Отбор признаков для нечеткого классификатора с использованием алгоритма паукообразных обезьян

И.А. Ходашинский

E-mail: hodashn@rambler.ru

М.М. Немирович-Данченко

E-mail: nemdan2011@gmail.com

С.С. Самсонов

E-mail: samsonicx@mail.ru

Томский государственный университет систем управления и радиоэлектроники
Адрес: 634050, г. Томск, пр. Ленина, д. 40

Аннотация

В статье рассмотрены вопросы построения нечетких классификаторов. Определены три этапа построения нечеткого классификатора: формирование базы нечетких правил, отбор информативных признаков, оптимизация параметров функций принадлежности нечетких правил. Генерация структуры нечеткого классификатора выполнялась алгоритмом формирования базы правил по минимальным и максимальным значениям признаков в каждом классе. Применение такого алгоритма позволяет сформировать базу с минимальным числом правил, равным количеству меток классов в исходном классифицируемом наборе данных. Отбор признаков проводился методом обертки бинарным алгоритмом паукообразных обезьян. Отбор признаков, как метод предварительной обработки данных, может способствовать не только повышению эффективности алгоритмов обучения, но и повышению способности обобщения. В процессе отбора информативных признаков проведено исследование динамики изменения точности классификации по итерациям при различных параметрах бинарного алгоритма, выявлено влияние параметров бинарного алгоритма на скорость сходимости. Для оптимизации параметров антецедентов нечетких правил используется другой алгоритм обезьян, работающий с непрерывными числовыми данными. Эффективность нечетких классификаторов, построенных на правилах и признаках, отобранных с помощью указанных алгоритмов, проверена на наборах данных из репозитория KEEL. Проведены сравнения с алгоритмами-аналогами, протестированными на тех же наборах данных. Результаты сравнения показали, классификаторы могут быть разработаны с минимальным количеством правил и существенно сокращенным количеством признаков при статистически неразличимой точности с классификаторами-аналогами.

Ключевые слова: отбор признаков; метод обертки; бинарный алгоритм паукообразных обезьян; нечеткий классификатор; бинарные метаэвристики; нечеткое правило.

Цитирование: Ходашинский И.А., Немирович-Данченко М.М., Самсонов С.С. Отбор признаков для нечеткого классификатора с использованием алгоритма паукообразных обезьян // Бизнес-информатика. 2019. Т. 13. № 2. С. 29–42. DOI: 10.17323/1998-0663.2019.2.29.42

Введение

Классификация относится к области распознавания образов и машинного обучения. К настоящему времени разработано довольно много различных алгоритмов классификации (классификаторов). При выборе адекватного алгоритма необходимо учитывать, каким критериям должен соответствовать полученный результат: точность классификации, интерпретируемость полученного результата, время обучения, время классификации и т.д. Классифицировать объект — значит указать имя (метку) класса, к которому данный объект будет отнесен. Традиционный классификатор ставит исследуемому объекту в соответствие единственную метку класса. Нечеткий классификатор может присваивать степени принадлежности или мягкие метки всем классам. Достоинством нечетких классификаторов является их высокая интерпретируемость и точность [1–3]. Полученную интерпретируемую модель можно легко использовать в процессе принятия решений [4].

Одной из важных задач в интеллектуальном анализе данных и распознавании образов является отбор релевантных признаков. Признак — это индивидуальное измеримое свойство наблюдаемого объекта, процесса или явления. Используя набор признаков, алгоритм машинного обучения может выполнять классификацию. Задача отбора признаков формулируется как поиск оптимального подмножества признаков, обладающего минимальной избыточностью и максимальной прогностической способностью. Компактное подмножество исходного множества признаков позволяет уменьшить вычислительные затраты и повысить эффективность классификации [5]. Методы отбора признаков классифицируют по различным основаниям. В зависимости от вида данных в обучающей выборке (помеченные, непомеченные, частично помеченные) выделяют супервизорные (с учителем), не супервизорные (без учителя) и полу-супервизорные методы. В зависимости от способа взаимодействия между алгоритмами отбора признаков и классификаторами выделяют три группы методов отбора: фильтры (filters), обертки (wrappers) и встроенные (embedded) методы. Метод фильтров не зависит от алгоритма построения классификатора и обладает следующими достоинствами: невысокая вычислительная сложность, неплохая обобщающая способность, независимость от классификатора. Основной недостаток метода заключается в том, что признаки, как правило, отбираются

независимо друг от друга. Метод обертки включен в процесс построения классификатора и использует меру оценки эффективности классификатора для оценки отобранного подмножества признаков. Такое взаимодействие с классификатором, как правило, дает лучшие результаты, чем метод фильтра, однако при этом существенно увеличивается сложность алгоритма и существует риск переобучения [6, 7]. Во встроенных методах отбор признаков выполняется в процессе обучения и включается в алгоритм построения классификатора.

Отбор признаков по методу обертки относится к классу NP-сложных проблем, поэтому в этом случае целесообразно использовать эвристические и метаэвристические методы [8–10]. Метаэвристики можно разделить на два класса: дискретные и непрерывные. Дискретные метаэвристики, такие как генетический алгоритм [11–15], алгоритм муравьиной колонии [16–18] и гармонический поиск [19, 20] давно и успешно применяются для решения задач отбора признаков.

Отбор признаков — задача бинарной оптимизации, поэтому применению большинства непрерывных метаэвристик для решения задачи отбора предшествует этап выполнения бинаризации этих метаэвристик. Бинаризация чаще всего выполняется с использованием функций трансформации [21], которые определяют шанс изменения элементов вектора-решения с 0 на 1 и наоборот. По указанному принципу проводится отбор признаков алгоритмом роящихся частиц [22, 23], алгоритмом гравитационного поиска [24, 25], алгоритмом мозгового штурма [26], алгоритмом «учение через обучение» (teaching learning based optimization) [27], алгоритмом черной дыры [8], алгоритмом кузнечика [28], алгоритмом скопления сальпов [29, 30], алгоритмом муравьиного льва [31].

Недавно предложенный алгоритм паукообразных обезьян, имитирующий их социальное поведение, относится к алгоритмам роевого интеллекта [32]. Алгоритм показал хорошие результаты при оптимизации унимодальных и мультимодальных тестовых функций [32], а также при решении оптимизационных задач в электроэнергетике, электронике, телекоммуникации, биологии, медицине, обработке изображений и др. [33].

