

ИСПОЛЬЗОВАНИЕ АЛГОРИТМА «СЕРОГО ВОЛКА» ДЛЯ ОПТИМИЗАЦИИ НЕЧЕТКОГО КЛАССИФИКАТОРА

Д.В. Зарипов студент 3-го курса каф. КИБЭВС

*Научный руководитель И.А. Ходашинский, профессор каф. КИБЭВС
г. Томск, ТУСУР, zdy@keva.tusur.ru*

Проект ГПО КИБЭВС-1211 – Нечёткие системы

ВВЕДЕНИЕ

Часто данные абсолютно нелинейные, и это затрудняет или даже исключает возможность применение точных методов анализа. Нередким стало использование нечетких классификаторов, использующих для оптимизации различные алгоритмы[1].

В данной работе для настройки нечеткого классификатора будет использован алгоритм «серого волка».

НЕЧЁТКАЯ СИСТЕМА

Нечёткая система может быть представлена как: $y = f(\mathbf{x}, \boldsymbol{\theta}, \mathbf{r},$
где \mathbf{x} – входной вектор, $\boldsymbol{\theta} = \|\theta_1, \dots, \theta_N\|$ — вектор параметров
антецедентов, $N = k * (\text{число параметров, описывающих одну}$
функцию принадлежности)* (число термов, описывающих одну
входную переменную), y — скалярный выход системы, $\mathbf{r} = \|\mathbf{r}_1, \dots, \mathbf{r}_R\|$
— вектор параметров консеквентов.

В нечеткой системе типа Питсбургский классификатор, i -ое правило имеет следующий вид:

IF $x_1 = A_{1i}$ AND $x_2 = A_{2i}$ AND ... AND $x_n = A_{ni}$ THEN $y = r_i$

где A_{ij} — лингвистический терм, которым оценивается переменная x_i ; r_i — значение консеквента i -го правила [2].

Нечёткую систему можно кодировать в виде вектора, который и будет обрабатываться нашими алгоритмом (рис. 1).

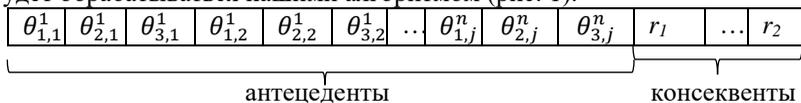


Рисунок 1 – Вектор параметров нечеткой системы

АЛГОРИТМ ОПТИМИЗАЦИИ СЕРОГО ВОЛКА

Наблюдения за охотой стаи «серых волков» привели к выводу о том, что в стае есть иерархическая зависимость, так на вершине стоит альфа, бета, гамма и в конце – омега. Остальные могут корректировать движение всей стаи до добычи. На этом и основан метод[3].

На вход алгоритм принимает вектор добычи или место куда должна попасть стая и векторы волков стаи. На выходе имеем вектор волка стоящего наиболее близко к добыче после всех перемещений стаи.

Рассмотрим пошаговую работу алгоритма.

1. Инициализация исходной стаи, а также вектора добычи:

$x^* = (x_1^*, x_2^*, \dots, x_n^*)$ - вектор добычи

$x_i = (x_{i1}, x_{i2}, \dots, x_{in})$ - вектор волка

2. Находятся три ближайших к цели волка по алгоритму близкому к алгоритму Евклида. Наиболее близкий принимается за альфу, за ним бета и гамма.

$$f_{(x_1 \dots x_n)} = \min \left\{ (x_1^* - x_1)^2 + (x_2^* - x_2)^2 + \dots + (x_n^* - x_n)^2 \right\}$$

3. Вычисляются координаты волка если он побежит к первым трем волкам.

