственное изменение значения элемента s_i определяется значением $b_i \in \mathbf{B}$, а именно, относительной частотой появления i -го признака в лучших решениях, вычисляемой на итерации $t \geq t_g$ как b_i/t , где t_g — итерация, по дости-

жении которой вероятность включения признака определяется относительной частотой. Тогда новое значение s_i будет равно 1, если $\text{rand}(0, 1) \leq b_i/t$, иначе s_i будет равно 0. Чтобы избежать ранней сходимости алгоритма, необходимо добавить ограничение на значение относительной частоты: $1 - p_g \leq p \leq p_g$, где p_g — пороговое значение. Алгоритм выполняется итерационно, после выполнения заданного числа итераций производится декодирование лучшего вектора в полученное решение, которое интерпретируется как оптимальное.

4. Алгоритм построения нечеткого классификатора

Алгоритм основан на нечеткой кластеризации обучающих данных. Каждому кластеру ставится в соответствие нечеткое правило. ТО-часть правила определяет класс, функции принадлежности нечетких термов имеют гауссовый тип

$$\mu = e^{-\left(\frac{x-s}{\sigma}\right)^2}$$

с параметром среднего, определяемого центром кластера, и с отклонением, которое представляет собой средневзвешенное квадратичное отклонение обучающих данных относительно центра. Одному классу соответствует один или несколько кластеров. Кластеризация выполняется известным алгоритмом Fuzzy C-means (FCM) [20]. FCM — итеративный алгоритм, в котором на каждой итерации происходит определение степени принадлежности данных кластеру и новых положений центров кластеров:

$$\mu_{ki} = \frac{1}{\sum_{j=1}^c \left(\frac{\|\mathbf{x}_k - \mathbf{v}_i\|}{\|\mathbf{x}_k - \mathbf{v}_j\|} \right)^{2/(h-1)}}, \quad v_{ij} = \frac{\sum_{k=1}^m (\mu_{kj})^h x_{kj}}{\sum_{k=1}^m (\mu_{kj})^h},$$

где c — число кластеров, μ_{ki} — степень принадлежности k -го экземпляра данных i -му кластеру, \mathbf{v}_i — вектор центра i -го кластера, h — константа больше 1 (обычно $h = 2$). Алгоритм начинает свою работу с определения значений векторов центров кластеров случайным образом и заканчивает, когда выполнено условие

$$\sum_{i=1}^c \|\mathbf{v}_i^{\text{new}} - \mathbf{v}_i^{\text{old}}\| \leq \varepsilon,$$

где $\mathbf{v}_i^{\text{old}}$ и $\mathbf{v}_i^{\text{new}}$ вектор i -го кластера до и после итерации соответственно, ε — малая величина (в данной работе $\varepsilon = 0,0001$). Число кластеров c задается перед выполнением алгоритма.

Ниже представлен алгоритм построения нечеткого классификатора. На вход алгоритма подается таблица наблюдений из z элементов и вектор \mathbf{c} , в котором c_j соответствует числу кластеров, на которое должно быть разбито пространство данных j -го класса, $j = \overline{1, m}$ (в экспериментах данной работы $c_j = 1$).

Алгоритм построения нечеткого классификатора.

1. Инициализация $k = 1$.

2. Разбить пространство данных k -го класса на c_k кластеров алгоритмом FCM.

3. Добавить в базу правил классификатора правила, соответствующие каждому кластеру. Выходом правила, соответствующего i -му кластеру, будет метка k -го класса, а параметрами среднего и отклонения функции принадлежности j -й переменной — соответственно

$$s_j = v_{ij}, \quad \sigma_j = \sqrt{\frac{2 \left(\sum_{p=1}^z \mu_{pi} (x_{pj} - v_{ij})^2 \right)}{\sum_{p=1}^z \mu_{pi}}}.$$

4. Если рассмотрены все классы ($k = m$), то завершить работу алгоритма, иначе $k = k + 1$ и перейти на шаг 2.