Согласно теореме о бесплатных завтраках (no free lunch theorem), ни один конкретный алгоритм не дает наилучших результатов для всех задач численной оптимизации [34, 35]. Как следствие, разрабатываются

новые алгоритмы оптимизации. При этом в работе [6] отмечается, что не существует единственно лучшего метода отбора признаков, поэтому усилия специалистов должны быть сосредоточены на поиске хорошего метода для каждой конкретной проблемы.

Цель работы — исследование бинарного алгоритма паукообразных обезьян для отбора по методу обертки оптимального числа информативных признаков при построении нечеткого классификатора.

1. Методы

1.1. Отбор признаков

Пусть имеется набор данных, содержащий Z образцов, каждый из которых характеризуется D признаками $X = \{x_1, x_2, \dots, x_D\}$. Отбор признаков состоит в том, чтобы выбрать d признаков из D ($d \leq D$), оптимизирующих целевую функцию. Иными словами, задача отбора признаков заключается в поиске на заданном множестве признаков X такого их подмножества, которое при уменьшении числа признаков не приводило бы к существенному уменьшению точности классификации. Решение по отбору признаков задается в виде бинарного вектора $\mathbf{S} = (s_1, s_2, \dots, s_D)^T$, в котором $s_i = 0$ означает, что i -й признак исключен из классификации, а $s_i = 1$ указывает на использование i -го признака в классификаторе. Каждое подмножество признаков оценивается точностью классификации.

1.2. Нечеткий классификатор

Традиционный классификатор задается следующей функцией:

$$f: \mathcal{R}^D \rightarrow \{0, 1\}^m,$$

где $f(\mathbf{x}; \boldsymbol{\theta}) = (h_1, h_2, \dots, h_m)^T$, причем $h_i = 1$, а $h_j = 0$, $j = \overline{1, m}$, $i \neq j$, когда объект, заданный вектором $\mathbf{x}$, принадлежит к классу $c_i \in C$, $C = \{c_1, c_2, \dots, c_m\}$ — множество классов, $\boldsymbol{\theta}$ — вектор параметров классификатора.

Нечеткий классификатор определяет класс объекта с некоторой степенью уверенности:

$$f: \mathcal{R}^D \rightarrow [0, g]^m.$$

Продукционное правило нечеткого классификатора с отбором признаков задается следующим образом:

$$R_j: \text{IF } s_1 \wedge x_1 = A_{1j} \text{ AND } s_2 \wedge x_2 = A_{2j} \text{ AND } \dots \\ \text{AND } s_D \wedge x_D = A_{Dj} \text{ THEN class} = c_j, j = 1, \dots, R,$$

где A_{kj} — нечеткий терм, характеризующий k -й признак в j -м правиле ($k = 1, \dots, D$); запись $s_i \wedge x_i$ указывает на наличие ($s_i = 1$) или отсутствие ($s_i = 0$) признака в классификаторе; R — число правил.

Ниже приведены формулы, определяющие итоговое решение.

$$\text{class} = c_t, \quad t = \arg \max_{1 \leq j \leq m} \{\beta_j\},$$

$$\mu_j(\mathbf{x}_p) = \mu_{A_{j1}}(x_{p1}) \cdot \dots \cdot \mu_{A_{jn}}(x_{pn}) = \prod_{k=1}^n \mu_{A_{jk}}(x_{pk}),$$

$$\beta_t(\mathbf{x}_p) = \sum_{\substack{R_j \\ C_j = \text{class } t}} \mu_j(\mathbf{x}_p) = \sum_{\substack{R_j \\ C_j = \text{class } t}} \prod_{k=1}^n \mu_{A_{jk}}(x_{pk}),$$

где $\mu_{A_{jk}}(x_{pk})$ — функция принадлежности нечеткого термина A_{jk} в точке x_{pk} .

На наборе данных $\{(\mathbf{x}_p; c_p), p = 1, \dots, Z\}$, определим меру точности классификации [20, 24]:

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

где $f(\mathbf{x}_p; \boldsymbol{\theta}, \mathbf{S})$ — выход нечеткого классификатора, заданного вектором параметрами $\boldsymbol{\theta}$ и вектором признаков $\mathbf{S}$.

Отбор признаков по методу обертки относится к классу NP-сложных проблем, для ее решения предлагается метаэвристика под названием «бинарный алгоритм паукообразных обезьян» (binary spider monkey optimization algorithm).

1.3. Бинарный алгоритм паукообразных обезьян

Предложенный авторами работы [32] популяционный алгоритм паукообразных обезьян работает в непрерывном пространстве поиска и поэтому не подходит для решения задач бинарной оптимизации. Авторы работы [36] адаптировали алгоритм для работы в двоичном пространстве поиска. В работах [33, 36] утверждается, что предложенный алгоритм исключительно эффективен для двоичной оптимизации, он быстро сходится и имеет меньше шансов застрять на локальных оптимумах.

В процессе работы бинарного алгоритма паукообразных обезьян (БАПО) популяция делится на несколько групп, в каждой из которых присутствует свой лидер с наилучшим решением в группе. В популяции также присутствует и глобальный лидер, ассоциирующийся с лучшим решением, най-

денным в популяции. Работа алгоритма может быть описана следующими основными этапами: инициализация, фаза локального лидера, фаза глобального лидера, фаза принятия решения. БАПО задается следующими параметрами: M – количество обезьян в популяции; позиция каждой i -й обезьяны представляет решение, задаваемое вектором $S_i = (s_{i,1}, s_{i,2}, \dots, s_{i,D})$; L – максимальное количество локальных лидеров; T – количество итераций; $p, pr \in [0, 1]$ – константы.

Инициализация j -го элемента i -го решения производится следующим образом:

$$s_{i,j} = \begin{cases} 0 & \text{если } \text{rand}(0, 1) < p \\ 1 & \text{иначе.} \end{cases} \quad (1)$$

В БАПО основные уравнения непрерывного алгоритма модифицируются с использованием логических операторов AND ($\otimes$), OR ($+$) и XOR ($\oplus$).

