Об алгебро-логической коррекции в задачах распознавания по прецедентам*

Дюкова Е. В.[1], ***Любимцева М. М.***[2], ***Прокофьев П. А.***[3]
[1]edjukova@mail.ru; [2]m.lyubimtseva@gmail.com; [3]p_prok@mail.ru
[1,3]ВЦ РАН; [2]ВМК МГУ

Исследуются логические корректоры — модели распознающих алгоритмов, основанные на голосовании по корректным наборам элементарных классификаторов (эл.кл.). Вводится понятие *антимонотонного* корректного набора эл.кл. На базе антимонотонных корректных наборов эл.кл. построен логический корректор. Приведены результаты тестирования новой модели логического корректора на реальных даннных.

Ключевые слова: *распознающий алгоритм, логическая процедура распознавания, алгебро-логическая коррекция, элементарный классификатор, корректный элементарный классификатор, логический корректор, (монотонный) корректный набор элементарных классификаторов.*

Algebraic-logical correction in recognition problems*

Djukova E. V.[1], ***Lyubimtseva M. M.***[2], ***Prokofjev P. A.***[3]
[1,3]Dorodnitsyn Computing Centre, Russian Academy of Sciences; [2]Faculty of Computational Mathematics and Cybernatics, Moscow State University

A problem of constructing the correct recognition algorithms based on incorrect elementary classifiers (EC) is considered. A new type of correct sets of EC is suggested that is called antimonotonic. A new model of correct recognition algorithms based on antimonotonic correct sets of EC is constructed. This model is tested on real tasks.

Keywords: *recognition algorithm, logical recognition procedure, algebraic-logical correction, elementary classifier, correct elementary classifier, logical corrector, (monotonic) correct set of elementary classifiers.*

Введение

Рассматривается подход к задаче распознавания по прецедентам, базирующийся на применении аппарата дискретной математики (логических и алгебро-логических методов анализа данных). Важнейшими для этого направления исследований являются вопросы построения корректных распознающих алгоритмов, т. е. алгоритмов, правильно классифицирующих обучающие объекты.

Классические логические алгоритмы распознавания основаны на поиске в целочисленных данных конъюнктивных закономерностей или элементарных классификаторов (эл.кл.). В решающем правиле используется процедура голосования по найденным эл.кл. В случае целочисленной информации в роли эл.кл. выступают элементарные конъюнкции, определенные на наборах, являющихся признаковыми описаниями объектов. Элементарная конъюнкция считается корректной, если ее интервал истинности не содержит описаний обучающих объектов из разных классов. Корректность распознающего алгоритма обеспечивается корректностью используемых эл.кл. Основной задачей логического

Работа частично поддержана грантом РФФИ № 13-01-00787-а и грантом президента РФ НШ-4652.2012.1.

анализа данных в распознавании является поиск наиболее информативных корректных эл.кл.

Одним из новых направлений исследований в рассматриваемой области является синтез корректных логических распознающих процедур с использованием алгебраического подхода [1]. В данном случае в качестве базисных распознающих алгоритмов выступают эл.кл., не обязательно являющиеся корректными, а в качестве корректирующей функции берется булева функция. Голосование ведется по найденным корректным наборам эл.кл. Основной задачей алгебро-логической коррекции является поиск корректных наборов эл.кл. с хорошей распознающей способностью.

Идея алгебро-логического синтеза корректных процедур распознавания (логических корректоров) предложена в [2]. В указанной работе показано, что задача нахождения корректных наборов элементарных классификаторов равносильна задаче нахождения покрытий булевой матрицы, специальным образом построенной по обучающей выборке.

Подход развит в работах [3] и [4], в которых рассмотрены вопросы практического применения различных моделей логических корректоров.

В [4] на базе простейших эл.кл., порождаемых элементарными конъюнкциями ранга 1, построен алгоритм МОН. В качестве корректирующей булевой функции используется монотонная булевая функция. Для сокращения перебора при поиске корректных наборов эл.кл. с хорошей распознающей способностью использован генетический алгоритм из [5].

