

## ПРИМЕНЕНИЕ РАНЖИРОВАНИЯ И СХЕМ КРОССВАЛИДАЦИИ ПРИ ОТБОРЕ ПРИЗНАКОВ ДЛЯ НЕЧЕТКОГО КЛАССИФИКАТОРА

**Ходашинский Илья Александрович**

Д.т.н., профессор, ТУСУР, e-mail: hodashn@rambler.ru

**Анфилофьев Александр Евгеньевич**

Аспирант, ТУСУР, e-mail: yowwi00@gmail.com

**Бардамова Марина Борисовна**

Аспирантка, ТУСУР, e-mail: 722bmb@gmail.com

**Самсонов Сергей Сергеевич**

Студент, ТУСУР, e-mail: samsonicx@mail.ru

**Филимоненко Игорь Витальевич**

Студент, ТУСУР, e-mail: ifilimon96@mail.ru

Томский государственный университет систем управления и радиоэлектроники  
(ТУСУР), 634050 г. Томск, пр. Ленина 40

**Аннотация.** Отбор признаков является NP-трудной задачей, гарантировано оптимальное решение может быть найдено только путем полного перебора. В статье описывается подход к отбору признаков на основе ранжирования и схем кроссвалидации. Для формирования оптимальных наборов признаков используются бинарные метаэвристические алгоритмы: гравитационный поиск, сорняковый алгоритм, алгоритм обезьян и алгоритм крилей.

**Ключевые слова:** кроссвалидация, отбор признаков, классификаторы, бинарные метаэвристики

**Цитирование:** Ходашинский И.А., Анфилофьев А.Е., Бардамова М.Б., Самсонов С.С., Филимоненко И.В. Применение ранжирования и схем кроссвалидации при отборе признаков для нечеткого классификатора // Информационные и математические технологии в науке и управлении. 2018. №2 (10). С. 31–41. DOI:10.25729/2413-0133-2018-2-03

**Введение.** Отбор релевантных признаков – важный этап при построении систем интеллектуального анализа данных. Отбор признаков, как метод предварительной обработки данных, может способствовать не только повышению эффективности алгоритмов обучения, но и повышению прогностической способности и интерпретируемости полученного результата.

Увеличение количества признаков приводит к ухудшению эффективности работы алгоритмов обучения, вопреки интуитивному представлению о том, что большее количество признаков позволит получить больше информации и, тем самым, с большей точностью выполнить задачу классификации. Причина в том, что по мере увеличения количества признаков алгоритмы нуждаются в большем количестве данных для обучения, необходимых для построения правил, определяющих соотношение между этими признаками и меткой класса. Кроме того, признаки, не содержащие информации о метке класса, могут способствовать неправильной классификации и замедлять процесс обучения. Поэтому

разработка методов отбора подмножества признаков, которые классифицируют данные более точно, чем полный набор признаков, является актуальной задачей [4].

Методы отбора признаков можно разделить на две группы: фильтры и обертки. Метод фильтров основан на обобщенных свойствах обучающих данных и не включает в процесс отбора признаков собственно алгоритм построения классификатора. Метод обертки включает построение классификатора в процесс отбора признаков и использует прогностическую точность классификатора для оценки эффективности отобранного подмножества признаков.

Цель нашей статьи – описать методику и бинарные алгоритмы отбора признаков по методу обертки с использованием схем кроссвалидации.

**1. Нечеткий классификатор.** Основой нечеткого классификатора является продукционное правило следующего вида:

$$R_{ij} : \text{ЕСЛИ } s_1 \wedge x_1 = A_{1i} \text{ И } s_2 \wedge x_2 = A_{2i} \text{ И } \dots \text{ И } s_n \wedge x_n = A_{ni} \text{ ТО class} = c_j,$$

где  $A_{ki}$  – нечеткий терм, характеризующий  $k$ -ый признак в  $i$ -ом правиле ( $i \in [1, R]$ ); запись  $s_i \wedge x_i$  указывает на наличие ( $s_i = 1$ ) или отсутствие ( $s_i = 0$ ) признака в классификаторе;  $R$  – число правил.

В нашей работе класс определяется следующим образом:

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

где  $\beta_j(\mathbf{x}) = \sum_{R_{ij}} \prod_{k=1}^n \mu_{A_{ki}}(x_k)$ ,  $j = \overline{1, m}$ ,  $\mu_{A_{ki}}(x_k)$  – значение функции принадлежности нечеткого термина  $A_{ki}$  в точке  $x_k$ .