5. Эксперимент

5.1. Эмпирический критерий оценки эффективности алгоритма

Для сравнения эффективности алгоритмов отбора признаков необходим критерий, позволяющий оценить их работу. Задача отбора сводится к нахождению такого множества признаков, на котором классификатор показывал бы лучшую точность классификации при минимально возможном числе пробных вычислений точности в процессе работы алгоритма. Точность классификации является целевой функцией при отборе признаков по методу обертки, для вычисления ее значения необходимо построение классификатора на обучающих данных с пробным множеством признаков. Основная вычислительная нагрузка ложится на получение этого значения. Алгоритм считается эффективным, если с его помощью найдено множество признаков, на котором построенный классификатор показывает лучшую точность классификации. Если точности совпадают, то эффективным считается алгоритм с меньшим числом пробных вычислений точности. Поскольку в данной работе производится сравнение популяционных алгоритмов, вычислительной единицей будем считать итерацию. Чтобы трудоемкость итераций была одинакова, для всех сравниваемых алгоритмов выбран одинаковый размер популяции. На основании изложенного выше предлагается эмпирически оценить эффективность алгоритмов. Если проводить сравнение алгоритмов на одном наборе данных, то в качестве критерия эффективности можно взять точность классификации на итерации t_{\min} , на которой найдено лучшее решение одним из алгоритмов. При оценивании работы на нескольких наборах данных необходимо учитывать суммарную составляющую эффективности каждого алгоритма для каждого набора данных. Суммарная составляющая основана на нормированных значениях эффективности алгоритма по каждому набору. Введем эмпирический критерий эффективности работы алгоритма отбора признаков ξ , который вычисляется приведенным ниже способом.

Алгоритм нахождения значения критерия эффективности.

Вход: a — число вариантов алгоритмов отбора признаков, d — число наборов данных.

Выход: ξ_i — значение критерия эффективности для каждого варианта алгоритма, где $i = \overline{1, a}$.

1. Найти t_{\min} и $E_{jt_{\min}}^i$ ($i = \overline{1, a}$, $j = \overline{1, d}$), где $E_{jt_{\min}}^i$ — усредненное значение точности i -го алгоритма на j -м наборе данных и итерации t_{\min}^j .

2. Провести суммирование нормированных значений точностей:

$$\xi_i = \sum_{j=1}^d \frac{E_{jt_{\min}}^i - \min_{i=\overline{1, n}} E_{jt_{\min}}^i}{\max_{i=\overline{1, n}} E_{jt_{\min}}^i - \min_{i=\overline{1, n}} E_{jt_{\min}}^i}.$$

Большее значение критерия соответствует более эффективному алгоритму. Значение критерия варьируется в диапазоне $[0, a]$.

5.2. Определение рабочих параметров алгоритма

Для определения рабочих значений параметров σ_g , p_g , t_g алгоритма предлагается найти критерий эффективности работы с различными значениями параметров и выбрать те, на которых алгоритм покажет лучший результат. Для оценки эффективности использовались 10 реальных наборов данных из репозитория KEEL (<http://keel.es/>), информация о них представлена в табл. 1. Использовались следующие варианты параметров: $\sigma_g \in \{1, 2\}$, $p_g \in \{0,75, 0,85, 1\}$, $t_g \in \{1, 50, 200\}$. Точность классификации на каждой итерации для вычисления критерия эффективности бралась как среднее значение точности по 20 запускам алгоритма. Максимальное число итераций $t = 300$, размер популяции — 20. Лучшее значение критерия эффективности получил алгоритм с параметрами $p_g = 0,75$, $t_g = 1$ и $\sigma_g = 2$. Указанные значения параметров являются рабочими для выполнения сравнений с другими алгоритмами.