В [3] построен алгоритм LOBAGA, использующий произвольные эл.кл. в качестве базисных операторов. На этапе обучения алгоритм работает итеративно. На каждой итерации строится достаточно большой корректный набор эл.кл. с хорошей распознающей способностью, который называется локальным базисом. В рамках построенного локального базиса генетическим алгоритмом из [5] ищутся корректные наборы эл.кл. Каждому корректному набору эл.кл. приписывается вес, характеризующий его распознающую способность. Найденные корректные наборы эл.кл. пополняют семейство наборов эл.кл., построенное на предыдущих итерациях. Итерация заканчивается классификацией обучающих объектов путем голосования по всем найденным к данному моменту корректным наборам эл.кл. В результате происходит обновление веса каждого обучающего объекта, характеризующего типичность объекта для своего класса. Число итераций является параметром алгоритма.

В данной работе введено понятие антимонотонного корректного набора эл.кл. По аналогии с логическим корректором МОН из [4], использующим монотонные корректные наборы эл.кл., построен логический корректор АМОН, использующий антимонотонные корректные наборы эл.кл. Приведены результаты тестирования алгоритмов МОН, АМОН и LOBAGA на реальных данных. Тестируемые логические корректоры сравнивались с рядом распознающих алгоритмов из системы [6].

Основные понятия и обозначения

Рассматривается задача распознавания по прецедентам с непересекающимися классами $K_1, ..., K_l$, системой признаков $\{x_1, \ldots, x_n\}$ и набором обучающих объектов $\{S_1, \ldots, S_m\}$.

Пусть объект S имеет описание $(s_1, \ldots, s_n)$ в системе признаков $\{x_1, \ldots, x_n\}$ и пусть H — набор из r различных признаков $\{x_{j_1}, \ldots, x_{j_r}\}$. Тогда вектор $(s_{j_1}, \ldots, s_{j_r})$ называется подописанием объекта S по набору признаков H и обозначается (S, H).

Пусть далее $\sigma = (\sigma_1, \ldots, \sigma_r)$ — набор, в котором σ_q — допустимое значение признака x_{j_q} из H. Пара (H, σ) называется *элементарным классификатором* (эл. кл.). Число r называется *рангом эл. кл.* (H, σ).

Близость объекта S и эл.кл. (H, σ) оценивается величиной

$$B_{(H,\sigma)}(S) = \begin{cases} 1, \text{ если } (S, H) = \sigma, \\ 0, \text{ в противном случае.} \end{cases}$$

Пусть $U = \{(H^1, \sigma^1), \ldots, (H^d, \sigma^d)\}$ — набор эл.кл. Для объекта S через $\omega_U(S)$ обозначается вектор

$$(B_{(H^1,\sigma^1)}(S), \ldots, B_{(H^d,\sigma^d)}(S)),$$

называемый откликом набора эл.кл. U на объекте S.

Пусть T — подмножество множества обучающих объектов (*подвыборка*). Набор эл.кл. U называется *корректным для класса K ($K \in \{K_1, \ldots, K_l\}$) относительно подвыборки* T, если существует булева функция $F_K(t_1, \ldots, t_d)$, такая что для любой пары обучающих объектов $S_i \in T \cap K$ и $S_j \in T \setminus K$ выполняется одно из неравенств

$$F_K(\omega_U(S_i)) > F_K(\omega_U(S_j), \tag{1}$$

$$F_K(\omega_U(S_i)) < F_K(\omega_U(S_j). \tag{2}$$

Корректный для K набор U называется *монотонным*, если существует монотонная функция F_K, для которой всегда выполняется (1). Корректный для K набор U называется *антимонотонным*, если существует монотонная функция F_K, для которой всегда выполняется (2).

Алгоритмы MOH, AMOH и LOBAGA

На этапе обучения алгоритмов MOH, AMOH и LOBAGA обучающие объекты делятся на базовую подвыборку T_0 и настроечную подвыборку T_1. Базовая выборка используется для построения корректных набор эл.кл., настроечная выборка — для оценки распознающей способности наборов. Для каждого класса $K \in \{K_1, \ldots, K_l\}$ формируется семейство W_K, состоящее из корректных для K относительно T_0 наборов эл.кл. с хорошей распознающей способностью.

Алгоритм MOH. Алгоритм при обучении строит семейство W_K из монотонных корректных для K наборов эл.кл.

Определяется близость объекта S к обучающему объекту S_i по набору эл.кл. $U = \{(H^1, \sigma^1), \ldots, (H^d, \sigma^d)\}$

$$\delta_U(S, S_i) = \begin{cases} 1, & B_{(H_j,\sigma_j)}(S) \geqslant B_{(H_j,\sigma_j)}(S_i), \ j \in \{1, \ldots, d\}, \\ 0, & \text{иначе.} \end{cases}$$

Значение функционала

$$\Gamma'_K(U, S) = \frac{1}{|K \cap T_0|} \sum_{S_i \in K \cap T_0} \delta_U(S, S_i)$$

характеризует насколько объект S близок к классу K.