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

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

где  $f(\mathbf{x}_p; \boldsymbol{\theta}, \mathbf{S})$  – выход, определённый нечетким классификатором с параметрами  $\boldsymbol{\theta}$  и признаками  $\mathbf{S}$  в точке  $\mathbf{x}_p$ .

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; решение представляется в виде вектора  $\mathbf{S} = (s_1, s_2, \dots, s_n)^T$ , 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 классификатора, структура которого сформирована на основе экстремумов значений признаков в таблице наблюдений [1].

## 2. Бинарные алгоритмы

**2.1. Бинарный гравитационный алгоритм** основан на непрерывном гравитационном алгоритме [6]. В бинарном алгоритме каждый оптимизируемый вектор  $\mathbf{S}$  представляется в виде частицы, масса которой определяется точностью классификатора, построенного на соответствующем векторе. Частицы, являющиеся лучшими решениями с точки зрения фитнес-функции, обладают наибольшей массой и притягивают к себе более

мелкие частицы. Так как силы притяжения действуют на все частицы, то двигаются и тяжелые частицы, тем самым осуществляя поиск в близлежащем пространстве. Таким образом, в качестве массы  $i$ -ой частицы используется значение ошибки классификации  $Err(\theta, \mathbf{S}) = 1 - E(\theta, \mathbf{S})$ , расстояние между двумя частицами рассматривается как евклидово расстояние между векторами. Далее для каждой частицы подсчитывается ускорение по законам Ньютона и скорость как сумма ускорения и случайной компоненты от скорости, вычисленной на предыдущей итерации [2].

Далее численное значение скорости частицы должно быть переведено в бинарный эквивалент для перемещения частицы. В данной работе используется функция трансформации, которая определяет вероятность изменения значения элемента вектора на противоположное:

$$\begin{cases} \text{ЕСЛИ } (rand(0;1) < F(V_i^d(t+1))), \text{ ТО } p = 1 \\ \text{ИНАЧЕ } p = 0 \\ S_i^d(t+1) = S_i^d(t) \oplus p \end{cases},$$

где  $F(V_i^d(t+1)) = \left| \frac{2}{\pi} \arctan\left(\frac{\pi}{2} V_i^d(t+1)\right) \right|$ ,  $t$  – текущая итерация,  $i$  – номер частицы,  $i = \overline{1; P}$ ,  $P$  – количество векторов,  $d$  – номер элемента вектора,  $d = \overline{1; |P_i|}$ ,  $\oplus$  – логическое исключающее «ИЛИ».

Начальная популяция векторов  $\mathbf{S}$  для дискретного алгоритма генерируется случайным образом; размерность вектора равняется количеству признаков.

**2.2. Непрерывный сорняковый алгоритм** является популяционным алгоритмом, отражающим ограниченный по времени жизненный цикл сорняков при их распространении и выживании на ограниченной территории [5]. Бинарный сорняковый алгоритм основан на применении функции Гаусса с нулевым средним и дисперсией  $\sigma^2$ . Значение функции определяется с помощью преобразования Бокса-Мюллера:  $\cos(b)\sqrt{-2\ln(a)}$ , где  $a$  и  $b$  независимые случайные величины, равномерно распределенные в диапазоне  $(0,1]$ . Для перехода к бинарным значениям вводится пороговое значение  $thr \in (0,1)$  и вычисляется  $w = \cos(b)\sqrt{-2\ln(a)}$ , если  $|w| < thr$ , то  $s_i^*(t) = \text{NOT}(s_i(t))$  иначе  $s_i^*(t) = s_i(t)$ . Бинарный сорняковый алгоритм учитывает время работы алгоритма:  $thr_{iter} := thr \cdot \left(\frac{N - iter}{N}\right)$ , чем

дольше работает алгоритм, тем медленнее происходят изменения в векторе признаков  $\mathbf{S}$ . Изменение порога приводит к тому, что на ранних этапах работы алгоритма каждый дочерний вектор признаков будет сильно отличаться от вектора, из которого он был порождён (родительского); этим обеспечивается реализация глобального поиска на ранних итерациях. При достижении конечных этапов работы алгоритма каждый элемент дочернего вектора будет с большей долей вероятности соответствовать элементу порождающего элемента родительского вектора, что позволяет произвести тонкую настройку и определить минимум в окрестности родительского решения.