Фаза локального лидера используется для обновления позиций обезьян в соответствии с позицией локального лидера по следующей формуле:

$$s_{i,j}^{\text{new}} = \begin{cases} s_{i,j} \oplus (b \otimes (LL_{k,j} \oplus s_{i,j})) + (d \otimes (s_{r,j} \oplus s_{i,j})), & \text{если } \text{rand} > pr \\ s_{i,j}, & \text{иначе,} \end{cases} \quad (2)$$

где b, d принимают случайные значения из множества $\{0; 1\}$; LL_k – вектор, определяющий координаты локального лидера k -й группы, S_r – вектор, определяющий координаты случайной обезьяны из той же группы.

Фаза глобального лидера задает координаты обезьян в соответствии с положением глобального лидера

$$s_{i,j}^{\text{new}} = \begin{cases} s_{i,j} \oplus (b \otimes (GL_{k,j} \oplus s_{i,j})) + (d \otimes (s_{r,j} \oplus s_{i,j})), & \text{если } \text{rand} > P_i \\ s_{i,j}, & \text{иначе,} \end{cases} \quad (3)$$

где GL – вектор, определяющий координаты глобального лидера,

$$P_i = 0,9 \cdot \frac{E(\theta, S_i)}{\max E(\theta, S)} + 0,1. \quad (4)$$

Фаза принятия решения задает координаты обезьян в соответствии с положением глобального и локального лидеров.

$$s_{i,j}^{\text{new}} = \begin{cases} s_{i,j} \oplus (b \otimes (LL_{k,j} \oplus s_{i,j})) + (d \otimes (GL_j \oplus s_{i,j})), & \text{если } \text{rand} > pr \\ s_{i,j}, & \text{иначе.} \end{cases} \quad (5)$$

Псевдокод БАПО представлен ниже:

Вход: $M, C, T, p, pr, \theta, D$.

Выход: S_{best} .

цикл по i от 1 до M

цикл по j от 1 до D

инициализация популяции по формуле 1;

конец цикла

конец цикла

$C_{\text{current}} := 1$;

цикл по i от 1 до M

$GL := \text{search_global_best}(E(\theta, S_i))$;

$LL := \text{search_group_best}(E(\theta, S_i))$;

конец цикла

цикл по t от 1 до T

$GL_{\text{old}} := GL$;

цикл по i от 1 до M

цикл по j от 1 до D

вычисление по формуле 2;

конец цикла

конец цикла

цикл по i от 1 до M

$LL := \text{search_group_best}(E(\theta, S_i))$;

конец цикла

цикл по i от 1 до M

цикл по j от 1 до D

вычисления по формулам 3 и 4;

конец цикла

конец цикла

цикл по i от 1 до M

$GL := \text{search_global_best}(E(\theta, S_i))$;

конец цикла

цикл по i от 1 до M

цикл по j от 1 до D

вычисление по формуле 5;

конец цикла

конец цикла

цикл по i от 1 до M

$GL := \text{search_global_best}(E(\theta, S_i))$;

$LL := \text{search_group_best}(E(\theta, S_i))$;

конец цикла

если ($GL_{\text{old}} = GL$) **то**

$C_{\text{current}} := C_{\text{current}} + 1$;

если ($C_{\text{current}} > C$) **то**

Объединить всех агентов в одну

группу;

$C_{\text{current}} := 1$;

иначе

Разделить наибольшую группу на две и

для новой группы сгенерировать новые

начальные значения по формуле 1;

конец если

конец если

конец цикла

Вывод $S_{\text{best}} := \text{best_solution}(S)$;

1.4. Алгоритм формирования базы нечетких правил

Алгоритм формирует компактную начальную базу нечетких правил, в которой каждый класс представлен одним правилом [20, 24]. Правило формируется на основе минимальных и максимальных значений по каждому классу в обучающей таблице наблюдений $\{(x_p; t_p)\}$.

Алгоритм имеет следующие параметры: D – число признаков, m – число классов.

Вход: $D, m, \{(x_p; t_p)\}$.

Выход: База нечетких правил Ω^* .

Инициализация пустой базы правил $\Omega := \emptyset$;

цикл по j от 1 до m

цикл по k от 1 до D

нахождение минимального и максимального значения признака k в классе j

$\max_{class_{jk}} := \max_p(x_{pk}); \min_{class_{jk}} := \min_p(x_{pk});$

генерация нечеткого термина A_{jk} , накрывающего интервал $[\min_{class_{jk}}, \max_{class_{jk}}]$;

конец цикла

формирование правила R_j на основе термов

A_{jk} , относящего наблюдение к c_j классу;

$\Omega^* := \Omega \cup \{R_j\}$;

конец цикла

вывод Ω^* .

1.5. Наборы данных

Проверка эффективности работы алгоритма отбора признаков на основе БАПО была выполнена на 38 наборах данных из репозитория KEEL (Knowledge Extraction based on Evolutionary Learning, <https://sci2s.ugr.es/keel/datasets.php>). В таблице 1 приведено описание использованных наборов данных. Здесь #Fall – количество признаков в исходном наборе данных, #I – количество образцов, #C – количество классов.

2. Результаты

Эксперименты были проведены в соответствии с принципом 10-кратной кросс-валидации, исследуемый набор данных разбивался на обучающие и тестовые выборки в соотношении 9:1. Построение классификатора осуществлялась на обучающих выборках, а оценка точности производилась на тестовых. Усредненное значение точности на тестовых и

Таблица 1.

Описание наборов данных

Набор данных	#Fall	#I	#C	Набор данных	#Fall	#I	#C
appendicitis	7	106	2	phoneme	5	5404	2
balance	4	625	3	pima	8	768	2
banana	2	5300	2	ring	20	7400	2
bupa	6	345	2	satimage	36	6435	7
cleveland	13	297	5	segment	19	2310	7
coil2000	85	9822	2	shuttle	9	58000	7
contraceptive	9	1473	3	sonar	60	208	2
dermatology	34	358	6	spambase	57	4597	2
ecoli	7	336	8	spectfheart	44	267	2
glass	9	214	7	texture	40	5500	11
haberman	3	306	2	thyroid	21	7200	3
heart	13	270	2	titanic	3	2201	2
hepatitis	19	80	2	twonorm	20	7400	2
ionosphere	33	351	2	vehicle	18	846	4
iris	4	150	3	vowel	13	990	11
monk-2	6	432	2	wdbc	30	569	2
newthyroid	5	215	3	wine	13	178	3
page-blocks	10	5472	5	wisconsin	9	683	2
penbased	16	10992	10	yeast	8	1484	10

обучающих данных определялось путем вычисления среднего значения:

$$Accuracy = \frac{1}{V} \sum_{j=1}^V E(\theta, S),$$

где V – число запусков БАПО на одной и той же выборке, в наших экспериментах $V = 10$.