Вычисляются векторы D_a, D_b, D_d и векторы X_1, X_3, X_2 .

$$\vec{D}_a = |\vec{C}_1 * \vec{X}_a - \vec{X}|, \quad \vec{D}_b = |\vec{C}_2 * \vec{X}_b - \vec{X}|, \quad \vec{D}_d = |\vec{C}_3 * \vec{X}_d - \vec{X}|$$

$$\vec{X}_1 = \vec{X}_a - \vec{A}_1 * (\vec{D}_a), \quad \vec{X}_2 = \vec{X}_b - \vec{A}_2 * (\vec{D}_b), \quad \vec{X}_3 = \vec{X}_d - \vec{A}_3 * (\vec{D}_d)$$

Векторы А и С вычисляются:

$$\vec{A} = 2\vec{a} * \vec{r}_1 - \vec{a}; \quad \vec{C} = 2 * \vec{r}_2; \quad \vec{a} = 2 - 2 * \left(\frac{itr}{maxitr} \right)$$

\vec{X}_a - вектор альфы; \vec{X}_b - вектор беты; \vec{X}_d - вектор гаммы;

r_1 и r_2 - генерирующиеся случайным образом вектора с максимальным значением равным единице;

It_r - номер итерации (повторения)

Maxitr - сколько повторений будет всего.

4. Нахождение новых координат волка по средствам среднего арифметического, из координат полученных в прошлом действии.

$$\bar{x}(i, 1) = \frac{x_1 + x_2 + x_3}{3}$$

5. Проверка количества итераций, если итераций меньше заданного количества вернутся к шагу 2. Если количество произведенных итераций совпадает с заданным, то вывести вектор альфы[4].

ЭКСПЕРИМЕНТ

Эффективность работы нечеткой системы, оптимизированной алгоритмом «серого волка», оценивалась на наборах данных для классификации из репозитория KEEL: appendicitis, balance, banana, Cleveland, glass, iris, phoneme, pima, ring, sonar, spambase, titanic, twonorm, wine. Результаты, приведенные в таблице 1, сравнивались с результатами работы эволюционными испанскими алгоритмами D-MOFARC, FARC-HD[4].

Таблица 1 – результаты работы алгоритма

	генерация		Wolf Algorithm			D-MOFARC		FARC-HD		
	Обуч	Тест	Обуч	СКО	Тест	СКО	Обуч	Тест	Обуч	Тест
appendicitis	69	72,7	92,82	1,37	85,08	4,04				
balance	46,6	44,8	92,99	0,62	88,31	1,59	89,4	85,6	92,2	91,2
banana	43,67	43,77	79,53	0,56	78,87	1,64	90,3	89	86	85,5
cleveland	55,12	54,88	63,85	0,67	58,20	2,41	90,9	52,9	82,2	58,3
glass	63,15	53,48	70,22	0,77	58,60	3,37	95,2	70,6	79	69
iris	94,16	93,33	99,03	0,43	96,11	2,11	98,1	96	98,6	95,3
phoneme	72,00	71,83	81,06	0,31	79,90	0,96	84,8	83,5	83,9	82,5
pima	62,83	67,44	79,11	0,67	76,24	1,36	82,3	75,5	82,3	76,2
ring	48,85	48,85	78,95	6,98	77,78	6,80	94,2	93,3	95,1	94
sonar	58,41	56,24	87,23	1,079	74,29	1,62				
spambase	58,92	58,95	81,21	8,24	79,33	7,55	91,7	90,5	92,4	91,6
titanic	68,60	68,60	78,69	0,22	78,55	1,50	78,9	78,7	79,1	78,8
twonorm	95,29	95,01	97,22	0,11	96,49	0,3	94,5	93,1	96,6	95,1
wine	89,46	89,31	99,74	0,18	96,31	2,46	100	95,8	100	95,5

ЗАКЛЮЧЕНИЕ

Алгоритм «серого волка» показал хорошую работу для оптимизации нечеткой системы на выборках: balance, iris, pima, twonorm, wine. Его показательным отличием является простота реализации и интеграции в нечеткие системы.

ЛИТЕРАТУРА

1. Леоненков А.В. Нечёткое моделирование в среде MATLAB и fuzzyTECH. // БХВ-Петербург. –2005. – С. 736.
2. Ходашинский И.А. Идентификация нечетких систем: методы и алгоритмы // Проблемы управления. –2009. – No. 4. – С. 15–23.
3. Mirjalili S., Lewis A. Grey Wolf Optimizer //Advances in Engineering Software. –2014. – No. 69 – P.46-61
4. Fazzolari M., Alcalá R., Herrera F.: D-MOFARC Classification Algorithm // Applied Soft Computing. – 2014 – No. 24 – P.470–481.