**2.3. Непрерывный алгоритм обезьян** основан на наблюдениях за передвижением обезьян в горной местности [8]. Бинарный алгоритм обезьян начинает свою работу с инициализации популяции бинарных решений. В процессе работы алгоритма популяция

делится на группы, в каждой из которых присутствует свое лучшее решение, определяемое по наибольшему значению  $E(\boldsymbol{\theta}, \mathbf{S}_i)$ , кроме того находится глобально лучшее решение. Новое значение  $s_{ij}$  в векторе признаков  $\mathbf{S}$  в соответствии с локальным решением определяется следующим образом:

$$s_{i,j_{new}} = \begin{cases} s_{i,j} \oplus \left( \begin{array}{l} (b \otimes (LL_{k,j} \oplus s_{i,j})) \\ + (d \otimes (s_{r,j} \oplus s_{i,j})) \end{array} \right), rand > pr, \\ s_{i,j}, иначе \end{cases},$$

где  $b, d$  принимают случайные значения из множества  $\{0;1\}$ ;  $LL_k$  – лучшее решение в группе.

Новое значение  $s_{ij}$  в векторе признаков  $\mathbf{S}$  в соответствии с глобальным решением определяется следующим образом:

$$s_{i,j_{new}} = \begin{cases} s_{i,j} \oplus \left( \begin{array}{l} (b \otimes (GL_j \oplus s_{i,j})) \\ + (d \otimes (S_{r,j} \oplus s_{i,j})) \end{array} \right), rand > P_i, \\ s_{i,j}, иначе \end{cases},$$

где  $GL$  – вектор, определяющий координаты глобального решения,

$$P_i = 0.9 \cdot \frac{E(\boldsymbol{\theta}, \mathbf{S}_i)}{\max E(\boldsymbol{\theta}, \mathbf{S})} + 0.1.$$

**2.4. Непрерывный алгоритм крилей** имитирует перемещение антарктических крилей в процессе поиска пищи с учетом взаимосвязанных факторов. Одним из факторов является наличие хищника, которое заставляет крилей держаться вместе и увеличивает плотность стада ( $N$ ); другой фактор – поиск пищи ( $F$ ); третий фактор – случайное блуждание криля ( $RD$ ) [3]. В бинарном алгоритме перемещение криля  $\mathbf{N}_i(t)$ , вызванное движением других членов стада, вычисляется следующим образом:

$$\mathbf{N}_i(t+1) = \text{disct}(\text{disct}(\mathbf{N}_{\max}, \boldsymbol{\alpha}_i), \mathbf{N}_i(t)), \quad \boldsymbol{\alpha}_i = \text{disct}(\boldsymbol{\alpha}_i^{local}, \boldsymbol{\alpha}_i^{target}),$$

где  $N_{\max}$  – максимальная скорость крилей в стаде;  $\boldsymbol{\alpha}_i^{local}$  – вектор движения, учитывающий влияние соседей  $i$ -го криля;  $\boldsymbol{\alpha}_i^{target}$  – вектор движения, учитывающий влияние лучшего криля в стаде. Влияние соседей определяется следующим образом:

$$\boldsymbol{\alpha}_i^{local} = \text{disct}(\hat{K}_{i,j}, \hat{\boldsymbol{\theta}}_{i,j}), \quad \hat{\boldsymbol{\theta}}_{i,j} = \text{disct}(\boldsymbol{\theta}_j, \boldsymbol{\theta}_i), \quad \hat{K}_{i,j} = \text{disct}(\text{randvector}(0,1), \boldsymbol{\theta}_{i,j}),$$

$$\boldsymbol{\alpha}_i^{target} = \text{disct}(\text{disct}(\text{randvector}(0,1), \hat{K}_{i,best}), \hat{\boldsymbol{\theta}}_{i,best}),$$

$$\mathbf{F}_i(t+1) = \text{disct}(\text{disct}(1, \boldsymbol{\beta}_i), \text{disct}(0, \mathbf{F}_i(t))), \quad \boldsymbol{\beta}_i = \text{disct}(\boldsymbol{\beta}_i^{food}, \boldsymbol{\beta}_i^{best}),$$

$$\boldsymbol{\beta}_i^{food} = \text{disct}(\text{disct}(C_{food}, \hat{K}_{i,food}), \hat{\boldsymbol{\theta}}_{i,food}), \quad C_{food} = \begin{cases} 1, \text{если } 2(1 - (\frac{t}{t_{\max}})) > 1 \\ 0, \text{иначе} \end{cases},$$

$$\boldsymbol{\theta}_{food} = \text{disct}(\boldsymbol{\theta}_i, \text{randvector}(0,1)), \quad \boldsymbol{\beta}_i^{best} = \text{disct}(\hat{K}_{i,best}, \hat{\boldsymbol{\theta}}_{i,best}),$$

$$\text{randvector}(0,1) = (bin_1, bin_2, \dots, bin_n), \quad bin_i = \text{rand}\{0,1\},$$

$$\text{disct}(\mathbf{x}, \mathbf{y}) = (res_1, res_2, \dots, res_n), \quad res_i = \begin{cases} x_i, \text{если } x_i = y_i \\ \text{rand}\{0,1\}, \text{иначе} \end{cases}.$$