Распознающая способность корректного для K набора эл.кл. U оценивается величиной

$$\tau'_K(U) = \frac{1}{|T_1 \cap K|} \sum_{S_i \in T_1 \cap K} \Gamma'_K(U, S_i) - \frac{1}{|T_1 \setminus K|} \sum_{S' \in T_1 \setminus K} \Gamma'_K(U, S_i).$$

Пусть $\mathcal{U}^1$ — множество всех эл.кл. ранга 1, порождаемых обучающими объектами, и пусть L'_K — булева матрица, строящаяся по следующему правилу. Каждой строке матрицы L'_K соответствует пара обучающих объектов (S_i, S_j) из T_0, $S_i \in K$, $S_j \notin K$, каждому столбцу — эл.кл. (H, σ) из $\mathcal{U}^1$. Элемент матрицы L'_K, расположенный на пересечении строки (S_i, S_j) и столбца (H, σ), равен

$$B_{(H,\sigma)}(S_i) \wedge \neg B_{(H,\sigma)}(S_j).$$

Через $J(U)$ обозначим набор столбцов матрицы L'_K, порожденный эл.кл. из U. Набору столбцов $J(U)$ матрицы L'_K приписывается вес $\tau'_K(U)$.

Набор столбцов $J(U)$ называется покрытием булевой матрицы L, если в подматрице, составленной из столбцов $J(U)$ нет нулевой строки.

Утверждение 1 ([2]). *Набор эл.кл. $U \subseteq \mathcal{U}^1$ является монотонным корректным для класса K тогда и только тогда, когда набор столбцов $J(U)$ является покрытием матрицы L'_K.*

Таким образом, задача построения корректных для K наборов эл.кл. сводится в задаче построения покрытий матрицы L'_K с весами, близкими к максимальным. Для этого используется генетический алгоритм из [5]. Необходимое для формирования W_K число покрытий является параметром обучения.

Результатом обучения алгоритма МОН является совокупность построенных семейств наборов эл.кл. $W_{K_1}, \dots, W_{K_l}$.

При распознавании объект S относится к классу с максимальным значением функционала

$$\Gamma'(W_K, S) = \frac{1}{|W_K|} \sum_{U \in W_K} \Gamma'_K(U, S),$$

В случае существования нескольких классов, для которых значение $\Gamma'(W_K, S)$ максимально, происходит отказ от распознавания.

Алгоритм АМОН. Алгоритм при обучении строит W_K из антимонотонных корректных для K наборов эл.кл.

Для оценки близости объекта S к классу K вместо $\Gamma'_K(U, S)$ используется функционал

$$\Gamma''_K(U, S) = \frac{1}{|K \setminus T_0|} \sum_{S_i \in K \setminus T_0} (1 - \delta_U(S, S_i)).$$

Функционал для оценки распознающей способности набора U и оценка за отнесение объекта S в классу K по семейству W_K строятся аналогично $\tau'_K(U)$ и $\Gamma'(W_K, S)$, и соответственно имеют вид

$$\tau''_K(U) = \tfrac{1}{|T_1 \cap K|} \textstyle\sum_{S_i \in T_1 \cap K} \Gamma''_K(U, S_i) - \tfrac{1}{|T_1 \setminus K|} \textstyle\sum_{S' \in T_1 \setminus K} \Gamma''_K(U, S_i),$$
$$\Gamma''(W_K, S) = \tfrac{1}{|W_K|} \textstyle\sum_{U \in W_K} \Gamma''_K(U, S).$$

Для поиска антимонотонных корректных для K наборов строится булева матрица L''_K. Каждой строке L''_K соответствует пара обучающих объектов (S_i, S_j) из T_0, таких что $S_i \in$

K, $S_j \notin K$, каждому столбцу — эл.кл. $(H,\sigma) \in \mathcal{U}^1$. Элемент матрицы L''_K, находящийся в пересечении строки (S_i, S_j) и столбца (H,σ), равен

$$\neg B_{(H,\sigma)}(S_i) \wedge B_{(H,\sigma)}(S_j).$$

Утверждение 2. *Набор эл.кл. $U \subseteq \mathcal{U}^1$ является антимонотонным корректным для класса K тогда и только тогда, когда набор столбцов $J(U)$ является покрытием матрицы L''_K.*

Покрытию $J(U)$ матрицы L''_K приписывается вес $\tau''_K(U)$. Для поиска покрытий с весом, близким к максимальному, также используется генетический алгоритм [5].