Количество отобранных признаков определялось как среднее на всех обучающих выборках.

БАПО имеет четыре настраиваемых параметра: количество лидеров (C), пороговые значения p , pr и размер популяции (M). Значения параметров алгоритма по умолчанию следующие: размер популяции – 40, число лидеров – 8, число итераций – 100, параметры алгоритма p и pr равны 0,4 и 0,5 соответственно. Для наборов данных, имеющих 16 и больше признаков, размер популяции увеличен до 70.

В процессе отбора информативных признаков проведено исследование динамики изменения точности классификации по итерациям при раз-

личных параметрах БАПО. Кривые сходимости для четырех наборов данных Ionosphere, Hepatitis, Spectfheart, Wine представлены на рисунках 1–4. Здесь обозначение $p00$ соответствует значению параметра p , равному 0,0; аналогичные обозначения приняты для параметра pr ; обозначение $pop10$ соответствует размеру популяции, равному 10; обозначение $L2$ соответствует числу лидеров $C = 2$.

С использованием t -критерия парных выборок проверялись распределения средней точности на наборе данных Wine при изменении количества лидеров. Нулевая гипотеза сформулирована на уровне значимости 0,05 следующим образом: сравниваемые распределения одинаковы. Результаты сравнения (p -value) приведены в таблице 2. Здесь s – знак разности ($\#L_i - \#L_j$), где i – указатель столбца, а j – указатель строки.

В таблице 3 приведены результаты отбора признаков без оптимизации параметров нечеткого классификатора. Здесь $\#F$ – усредненное количество отобранных признаков, $\#Rt$ – отношение исходного количества признаков к отобранному алгоритмом, $\#Tra$ – усредненное значение точности на обучающей выборке, $\#Tst$ – усредненное значение точности на тестовой выборке.

3. Дискуссия

Экспериментальные исследования по отбору признаков с использованием БАПО на 38 наборах данных позволяют сделать вывод о том, что количество признаков может быть сокращено в среднем в три раза при приемлемой точности классификации. Небольшие расхождения между значениями точности классификации на обучающих и тестовых данных могут свидетельствовать об отсутствии переобучения.

Пороговым значением p можно регулировать скорость сходимости алгоритма. Малые значения параметра p , равные 0,0, или 0,1, уменьшают скорость сходимости. При значениях параметра близких к 1 алгоритм быстро сходится, но не всегда к глобальному оптимуму. Рекомендуемые значения параметра p должны быть установлены близкими к значению, равному 0,5. Влияние параметра pr менее выражено, устанавливать его рекомендуется в диапазоне 0,3–0,6. Размер популяции рекомендуется устанавливать близким к максимальному – 80–100. Число лидеров играет менее заметную роль в сходимости алгоритма, при этом оптимальным будет значение из интервала 6–10.

Рис. 1. Кривые сходимости алгоритма для одиннадцати значений параметра p

Рис. 2. Кривые сходимости алгоритма для одиннадцати значений параметра pr

Рис. 3. Кривые сходимости алгоритма для десяти значений параметра «размер популяции»

Рис. 4. Кривые сходимости алгоритма для восьми значений параметра C

В таблице 4 приведено сравнение результатов отбора признаков алгоритмом БАПО с последующей оптимизацией параметров нечеткого классификатора алгоритмом обезьян (АО) [37] и результатов работы классификаторов-аналогов D-MOFARC и FARC-HD [38]. В данной таблице указано среднее количество правил ($\#R$), количество признаков ($\#F$), средняя точность на обучающих выборках ($\#L$) и тестовых выборках ($\#T$).

Используя ранее описанный принцип 10-кратной кросс-валидации для каждого набора признаков, полученных алгоритмом БАПО на каждом отдельном наборе данных, осуществлялось построение классификатора на обучающих выборках. Параметры классификатора оптимизировались алгоритмом обезьян. Усредненное значение точности на тестовых и обучающих данных определялось вычислением среднего значения. Далее для одного набора данных отбирался классификатор, имеющий наибольшую усредненную точность на тестовых выборках.

Оценка статистической значимости различий в точности классификации, количестве признаков и количестве правил классификаторов, сформиро-

ванных алгоритмами обезьян и классификаторов-аналогов, проводилась на основе критерия парных сравнений Уилкоксона.

1. Критерий указывает на значимое различие между количеством правил в классификаторах на основе алгоритма обезьян и классификаторах-аналогах ($p\text{-value} < 0,001$).

2. Критерий указывает на значимое различие между количеством признаков в классификаторах на основе алгоритма обезьян и классификаторах-аналогах ($p\text{-value} < 0,001$).

3. Критерий указывает на отсутствие значимого различия между точностью классификации на тестовых выборках в сравниваемых классификаторах: $p\text{-value} = 0,153$ для пары (БАПО+АО – D-MOFARC), $p\text{-value} = 0,148$ для пары (БАПО+АО – FARC-HD).

Можно сделать следующие выводы по результатам статистического сравнения: сокращение количества правил и признаков не привело к существенному уменьшению точности классификации; классификаторы с меньшим количеством признаков и правил

Таблица 2.

**Сравнение средней точности на наборе Wine
в зависимости от числа лидеров**

Число лидеров	#L_2		#L_4		#L_6		#L_8		#L_10		#L_12		#L_14		#L_16	
	s	t	s	t	s	t	s	t	s	t	s	t	s	t	s	t
#L_2			+	0,238	+	0,000	+	0,006	–	0,058	–	0,000	–	0,227	–	0,000
#L_4	–	0,238			+	0,010	+	0,201	–	0,001	–	0,000	–	0,001	–	0,000
#L_6	–	0,000	–	0,010			–	0,009	–	0,000	–	0,000	+	0,000	–	0,000
#L_8	–	0,006	–	0,201	+	0,009			–	0,001	–	0,000	–	0,004	–	0,000
#L_10	+	0,058	+	0,001	+	0,000	+	0,001			–	0,000	+	0,974	–	0,000
#L_12	+	0,000	+	0,000	+	0,000	+	0,000	+	0,000			+	0,000	–	0,111
#L_14	+	0,227	+	0,001	+	0,000	+	0,004	–	0,974	–	0,000			–	0,000
#L_16	+	0,000	+	0,000	+	0,000	+	0,000	+	0,000	+	0,111	+	0,000		

Таблица 3.