Позиция  $i$ -го криля на  $(t+1)$ -ой итерации выполнения алгоритма определяется следующим образом:

$$\theta_i(t+1) = \text{disct}(\theta_i(t), \Delta\theta_i), \Delta\theta_i = \text{disct}(\text{disct}(\mathbf{N}_i(t), \mathbf{F}_i(t)), 1).$$

**3. Методика отбора признаков** с использованием схем кроссвалидации состоит из следующих этапов:

- 1) классифицируемый набор данных разбивается на  $k$  выборок в соответствии со схемой кросс-валидации;
- 2) на всех выборках по  $n$  раз запускается бинарный метаэвристический алгоритм; в результате каждого из  $k \cdot n$  запусков выявлялся набор признаков с наиболее высокой точностью на обучающей выборке;
- 3) полученные признаки ранжируются по частоте встречаемости в наборах;
- 4) на основе полученных рангов составляются наборы признаков и осуществляется построение классификаторов.

Для подтверждения работоспособности алгоритмов проведены эксперименты на наборах данных, приведенных в таблице 1.

**Таблица 1.** Наборы данных

Набор данных	Число признаков	Число образцов	Число классов
Vowel	13	990	11
Penbased	16	10992	10

**4. Эксперимент** проводился по следующей схеме:

Шаг 1. Отбор информативных признаков с использованием бинарных алгоритмов.

Шаг 2. Генерация на отобранных признаках правил нечетких классификаторов [1].

Для оценки статистической значимости различий в рангах отобранных классифицирующих признаков, сформированных алгоритмом гравитационного поиска (АГП), сорняковым алгоритмом (СА), алгоритмом обезьян (АО) и алгоритмом крилей (АК), использованы критерий знаковых рангов Уилкоксона и двухфакторный ранговый дисперсионный анализ Фридмана для связанных выборок.

Для первого критерия была сформулирована нулевая гипотеза: на уровне значимости 0.05 медиана разности между парой ранжировок равна нулю.

Для двухфакторного рангового дисперсионного анализа Фридмана для связанных выборок критерия была сформулирована нулевая гипотеза: на уровне значимости 0.05 распределения рангов, полученных четырьмя алгоритмами, статистически неразличимы.

В таблице 2 приведены частоты встречаемости признаков и ранги, присвоенные признакам набора данных Vowel алгоритмом гравитационного поиска, сорняковым алгоритмом, алгоритмом обезьян и алгоритмом крилей.

В таблице 3 приведены результаты проверки нулевой гипотезы критерием знаковых рангов Уилкоксона для набора данных Vowel, здесь в ячейке таблицы указана значимость.

Поскольку значимость для этого теста для всех пар больше 0.05, нулевая гипотеза на уровне значимости 95.0% не может быть отклонена. Из чего следует: для набора данных Vowel ранжировки, выполненные четырьмя алгоритмами, попарно статистически неразличимы на заданном уровне значимости.

Для двухфакторного рангового дисперсионного анализа Фридмана для связанных выборок получена значимость, равная 0.947, из чего можно сделать вывод: для набора

данных Vowel ранжировки, выполненные четырьмя алгоритмами, статистически неразличимы на заданном уровне значимости.

**Таблица 2.** Частота встречаемости признаков и их ранги на данных Vowel

№ признака	АГП		СА		АО		АК	
	Частота	Ранг	Частота	Ранг	Частота	Ранг	Частота	Ранг
1	0	11,5	0	11	4	8	0	12
2	12	6	5	7	8	7	2	9
3	5	8	0	11	2	10	4	6
4	50	3	50	3	50	3	10	1,5
5	50	3	50	3	50	3	10	1,5
6	0	11,5	0	11	1	11	1	10
7	50	3	50	3	50	3	9	3,5
8	50	3	50	3	50	3	9	3,5
9	10	7	9	6	12	6	0	12
10	1	9	0	11	0	12,5	3	7,5
11	0	11,5	0	11	0	12,5	3	7,5
12	50	3	50	3	50	3	8	5
13	0	11,5	1	8	3	9	0	12

**Таблица 3.** Набор данных Vowel: итоги проверки нулевой гипотезы критерием знаковых рангов Уилкоксона

Алгоритм	АГП	СА	АО	АК
АГП	1.000	0.944	0.944	0.888
СА	0.944	1.000	0.785	1.000
АО	0.944	0.785	1.000	0.944
АК	0.888	1.000	0.944	1.000

В таблице 4 приведены частоты встречаемости признаков и ранги, присвоенные признакам набора данных Penbased алгоритмом гравитационного поиска, сорняковым алгоритмом, алгоритмом обезьян и алгоритмом крилей.