Таблица 1. Наборы данных для проведения эксперимента

Название набора	Число переменных	Число классов	Объем выборки
Wine	13	3	178
Heart	13	2	270
Cleveland	13	5	297
Vowel	13	11	990
Australian	14	2	690
Vehicle	18	4	846
Hepatitis	19	2	80
Bands	19	2	365
Ionosphere	33	2	351
Dermatology	34	6	358

5.3. Сравнение с алгоритмами-аналогами

Эффективность алгоритма случайного поиска с памятью (АСПП) сравнивалась с алгоритмом роящихся частиц (АРЧ) [21], бинарным алгоритмом «кукушкин поиск» (АКП) [22] и алгоритмом случайного поиска (АСП). В алгоритмах использовался размер популяции, равный 20, максимальное число итераций $t = 300$. Точность классификации на каждой итерации вычислялась как среднее значение по 20 запускам алгоритма. Параметры алгоритмов «кукушкин поиск» и роящихся частиц устанавливались в значения, рекомендуемые в [21, 22] соответственно. В табл. 2 показаны значения точности классификации на итерации t_{\min} ; в позициях лучших значений в скобках приведено t_{\min} . В табл. 3 приведены значения критерия ξ , алгоритм АСПП

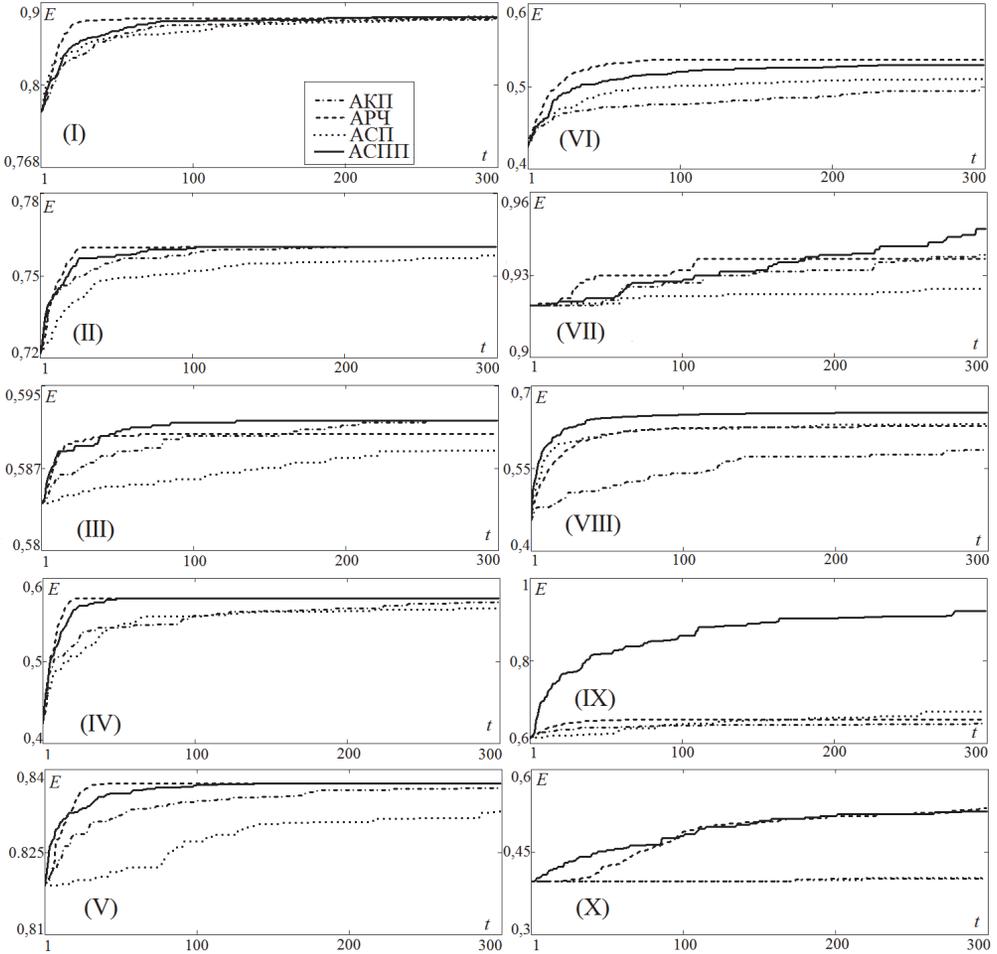


Рис. 2. Зависимость точности классификации от итерации на наборах данных (I) — Wine, (II) — Heart, (III) — Cleveland, (IV) — Vowel, (V) — Australian, (VI) — Vehicle, (VII) — Hepatitis, (VIII) — Bands, (IX) — Ionosphere, (X) — Dermatology.