**Результаты отбора признаков
без оптимизации параметров классификатора**

Набор данных	#F	#Rt	#Tra	#Tst	Набор данных	#F	#Rt	#Tra	#Tst
appendicitis	2,9	2,4	82,81	76,36	phoneme	4	1,3	76,35	76,22
balance	4	1	46,24	46,24	pima	4,1	2	73,63	72,67
banana	1	2	59,42	59,36	ring	1	20	58,51	58,39
bupa	2,7	2,2	60,61	58,17	satimage	10,2	3,5	65,48	64,69
cleveland	8,2	1,6	56,38	54,15	segment	9	2,1	81,80	81,90
coil2000	31,3	2,7	74,93	74,96	shuttle	3,9	2,3	88,82	88,81
contraceptive	2,8	3,2	44,32	44,06	sonar	27,2	2,2	85,05	73,46
dermatology	21,6	1,6	97,95	90,99	spambase	23,7	2,4	75,01	74,27
ecoli	3,9	1,8	54,20	51,22	spectfheart	20,7	2,1	91,21	84,55
glass	6,2	1,5	60,29	56,12	texture	12,1	3,3	74,01	73,69
haberman	1,1	2,7	70,44	67,96	thyroid	1,8	11,7	91,06	90,93
heart	5,4	2,4	79,14	74,21	titanic	1,4	2,1	77,60	77,60
hepatitis	8,1	2,3	94,45	85,51	twonorm	19,7	1	96,87	96,74
ionosphere	16,8	2	93,06	91,58	vehicle	5,9	3,1	49,28	47,87
iris	1,8	2,2	96,81	94,67	vowel	5,3	2,5	47,28	47,68
monk-2	1	6	63,89	63,95	wdbc	10,4	2,9	97,06	95,18
newthyroid	3,5	1,4	97,88	96,77	wine	6,2	2,1	96,5	96,08
page-blocks	2	5	80,42	80,52	wisconsin	5,3	1,7	94,45	93,59
penbased	7	2,3	51,58	51,84	yeast	5,1	1,6	43,67	41,91

Таблица 4.

Результаты сравнения работы нескольких алгоритмов

№	Набор данных	Алгоритмы											
		БАПО+АО				D-MOFARC				FARC-HD			
		#R	#F	#L	#T	#R	#F	#L	#T	#R	#F	#L	#T
1	balance	3	4	87,5	86,7	20,1	4	89,4	85,6	18,8	4	92,2	91,2
2	banana	2	2	78,9	78,4	8,7	2	90,3	89	12,9	2	86	85,5
3	bupa	2	3	74,2	70,4	7,7	6	82,8	70,1	10,6	6	78,2	66,4
4	cleveland	5	8	60,7	57,1	45,6	13	90,9	52,9	42,1	13	82,2	58,3
5	ecoli	7	4	72,6	64,0	26,2	7	94	82,7	32,2	7	91,6	81,2
6	glass	7	7	69	63,3	27,4	9	95,2	70,6	18,2	9	79	69
7	haberman	2	1	79,2	74,8	9,2	3	81,7	69,4	5,7	3	79,2	73,5
8	heart	2	6	81,6	80,2	18,7	13	94,4	84,4	27,8	13	93,1	83,7
9	hepatitis	2	10	96,8	91,1	11,4	19	100	90	10,4	19	99,4	88,7
10	iris	3	2	97,8	96,3	5,6	4	98,1	96	4,4	4	98,6	95,3
12	newthyroid	3	4	98,7	96,8	9,5	5	99,8	95,5	9,6	5	99,2	94,4
13	page-blocks	5	2	95,5	95,3	21,5	10	97,8	97	18,4	10	95,5	95
14	penbased	10	6	73,7	72,6	119,2	16	97,4	96,2	152,7	16	97	96
15	phoneme	2	4	79,9	79,2	9,3	5	84,8	83,5	17,2	5	83,9	82,4
16	pima	2	4	75,5	74,3	10,4	8	82,3	75,5	20,2	8	82,3	76,2
17	segment	7	8	88,7	82,3	26,2	19	98	96,6	41,1	19	94,8	93,3
23	spambase	2	25	77,9	76,3	24,3	57	91,7	90,5	30,5	57	92,4	91,6
18	thyroid	3	2	99,6	99,3	5,9	21	99,3	99,1	4,9	21	94,3	94,1
19	titanic	2	2	79,5	79,0	10,4	3	78,9	78,7	4,1	3	79,1	78,8
20	twonorm	2	20	97,5	97,1	10,2	20	94,5	93,1	60,4	20	96,6	95,1
21	wine	3	8	98,9	96,6	8,6	13	100	95,8	8,3	13	100	95,5
22	wisconsin	2	5	96,8	96,4	9	9	98,6	96,8	13,6	9	98,3	96,2

более предпочтительны в силу большей возможности их интерпретации.

Заключение

В статье рассмотрен подход к построению нечетких классификаторов. База правил классификатора формировалась на основе минимальных и максимальных значений признака в каждом классе. Отбор признаков проводился в режиме обертки с использованием бинарного алгоритма паукообразных обезьян. Значения параметров функций принадлежности, используемых в нечетких правилах, были получены с помощью алгоритма обезьян, работающего в непрерывной области поиска. Эффективность предлагаемого подхода была проверена экспериментально на нескольких наборах данных. Первый

эксперимент, проведенный на 38 наборах данных, был направлен на сокращение исходного множества и поиск информативных признаков, а также на исследование влияния параметров бинарного алгоритма паукообразных обезьян на сходимость алгоритма. Во втором эксперименте проведено сравнение с двумя хорошо известными метаэвристическими методами с использованием критерия парных сравнений Уилкоксона. Проведенные исследования подтвердили эффективность и конкурентные преимущества предлагаемого подхода. ■

Благодарности

Исследование выполнено в рамках проектной части государственного задания министерства образования и науки Российской Федерации на 2017-2019 гг. № 2.3583.2017/4.6.