**Таблица 4.** Частота встречаемости признаков и их ранги на данных Penbased

№ признака	АГП		СА		АО		АК	
	Частота	Ранг	Частота	Ранг	Частота	Ранг	Частота	Ранг
1	16	9	20	9	17	9	0	13
2	13	10	14	10	12	11	0	13
3	10	11	0	15,5	11	12	0	13
4	0	16	0	15,5	0	16	0	13
5	47	4	48	4	47	4	8	6
6	6	13	6	12	5	13,5	0	13
7	35	6	25	8	34	6	7	7
8	25	8	29	7	31	7	9	4
9	39	5	38	5	35	5	1	9
10	28	7	31	6	29	8	9	4
11	8	12	10	11	15	10	5	8
12	50	1,5	50	2	50	1,5	10	1,5
13	2	15	2	14	5	13,5	0	13
14	49	3	50	2	49	3	10	1,5
15	4	14	4	13	2	15	0	13
16	50	1,5	50	2	50	1,5	9	4

В таблице 5 приведены результаты проверки нулевой гипотезы критерием знаковых рангов Уилкоксона для набора данных Penbased, здесь в ячейке таблицы указана значимость.

Поскольку значимость для этого теста для всех пар больше 0.05, нулевая гипотеза на уровне значимости 95.0% не может быть отклонена. Из чего следует: для набора данных Penbased ранжировки, выполненные четырьмя алгоритмами, попарно статистически неразличимы на заданном уровне значимости.

Для двухфакторного рангового дисперсионного анализа Фридмана для связанных выборок получена значимость, равная 0.917, из чего можно сделать вывод: для набора данных Penbased ранжировки, выполненные четырьмя алгоритмами, статистически неразличимы на заданном уровне значимости.

**Таблица 5.** Набор данных Penbased: итоги проверки нулевой гипотезы критерием знаковых рангов Уилкоксона

Алгоритм	АГП	СА	АО	АК
АГП	1.000	0.335	0.886	1.000
СА	0.335	1.000	0.782	0.955
АО	0.886	0.782	1.000	1.000
АК	1.000	0.955	1.000	1.000

В таблицах 6 и 7 приведены усредненные значения обучающей ( $E_{TR}$ ) и тестовой ( $E_{TST}$ ) точности нечетких классификаторов, построенных на наборах данных Vowel и Penbased соответственно. Первый классификатор обучается на наборе признаков, имеющих наименьший ранг, следующие наборы составляются путем поочередного добавления признаков по возрастанию рангов. Количество признаков в наборе указано в столбце  $F$ .

**Заключение.** Для решения задач отбора признаков для нечетких классификаторов предложено использовать методы ранжирования и схем кроссвалидации. Отбор признаков ведется по методу обертки. Для формирования оптимальных наборов признаков используются бинарные метаэвристические алгоритмы: гравитационный поиск, сорняковый алгоритм, алгоритм обезьян и алгоритм крилей. Сравнительный статистический анализ не выявил алгоритма, превосходящего остальные по точности классификации без оптимизации параметров нечеткого классификатора. Опираясь на «теорему о бесплатных завтраках» [7], можно сделать вывод: для решения оптимизационных задач отбора признаков должны использоваться различные бинарные метаэвристические алгоритмы.

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 16-07-00034а.

**Таблица 6.** Набор данных Vowel: результаты построения нечетких классификаторов на отобранных бинарными алгоритмами наборах признаков