Таблица 2. Значения точности классификации

Название набора	АКП	АРЧ	АСП	АСПП
Wine	0,879	0,881	0,883 (279)	0,882
Heart	0,757	0,761 (94)	0,751	0,759
Cleveland	0,589	0,59	0,586	0,592 (54)
Vowel	0,518	0,576 (21)	0,505	0,562
Australian	0,828	0,837 (28)	0,82	0,833
Vehicle	0,478	0,532 (81)	0,499	0,514
Hepatitis	0,937	0,936	0,925	0,947 (294)
Bands	0,571	0,628	0,63	0,652 (216)
Ionosphere	0,647	0,658	0,677	0,921 (280)
Dermatology	0,401	0,53(296)	0,402	0,524

Таблица 3. Значения критерия эффективности

Название набора	АКП	АРЧ	АСП	АСПП
Wine	0	0,43	1	0,8
Heart	0,66	1	0	0,9
Cleveland	0,62	0,88	0	1
Vowel	0,18	1	0	0,81
Australian	0,5	1	0	0,78
Vehicle	0	1	0,38	0,67
Hepatitis	0,53	0,5	0	1
Bands	0	0,71	0,73	1
Ionosphere	0	0,04	0,11	1
Dermatology	0	1	0,01	0,95
ξ	2,49	7,57	2,23	8,91

на каждом наборе данных показал либо лучшее, либо второе по значимости решение.

На рис. 2 представлены зависимости точности классификации от числа итераций поиска. Точность классификации определялась средним значением по 20 запускам алгоритмов. Бинарные алгоритмы роящихся частиц и случайного поиска с памятью демонстрируют наиболее быструю сходимость к лучшему значению.

6. Заключение

В статье предложен подход к отбору признаков, основанный на популяционном случайном поиске с памятью. Отбор признаков выполняется по методу оберток с использованием нечеткого классификатора. Алгоритм отбора признаков основан на стратегии случайного поиска, управляемого следующей эвристикой: шанс признака попасть в новую популяцию пропорционален частоте присутствия этого признака в предыдущих лучших решениях. Важным преимуществом алгоритма является сбалансированное сочетание диверсификации и интенсификации поиска решений. Диверсификацию обеспечивает

случайный поиск по всему пространству решений, интенсификацию — принятая эвристика, позволяющая помнить лучшие решения и тем самым учитывать ландшафт пространства поиска. Описан способ задания параметров алгоритма отбора признаков. Также был рассмотрен алгоритм построения нечеткого классификатора, основанный на нечеткой кластеризации обучающих данных. Проведено сравнение предложенного алгоритма отбора признаков с общеизвестными алгоритмами-аналогами. Результаты сравнений свидетельствуют о конкурентоспособности предлагаемого подхода.

Дальнейшее развитие алгоритма предполагает исследование повышения диверсификации за счет генерации исходных решений на основе квазислучайных последовательностей и хаотической оппозиции. Адаптационные возможности алгоритма предполагается увеличить путем изменения значений отклонения при генерации нормально распределенных чисел и путем введения изменяемого параметра в функцию гиперболического тангенса. Еще одна предполагаемая модификация алгоритма — введение списка ранее сформированных решений, что позволит предотвратить преждевременную сходимость алгоритма и сократить время работы алгоритма.