Литература

1. Alonso J.M., Castiello C., Mencar C. Interpretability of fuzzy systems: Current research trends and prospects // Springer Handbook of Computational Intelligence. Berlin: Springer, 2015. P. 219–237.
2. Мех М.А., Ходашинский И.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.
3. Zhang Y., Ishibuchi H., Wang S. Deep Takagi–Sugeno–Kang fuzzy classifier with shared linguistic fuzzy rules // IEEE Transactions on Fuzzy Systems. 2018. Vol. 26. No 3. P. 1535–1549.
4. Lucca G., Dimuro G.P., Fernández J., Bustince H., Bedregal B.R.C., Sanz J.A. (2019) Improving the performance of fuzzy rule-based classification systems based on a nonaveraging generalization of CC-integrals named C_{PI2} -integrals / G. Lucca [et al.] // IEEE Transactions on Fuzzy Systems. 2019. Vol. 27. No 1. P. 124–134.
5. Zhang P., Gao W., Liu G. Feature selection considering weighted relevancy // Applied Intelligence. 2018. Vol. 48. P. 4615–4625.
6. Bolon-Canedo V., Sanchez-Marono N., Alonso-Betanzos A. Feature selection for high-dimensional data. London: Springer, 2015.
7. Cai J., Luo J., Wang S., Yang S. Feature selection in machine learning: A new perspective // Neurocomputing. 2018. No 300. P. 70–79.
8. Pashaei E., Aydin N. Binary black hole algorithm for feature selection and classification on biological data // Applied Soft Computing. 2017. No 56. P. 94–106.
9. Yusta S. C. Different metaheuristic strategies to solve the feature selection problem // Pattern Recognition Letters. 2009. No 30. P. 525–534.
10. Kohavi R., John G.H. Wrappers for feature subset selection // Artificial Intelligence. 1997. No 97. P. 273–324.
11. Dong H., Li T., Ding R., Sun J. A novel hybrid genetic algorithm with granular information for feature selection and optimization // Applied Soft Computing. 2018. No 65. P. 33–46.
12. Sayed S., Nassef M., Badr A., Farag I. A nested genetic algorithm for feature selection in high-dimensional cancer Microarray datasets // Expert Systems with Applications. 2019. No 121. P. 233–243.
13. Ma B., Xia Y. A tribe competition-based genetic algorithm for feature selection in pattern classification // Applied Soft Computing. 2017. No 58. P. 328–338.
14. Wu Y.-L., Tang C.-Y., Hor M.-K., Wu P.-F. Feature selection using genetic algorithm and cluster validation // Expert Systems with Applications. 2011. No 38. P. 2727–2732.
15. Sikora R., Piramuthu S. Framework for efficient feature selection in genetic algorithm based data mining // European Journal of Operational Research. 2007. No 180. P. 723–737.
16. Vieira S.M., Sousa J.M.C., Runkler T.A. Ant colony optimization applied to feature selection in fuzzy classifiers // IFSA 2007, LNAI 4529. Berlin: Springer, 2007. P. 778–788.
17. Ghimatgar H., Kazemi K., Helfroush M.S., Aarabi A. An improved feature selection algorithm based on graph clustering and ant colony optimization // Knowledge-Based Systems. 2018. No 159. P. 270–285.
18. Dadaneh B.Z., Markid H.Y., Zakerolhosseini A. Unsupervised probabilistic feature selection using ant colony optimization // Expert Systems with Applications. 2016. No 53. P. 27–42.
19. Diao R., Shen Q. Feature Selection with Harmony Search // IEEE Transactions on Systems, Man, and Cybernetics – PART B: Cybernetics. 2012. No 42. P. 1509–1523.
20. Ходашинский И.А., Мех М.А. Построение нечеткого классификатора на основе методов гармонического поиска // Программирование. 2017. № 1. С. 54–65.
21. Mirjalili S., Lewis A. S-shaped versus V-shaped transfer functions for binary particle swarm optimization // Swarm and Evolutionary Computation. 2013. No 9. P. 1–14.
22. Yadav S., Ekbal A., Saha S. Feature selection for entity extraction from multiple biomedical corpora: A PSO-based approach // Soft Computing. 2018. No 22. P. 6881–6904.
23. Ajit Krishna N.L., Deepak V. K., Manikantan K., Ramachandran S. Face recognition using transform domain feature extraction and PSO-based feature selection // Applied Soft Computing. 2014. No 22. P. 141–161.
24. Bardamova M., Konev A., Hodashinsky I., Shelupanov A. A fuzzy classifier with feature selection based on the gravitational search algorithm // Symmetry. 2018. No 10 (11), 609.
25. Rashedi E., Nezamabadi-pour H. Feature subset selection using improved binary gravitational search algorithm // Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology. 2014. No 26. P. 1211–1221.
26. Papa J.P., Rosa G.H., de Souza A.N., Afonso L.C.S. Feature selection through binary brain storm optimization // Computers and Electrical Engineering. 2018. No 72. P. 468–481.
27. Kiziloz H.E., Deniz A., Dokeroglu T., Cosar A. Novel multiobjective TLBO algorithms for the feature subset selection problem // Neurocomputing. 2018. No 306. P. 94–107.
28. Binary grasshopper optimisation algorithm approaches for feature selection problems / Mafarja M. [et al.] // Expert Systems with Applications. 2019. No 117. P. 267–286.
29. Sayed G.I., Khoriba G., Haggag M.H. A novel chaotic salp swarm algorithm for global optimization and feature selection // Applied Intelligence. 2018. No 48. P. 3462–3481.
30. An efficient binary salp swarm algorithm with crossover scheme for feature selection problems / Faris H. [et al.] // Knowledge-Based Systems. 2018. No 154. P. 43–67.