Классификаторы, построенные на признаках, отобранных АГП						Классификаторы, построенные на признаках, отобранных СА					
№	Ранги	Признаки	<i>F</i>	<i>E<sub>TR</sub></i>	<i>E<sub>TST</sub></i>	№	Ранги	Признаки	<i>F</i>	<i>E<sub>TR</sub></i>	<i>E<sub>TST</sub></i>
1	3	<b>4, 5, 7, 8, 12</b>	5	47,2	47,9	1	3	<b>4, 5, 7, 8, 12</b>	5	47,2	47,9
2	3, 6	<b>2, 4, 5, 7, 8, 12</b>	6	47,0	47,6	2	3, 6	<b>4, 5, 7, 8, 9, 12</b>	6	46,8	46,9
3	3, 6, 7	<b>2, 4, 5, 7, 8, 9, 12</b>	7	46,9	46,7	3	3, 6, 7	<b>2, 4, 5, 7, 8, 9, 12</b>	7	46,9	46,7
4	3, 6, 7, 8	<b>2, 3, 4, 5, 7, 8, 9, 12</b>	8	46,6	46,1	4	3, 6, 7, 8	<b>2, 4, 5, 7, 8, 9, 12, 13</b>	8	45,8	45,4
5	3, 6, 7, 8, 9	<b>2, 3, 4, 5, 7, 8, 9, 10, 12</b>	9	44,5	44,0	5	3, 6, 7, 8, 11	Полный набор	13	39,4	38,3
6	3, 6, 7, 8, 9, 11.5	Полный набор	13	39,4	38,3						
Классификаторы, построенные на признаках, отобранных АО						Классификаторы, построенные на признаках, отобранных АК					
№	Ранги	Признаки	<i>F</i>	<i>E<sub>TR</sub></i>	<i>E<sub>TST</sub></i>	№	Ранги	Признаки	<i>F</i>	<i>E<sub>TR</sub></i>	<i>E<sub>TST</sub></i>
1	3	<b>4, 5, 7, 8, 12</b>	5	47,2	47,9	1	1.5	<b>4, 5</b>	2	35,8	36,0
2	3, 6	<b>4, 5, 7, 8, 9, 12</b>	6	46,8	46,6	2	1.5, 3.5	<b>4, 5, 7, 8</b>	4	42,1	42,3
3	3, 6, 7	<b>2, 4, 5, 7, 8, 9, 12</b>	7	46,9	46,7	3	1.5, 3.5, 5	<b>4, 5, 7, 8, 12</b>	5	47,2	47,9
4	3, 6, 7, 8	<b>1, 2, 4, 5, 7, 8, 9, 12</b>	8	46,3	46,0	4	1.5, 3.5, 5, 6	<b>3, 4, 5, 7, 8, 12</b>	6	47,0	47,6
5	3, 6, 7, 8, 9	<b>1, 2, 4, 5, 7, 8, 9, 12, 13</b>	9	44,9	44,4	5	1.5, 3.5, 5, 6, 7.5	<b>3, 4, 5, 7, 8, 10, 11, 12</b>	8	42,4	42,8
6	3, 6, 7, 8, 9, 10	<b>1, 2, 3, 4, 5, 7, 8, 9, 12, 13</b>	10	44,5	44,0	6	1.5, 3.5, 5, 6, 7.5, 9	<b>2, 3, 4, 5, 7, 8, 10, 11, 12</b>	9	42,2	42,7
7	3, 6, 7, 8, 9, 10, 11	<b>1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 13</b>	11	41,3	41,3	7	1.5, 3.5, 5, 6, 7.5, 9, 10	<b>2, 3, 4, 5, 6, 7, 8, 10, 11, 12</b>	10	41,5	41,4
8	3, 6, 7, 8, 9, 10, 11, 12.5	Полный набор	13	39,4	38,3	8	1.5, 3.5, 5, 6, 7.5, 9, 10, 12	Полный набор	13	39,4	38,3



**Таблица 7.** Набор данных Penbased: результаты построения нечетких классификаторов на отобранных бинарными алгоритмами наборах признаков