СПИСОК ЛИТЕРАТУРЫ

1. *Lin W.-C., Tsai C.-F., Ke S.-W., Hung C.-W., Eberle W.* Learning to detect representative data for large scale instance selection // *J. Syst. Software*. 2015. V. 106. P. 1–8.
2. *Tsai C.-F., Chen Z.-Y.* Towards high dimensional instance selection: An evolutionary approach // *Decision Support Syst.* 2014. V. 61. P. 79–92.
3. *Hughes G.* On the mean accuracy of statistical pattern recognizers // *IEEE Trans. Inf. Theory*. 1968. V. 14. No. 1. P. 55–63.
4. *Bolon-Canedo V., Sanchez-Marono N., Alonso-Betanzos A.* Feature Selection for High-Dimensional Data. London: Springer, 2015.
5. *Veerabhadrapa, Rangarajan L.* Multi-level dimensionality reduction methods using feature selection and feature extraction // *Int. J. Artif. Intell. Appl.* 2010. V. 1. No. 4. P. 54–68.
6. *Kohavi R., John G.H.* Wrappers for feature subset selection // *Artif. Intell.* 1997. V. 97. No. 1. P. 273–324.
7. *Цурко В.В., Михальский А.И.* Метод контрастирования для отбора информативных признаков по эмпирическим данным // *АиТ*. 2016. № 12. С. 136–154.
Tsurko V.V., Michalski A.I. The Contrast Features Selection with Empirical Data // *Autom. Remote Control*. 2016. V. 77. No. 12. P. 2212–2226.
8. *Dash M., Liu H.* Consistency-based search in feature selection // *Artif. Intell.* 2003. V. 151. No. 1–2. P. 155–176.
9. *Yusta S.C.* Different metaheuristic strategies to solve the feature selection problem // *Pattern Recognit. Lett.* 2009. V. 30. No. 5. P. 525–534.
10. *Learidi R., Boggia R., Terrile M.* Genetic algorithms as a strategy for feature selection // *J. Chemom.* 1992. V. 6. No. 5. P. 267–281.
11. *Yang J., Honavar V.* Feature subset selection using a genetic algorithm // *IEEE Intell. Syst.* 1998. V. 13. No. 2. P. 44–49.
12. *Chuang L.-Y., Tsai S.-W., Yang C.-H.* Improved binary particle swarm optimization using catfish effect for feature selection // *Expert Syst. Appl.* 2011. V. 38. No. 10. P. 12699–12707.

13. *Xue B., Zhang M., Browne W.N.* Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms // *Appl. Soft Comput.* 2014. V. 18. P. 261–276.
14. *Wan Y., Wang M., Ye Z., Lai X.* A feature selection method based on modified binary coded ant colony optimization algorithm // *Appl. Soft Comput.* 2016. V. 49. P. 248–258.
15. *Hancer E., Xue B., Zhang M., Karaboga D., Akay B.* Pareto front feature selection based on artificial bee colony optimization // *Inform. Sci.* 2018. V. 422. P. 462–479.
16. *Shunmugapriya P., Kanmani S.* A hybrid algorithm using ant and bee colony optimization for feature selection and classification // *Swarm Evol. Comput.* 2017. V. 36. P. 27–36.
17. *Diao R., Shen Q.* Feature selection with harmony search // *IEEE Trans. Syst. Man Cybern. Part B Cybern.* 2012. V. 42. No. 6. P. 1509–1523.
18. *Ходашинский И.А., Мех М.А.* Построение нечеткого классификатора на основе методов гармонического поиска // *Программирование.* 2017. № 1. С. 54–65.
19. *Sikdar U.K., Ekbal A., Saha S.* MODE: multiobjective differential evolution for feature selection and classifier ensemble // *Soft Comput.* 2015. V. 19. No. 12. P. 3529–3549.
20. *Bezdek J.C. Ehrlh R., Full W.* FCM: the fuzzy c-means clustering algorithm // *Comput. Geosci.* 1984. V. 10. No. 2–3. P. 191–203.
21. *Kennedy J., Eberhart R.* A discrete binary version of the particle swarm algorithm // *IEEE Int. Conf. Syst., Man, Cybernet.* 1997. V. 5. P. 4104–4108.
22. *Pereria L.A.M, Rodrigues D., Almedia T.N.S., et al.* A binary Cuckoo Search and its application for feature selection / *Cuckoo search and firefly algorithm. Studies in computational intelligence* 516. London: Springer, 2014. P. 141–154.

Статья представлена к публикации членом редколлегии В.И. Васильевым.

Поступила в редакцию 30.01.2018

После доработки 03.06.2018

Принята к публикации 08.11.2018