31. Emary E., Zawbaa H.M., Hassanien A.E. (2016) Binary ant lion approaches for feature selection // *Neurocomputing*. No 213. P. 54–65.
32. Bansal J.C., Sharma H., Jadon S.S. Clerc M. Spider monkey optimization algorithm for numerical optimization // *Memetic Computing*. 2014. No 6. P. 31–47.
33. Agrawal V., Rastogi R., Tiwari D.C. Spider monkey optimization: a survey // *International Journal of System Assurance Engineering and Management*. 2018. No 9. P. 929–941.
34. Wolpert D.H. The existence of a priori distinctions between learning algorithms // *Neural Computation*. 1996. No 8. P. 1341–1390.
35. Wolpert D.H. The lack of a priori distinctions between learning algorithms // *Neural Computation*. 1996. No 8. P. 1391–1420.
36. Singh U., Salgotra R., Rattan M. A novel binary spider monkey optimization algorithm for thinning of concentric circular antenna arrays // *IETE Journal of Research*. 2016. No 62. P. 736–744.
37. Hodashinsky I.A., Samsonov S.S. Design of fuzzy rule based classifier using the monkey algorithm // *Business Informatics*. 2017. No 1. P. 61–67.
38. Fazzolari F., Alcalá R., Herrera F. A multi-objective evolutionary method for learning granularities based on fuzzy discretization to improve the accuracy-complexity trade-off of fuzzy rule-based classification systems: D-MOFARC algorithm // *Applied Soft Computing*. 2014. No 24. P. 470–481.

Об авторах

Ходашинский Илья Александрович

доктор технических наук, профессор;

профессор кафедры комплексной информационной безопасности электронно-вычислительных систем, Томский государственный университет систем управления и радиоэлектроники, 634050, г. Томск, пр. Ленина, д. 40;

E-mail: hodashn@rambler.ru

Немирович-Данченко Михаил Михайлович

доктор физико-математических наук;

профессор кафедры комплексной информационной безопасности электронно-вычислительных систем, Томский государственный университет систем управления и радиоэлектроники, 634050, г. Томск, пр. Ленина, д. 40;

E-mail: nemdan2011@gmail.com

Самсонов Сергей Сергеевич

аспирант кафедры комплексной информационной безопасности электронно-вычислительных систем, Томский государственный университет систем управления и радиоэлектроники, 634050, г. Томск, пр. Ленина, д. 40;

E-mail: samsonicx@mail.ru

Feature selection for fuzzy classifier using the spider monkey algorithm

Ilya A. Hodashinsky

E-mail: hodashn@rambler.ru

Mikhail M. Nemirovich-Danchenko

E-mail: nemdan2011@gmail.com

Sergey S. Samsonov

E-mail: samsonicx@mail.ru

Tomsk State University of Control Systems and Radioelectronics (TUSUR)

Address: 40, Prospect Lenina, Tomsk 634050, Russia

Abstract

In this paper, we discuss the construction of fuzzy classifiers by dividing the task into the three following stages: the generation of a fuzzy rule base, the selection of relevant features, and the parameter optimization of membership functions for fuzzy rules. The structure of the fuzzy classifier is generated by forming the fuzzy rule base with

use of the minimum and maximum feature values in each class. This allows us to generate the rule base with the minimum number of rules, which corresponds to the number of class labels in the dataset to be classified. Feature selection is carried out by a binary spider monkey optimization (BSMO) algorithm, which is a wrapper method. As a data preprocessing procedure, feature selection not only improves the efficiency of training algorithms but also enhances their generalization capability. In the process of feature selection, we investigate the dynamics of changes in classification accuracy, iteration by iteration, for various parameter values of the binary algorithm and analyze the effect of its parameters on its convergence rate. The parameter optimization of fuzzy rule antecedents uses another spider monkey optimization (SMO) algorithm that processes continuous numerical data. The performance of the fuzzy classifiers based on the rules and features selected by these algorithms is tested on some datasets from the KEEL repository. Comparison with two competitor algorithms on the same datasets is carried out. It is shown that fuzzy classifiers with the minimum number of rules and a significantly reduced number of features can be developed with their accuracy being statistically similar to that of the competitor classifiers.

Key words: feature selection; wrapper method; binary spider monkey algorithm; fuzzy classifier; binary metaheuristics; fuzzy rule.

Citation: Hodashinsky I.A., Nemirovich-Danchenko M.M., Samsonov S.S. (2019) Feature selection for fuzzy classifier using the spider monkey algorithm. *Business Informatics*, vol. 13, no 2, pp. 29–42. DOI: 10.17323/1998-0663.2019.2.29.42

References

1. Alonso J.M., Castiello C., Mencar C. (2015) Interpretability of fuzzy systems: Current research trends and prospects. *Springer Handbook of Computational Intelligence*. Berlin: Springer, pp. 219–237.
2. Mekh M.A., Hodashinsky I.A. (2017) Comparative analysis of differential evolution methods to optimize parameters of fuzzy classifiers. *Journal of Computer and Systems Sciences International*, no 56, pp. 616–626.
3. Zhang Y., Ishibuchi H., Wang S. (2018) Deep Takagi–Sugeno–Kang fuzzy classifier with shared linguistic fuzzy rules. *IEEE Transactions on Fuzzy Systems*, vol. 26, no 3, pp. 1535–1549.
4. Lucca G., Dimuro G.P., Fernández J., Bustince H., Bedregal B.R.C., Sanz J.A. (2019) Improving the performance of fuzzy rule-based classification systems based on a nonaveraging generalization of CC-integrals named C_{F1F2} -integrals. *IEEE Transactions on Fuzzy Systems*, vol. 27, no 1, pp. 124–134.
5. Zhang P., Gao W., Liu G. (2018) Feature selection considering weighted relevancy. *Applied Intelligence*, no 48, pp. 4615–4625.
6. Bolon-Canedo V., Sanchez-Marono N., Alonso-Betanzos A. (2015) *Feature Selection for High-Dimensional Data*. London: Springer.
7. Cai J., Luo J., Wang S., Yang S. (2018) Feature selection in machine learning: A new perspective. *Neurocomputing*, no 300, pp. 70–79.
8. Pashaei E., Aydin N. (2017) Binary black hole algorithm for feature selection and classification on biological data. *Applied Soft Computing*, no 56, pp. 94–106.
9. Yusta S.C. (2009) Different metaheuristic strategies to solve the feature selection problem. *Pattern Recognition Letters*, no 30, pp. 525–534.
10. Kohavi R., John G.H. (1997) Wrappers for feature subset selection. *Artificial Intelligence*, no 97, pp. 273–324.
11. Dong H., Li T., Ding R., Sun J. (2018) A novel hybrid genetic algorithm with granular information for feature selection and optimization. *Applied Soft Computing*, no 65, pp. 33–46.
12. Sayed S., Nassef M., Badr A., Farag I. (2019) A nested genetic algorithm for feature selection in high-dimensional cancer microarray datasets. *Expert Systems with Applications*, no 121, pp. 233–243.
13. Ma B., Xia Y. (2017) A tribe competition-based genetic algorithm for feature selection in pattern classification. *Applied Soft Computing*, no 58, pp. 328–338.
14. Wu Y.-L., Tang C.-Y., Hor M.-K., Wu P.-F. (2011) Feature selection using genetic algorithm and cluster validation. *Expert Systems with Applications*, no 38, pp. 2727–2732.
15. Sikora R., Piramuthu S. (2007) Framework for efficient feature selection in genetic algorithm based data mining. *European Journal of Operational Research*, no 180, pp. 723–737.
16. Vieira S.M., Sousa J.M.C., Runkler T.A. (2007) Ant colony optimization applied to feature selection in fuzzy classifiers. *IFSA 2007, LNAI 4529*. Berlin: Springer, pp. 778–788.
17. Ghimatgar H., Kazemi K., Helfroush M.S., Aarabi A. (2018) An improved feature selection algorithm based on graph clustering and ant colony optimization. *Knowledge-Based Systems*, no 159, pp. 270–285.
18. Dadaneh B.Z., Markid H.Y., Zakerolhosseini A. (2016) Unsupervised probabilistic feature selection using ant colony optimization. *Expert Systems with Applications*, no 53, pp. 27–42.
19. Diao R., Shen Q. (2012) Feature selection with harmony search. *IEEE Transactions on Systems, Man, and Cybernetics – PART B: Cybernetics*, no 42, pp. 1509–1523.
20. Hodashinsky I.A., Mekh M.A. (2017) Fuzzy classifier design using harmonic search methods. *Programming and Computer Software*, no 43, pp. 37–46.