Классификаторы, построенные на признаках, отобранных АГП					Классификаторы, построенные на признаках, отобранных СА				
№	Признаки	$F$	$E_{TR}$	$E_{TST}$	№	Признаки	$F$	$E_{TR}$	$E_{TST}$
1	<b>12, 16</b>	2	35,5	35,3	1	<b>12, 14, 16</b>	3	36,7	36,5
2	12, <b>14</b> , 16	3	36,7	36,5	2	<b>5</b> , 12, 14, 16	4	41,6	41,6
3	<b>5</b> , 12, 14, 16	4	41,6	41,6	3	5, <b>9</b> , 12, 14, 16	5	47,4	47,5
4	5, <b>9</b> , 12, 14, 16	5	47,4	47,5	4	5, 9, <b>10</b> , 12, 14, 16	6	47,3	47,4
5	5, <b>7</b> , 9, 12, 14, 16	6	50,5	50,4	5	5, <b>8</b> , 9, 10, 12, 14, 16	7	47,3	47,3
6	5, 7, 9, <b>10</b> , 12, 14, 16	7	49,5	49,6	6	5, <b>7</b> , 8, 9, 10, 12, 14, 16	8	48,9	48,8
7	5, 7, <b>8</b> , 9, 10, 12, 14, 16	8	48,9	48,8	7	<b>1</b> , 5, 7, 8, 9, 10, 12, 14, 16	9	50,2	50,0
8	<b>1</b> , 5, 7, 8, 9, 10, 12, 14, 16	9	50,2	50,0	8	1, <b>2</b> , 5, 7, 8, 9, 10, 12, 14, 16	10	48,7	48,6
9	1, <b>2</b> , 5, 7, 8, 9, 10, 12, 14, 16	10	48,7	48,6	9	1, 2, 5, 7, 8, 9, 10, <b>11</b> , 12, 14, 16	11	44,3	44,4
10	1, 2, <b>3</b> , 5, 7, 8, 9, 10, 12, 14, 16	11	47,6	47,4	10	1, 2, 5, <b>6</b> , 7, 8, 9, 10, 11, 12, 14, 16	12	41,9	42,1
11	1, 2, 3, 5, 7, 8, 9, 10, <b>11</b> , 12, 14, 16	12	43,5	43,7	11	1, 2, 5, 6, 7, 8, 9, 10, 11, 12, 14, <b>15</b> , 16	13	38,6	38,7
12	1, 2, 3, 5, <b>6</b> , 7, 8, 9, 10, 11, 12, 14, 16	13	41,6	41,7	12	1, 2, 5, 6, 7, 8, 9, 10, 11, 12, <b>13</b> , 14, 15, 16	14	36,9	37,1
13	1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 14, <b>15</b> , 16	14	38,4	38,7	13	Полный набор	16	31,9	31,9
14	1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, <b>13</b> , 14, 15, 16	15	36,5	36,8					
15	Полный набор	16	31,9	31,9					
Классификаторы, построенные на признаках, отобранных АО					Классификаторы, построенные на признаках, отобранных АК				
№	Признаки	$F$	$E_{TR}$	$E_{TST}$	№	Признаки	$F$	$E_{TR}$	$E_{TST}$
1	<b>12, 16</b>	2	35,5	35,3	1	<b>12, 14</b>	2	35,4	35,4
2	12, <b>14</b> , 16	3	36,7	36,5	2	<b>8, 10</b> , 12, 14, 16	5	41,1	41,1
3	<b>5</b> , 12, 14, 16	4	41,6	41,6	3	<b>5</b> , 8, 10, 12, 14, 16	6	46,6	46,5
4	5, <b>9</b> , 12, 14, 16	5	47,4	47,5	4	5, <b>7</b> , 8, 10, 12, 14, 16	7	48,4	48,4
5	5, <b>7</b> , 9, 12, 14, 16	6	50,5	50,4	5	5, 7, 8, 10, <b>11</b> , 12, 14, 16	8	50,2	50,1
6	5, 7, <b>8</b> , 9, 12, 14, 16	7	49,4	49,2	6	5, 7, 8, <b>9</b> , 10, 11, 12, 14, 16	9	49,9	49,9
7	5, 7, 8, 9, <b>10</b> , 12, 14, 16	8	48,9	48,8	7	Полный набор	16	31,9	31,9
8	<b>1</b> , 5, 7, 8, 9, 10, 12, 14, 16	9	50,2	50,0					
9	1, 5, 7, 8, 9, 10, <b>11</b> , 12, 14, 16	10	49,9	49,8					
10	1, <b>2</b> , 5, 7, 8, 9, 10, 11, 12, 14, 16	11	44,3	44,4					
11	1, 2, <b>3</b> , 5, 7, 8, 9, 10, 11, 12, 14, 16	12	43,5	43,7					
12	1, 2, 3, 5, <b>6</b> , 7, 8, 9, 10, 11, 12, <b>13</b> , 14, 16	14	39,5	39,6					
13	1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, <b>15</b> , 16	15	36,5	36,8					
14	Полный набор	16	31,9	31,9					