21. Mirjalili S., Lewis A. (2013) S-shaped versus V-shaped transfer functions for binary particle swarm optimization. *Swarm and Evolutionary Computation*, no 9, pp. 1–14.
22. Yadav S., Ekbal A., Saha S. (2018) Feature selection for entity extraction from multiple biomedical corpora: A PSO-based approach. *Soft Computing*, no 22, pp. 6881–6904.
23. Ajit Krishna N.L., Deepak V.K., Manikantan K., Ramachandran S. (2014) Face recognition using transform domain feature extraction and PSO-based feature selection. *Applied Soft Computing*, no 22, pp. 141–161.
24. Bardamova M., Konev A., Hodashinsky I., Shelupanov A. (2018) A fuzzy classifier with feature selection based on the gravitational search algorithm. *Symmetry*, no 10 (11), 609.
25. Rashedi E., Nezamabadi-pour H. (2014) Feature subset selection using improved binary gravitational search algorithm. *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology*, no 26, pp. 1211–1221.
26. Papa J.P., Rosa G.H., de Souza A.N., Afonso L.C.S. (2018) Feature selection through binary brain storm optimization. *Computers and Electrical Engineering*, no 72, pp. 468–481.
27. Kiziloz H.E., Deniz A., Dokeroglu T., Cosar A. (2018) Novel multiobjective TLBO algorithms for the feature subset selection problem. *Neurocomputing*, no 306, pp. 94–107.
28. Mafarja M., Aljarah I., Faris H., Hammouri A.I., Ala'M A.Z., Mirjalili S. (2019) Binary grasshopper optimisation algorithm approaches for feature selection problems. *Expert Systems with Applications*, no 117, pp. 267–286.
29. Sayed G.I., Khoriba G., Haggag M.H. (2018) A novel chaotic salp swarm algorithm for global optimization and feature selection. *Applied Intelligence*, no 48, pp. 3462–3481.
30. Faris H., Mafarja M.M., Heidari A.A., Aljarah I., Al-Zoubi A.M., Mirjalili S., Fujita H. (2018) An efficient binary salp swarm algorithm with crossover scheme for feature selection problems. *Knowledge-Based Systems*, no 154, pp. 43–67.
31. Emary E., Zawbaa H.M., Hassanien A.E. (2016) Binary ant lion approaches for feature selection. *Neurocomputing*, no 213, pp. 54–65.
32. Bansal J.C., Sharma H., Jadon S.S., Clerc M. (2014) Spider monkey optimization algorithm for numerical optimization. *Memetic Computing*, no 6, pp. 31–47.
33. Agrawal V., Rastogi R., Tiwari D.C. (2018) Spider monkey optimization: a survey. *International Journal of System Assurance Engineering and Management*, no 9, pp. 929–941.
34. Wolpert D.H. (1996) The existence of a priori distinctions between learning algorithms. *Neural Computation*, no 8, pp. 1341–1390.
35. Wolpert D.H. (1996) The lack of a priori distinctions between learning algorithms. *Neural Computation*, no 8, pp. 1391–1420.
36. Singh U., Salgotra R., Rattan M. (2016) A novel binary spider monkey optimization algorithm for thinning of concentric circular antenna arrays. *IETE Journal of Research*, no 62, pp. 736–744.
37. Hodashinsky I.A., Samsonov S.S. (2017) Design of fuzzy rule based classifier using the monkey algorithm. *Business Informatics*, no 1, pp. 61–67.
38. Fazzolari F., Alcalá R., Herrera 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*, no 24, pp. 470–481.

About the authors

Ilya A. Hodashinsky

Dr. Sci. (Tech.), Professor;

Professor, Department of Complex Information Security of Computer Systems,
Tomsk State University of Control Systems and Radioelectronics (TUSUR), 40, Prospect Lenina, Tomsk 634050, Russia;
E-mail: hodashn@rambler.ru

Mikhail M. Nemirovich-Danchenko

Dr. Sci. (Phys.-Math.);

Professor, Department of Complex Information Security of Computer Systems,
Tomsk State University of Control Systems and Radioelectronics (TUSUR), 40, Prospect Lenina, Tomsk 634050, Russia;
E-mail: nemdan2011@gmail.com

Sergey S. Samsonov

Doctoral Student, Department of Complex Information Security of Computer Systems,
Tomsk State University of Control Systems and Radioelectronics (TUSUR), 40, Prospect Lenina, Tomsk 634050, Russia;
E-mail: samsonicx@mail.ru