#### СПИСОК ЛИТЕРАТУРЫ

1. 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.
  2. Ходашинский И.А., Бардамова М.Б. Бинаризация алгоритма гравитационного поиска в задачах отбора признаков для нечетких классификаторов // XIII Международная научно-практическая конференция «Электронные средства и системы управления» (Томск, ТУСУР, 29 ноября – 1 декабря 2017 г.). Томск. В-Спектр. 2017. Ч. 2. С. 29–31.
  3. Gandomi A.H., Alavi A.H. Krill herd: A new bio-inspired optimization algorithm // Communications in Nonlinear Science and Numerical Simulation. 2012. Vol. 17. Pp. 4831–4845.
  4. Jalalirad A., Tjalkens T. Using feature-based models with complexity penalization for selecting features // Journal of Signal Processing Systems. 2018. Vol. 90. Pp. 201–210.
  5. Mehrabian A.R, Lucas C. A novel numerical optimization algorithm inspired from weed colonization // Ecological informatics. 2006. Vol.1. Pp. 355–366.
  6. Rashedi E., Nezamabadi-pour H., Saryazdi S. GSA: a gravitational search algorithm // Information Sciences. 2009. Vol.179. Pp. 2232–2248.
  7. Wolpert D.H., Macready W.G. No Free Lunch Theorems for Optimization // IEEE Transactions on Evolutionary Computation. 1997. Vol.1. Pp. 67–82.
  8. Zhao R., Tang W. Monkey algorithm for global numerical optimization // Journal of Uncertain Systems. 2008. Vol.2. Pp. 165–176.
- 

**UDK 004.8**

#### **FEATURE SELECTION FOR FUZZY CLASSIFIERS USING THE RANKING AND CROSS-VALIDATION**

**Ilya A. Hodashinsky**

Dr., Professor, e-mail: hodashn@rambler.ru

**Alexander E. Anfilofiev**

Graduate student, e-mail: yowwi00@gmail.com

**Marina B. Bardamova**

Graduate student, e-mail: 722bmb@gmail.com

**Sergey S. Samsonov**

Student, e-mail: 723\_sss@fb.tusur.ru

**Igor V. Filimonenko**

Student, e-mail: ifilimon96@mail.ru

Tomsk State University of Control Systems and Radioelectronics,  
40 Lenina Prospect, Tomsk, Russia 634050, Russia

**Abstract.** The feature selection is an NP-hard problem, it is guaranteed the optimal solution can be found only by a full search. In the article, we describe the approach to feature selection based on ranking and cross-validation. For the formation of optimal

feature sets, binary meta-heuristic algorithms are used: gravitational search algorithm, weed algorithm, monkey algorithm and krill herd algorithm.

**Keywords:** cross-validation, feature selection, classifiers, binary metaheuristics.

### References

1. Mekh M.A., Hodashinsky I.A. Sravnitel'nyy analiz primeneniya metodov differentsial'noy evolyutsii dlya optimizatsii parametrov nechetkikh klassifikatorov [Comparative analysis of differential evolution methods to optimize parameters of fuzzy classifiers] // *Izvestiya RAN. Teoriya i sistemy upravleniya = Journal of Computer and Systems Sciences International*. 2017. Vol. 56, No. 4. Pp. 616–626. (in Russian)
2. Hodashinsky I.A., Bardamova M.B. Binarizatsiya algoritma gravitatsionnogo poiska v zadachakh otbora priznakov dlya nechetkikh klassifikatorov [Binarization of the gravitational search algorithm in the feature selection problems for fuzzy classifiers] // XIII Mezhdunarodnaya nauchno-prakticheskaya konferentsiya «Elektronnyye sredstva i sistemy upravleniya» (Tomsk, TUSUR, 29 noyabrya – 1 dekabrya 2017 g.) = XIII International Scientific and Practical Conference "Electronic Tools and Control Systems" (Tomsk, TUSUR, November 29 - December 1, 2017). Tomsk. V-Spektr Publ. 2017. Vol. 2. Pp. 29–31. (in Russian)
3. Gandomi A.H., Alavi A.H. Krill herd: A new bio-inspired optimization algorithm // *Communications in Nonlinear Science and Numerical Simulation*. 2012. Vol. 17. Pp. 4831–4845.
4. Jalalirad A., Tjalkens T. Using feature-based models with complexity penalization for selecting features // *Journal of Signal Processing Systems*. 2018. Vol. 90. Pp. 201–210.
5. Mehrabian A.R, Lucas C. A novel numerical optimization algorithm inspired from weed colonization // *Ecological informatics*. 2006. Vol.1. Pp. 355–366.
6. Rashedi E., Nezamabadi-pour H., Saryazdi S. GSA: a gravitational search algorithm // *Information Sciences*. 2009. Vol.179. Pp. 2232–2248.
7. Wolpert D.H., Macready W.G. No Free lunch theorems for optimization // *IEEE Transactions on Evolutionary Computation*. 1997. Vol.1. Pp. 67–82.
8. Zhao R., Tang W. Monkey algorithm for global numerical optimization // *Journal of Uncertain Systems*. 2008. Vol.2. Pp. 165-176.