Алгоритм LOBAGA. Алгоритм подробно описан в [3]. Строящиеся корректные наборы в отличие от алгоритмов МОН и АМОН состоят из эл.кл. произвольного ранга.

При инициализации берутся пустые семейства $W_{K_1}, \ldots, W_{K_l}$ и каждому обучающему объекту приписывается вес, равный $1/m$. В ходе последующих итераций веса объектов пересчитываются, и результирующий вес обучающего объекта характеризует типичность объекта для класса, к которому он принадлежит.

На каждой итерации специальным образом выбирается класс K, и формируется локальный базис $\mathcal{U}_K$ — достаточно большой корректный для K набор эл.кл. с хорошей распознающей способностью. В рамках локального базиса $\mathcal{U}_K$ генетическим алгоритмом ищется семество корректных наборов эл.кл. с распознающей способностью, близкой к максимальной. При вычислении распознающей способности корректных наборов эл.кл. учитываются веса объектов. Каждый найденный корректный набор эл.кл. U добавляется с весом α_U в семейство W_K. Вес α_U характеризует распознающую способность U. По найденным к настоящему моменту корректным наборам эл.кл. вычисляются оценки за отнесение каждого обучающего объекта к своему классу. На основании оценок пересчитываются веса объектов. Число итераций является параметром обучения.

Тестирование алгоритмов МОН, АМОН и LOBAGA

Алгоритмы МОН, АМОН и LOBAGA протестированы на 24 реальных задачах из области медицины, собранных в отделе Математических проблем распознавания и методов комбинаторного анализа ВЦ РАН [6], а также представленных в репозитории UCI. Характеристики задач представлены в табл. 1.

Тестирование проводилось по методу 10-fold cross-validation. В табл. 2 для каждого алгоритма указана величина

$$R = \frac{1}{10}\sum_{t=1}^{10} q_t,$$

где q_t — процент правильно распознанных объектов при тестировании с номером t.

Пусть для некоторых распознающих алгоритмов A_1 и A_2 на тестовой задаче Z получены соответственно результаты R_1 и R_2. Определим, что алгоритм A_1 на тестовой задаче Z показал лучшие результаты, чем алгоритм A_2, если $R_1 - R_2 \geqslant 3\%$. Будем говорить, что алгоритмы A_1 и A_2 на тестовой задаче Z показали сравнимые результаты, если $|R_1 - R_2| < 3\%$.

При тестировании МОН, АМОН и LOBAGA на задачах с действительнозначными данными (задачи 1–6, 15–24) предварительно осуществлялась корректная перекодировка методом из [7].

Проведено сравнение результатов тестирования алгоритмов МОН и АМОН на целочисленных задачах с представленными в [8] результатами тестирования других алгоритмов. Результаты счета для МОН приведены в табл. 3, а для АМОН — в таблице 4. В

Таблица 1: Характеристики задач

№	Задача	m	n	l	**Объектов в классах**	Тип данных
1	botwinimmuno	60	69	2	48/12	real
2	botwinSt	196	17	2	23/173	real
3	dorovski	33	12	2	16/17	real
4	ech_r	71	8	2	48/23	real
5	eco_l	144	7	4	76/33/24/11	real
6	Hepatit	155	19	2	32/123	real
7	input	344	9	2	218/126	int
8	manelis1	145	35	2	38/107	int
9	manelis2	107	35	2	35/72	int
10	manelis3	73	35	2	38/35	int
11	manelis4	110	35	2	38/72	int
12	matchak2	132	24	2	30/102	int
13	matchak3	269	24	2	51/218	int
14	matchak4	269	21	2	51/218	int
15	oil	114	5	3	60/15/39	real
16	patomorfoz	77	7	2	47/30	real
17	SARComa	80	18	2	40/40	real
18	sigapur	58	15	2	11/47	real
19	surv	77	8	2	52/25	real
20	botwinklbl	196	9	2	23/173	real
21	echu	131	9	2	89/42	real
22	heartUni	270	13	2	120/150	real
23	stupenexper	61	18	2	39/22	real
24	wineUni	178	13	3	59/71/48	real

табл. 5 указано число задач, на которых алгоритмы МОН и АМОН справились лучше (хуже/сравнимо), чем другие алгоритмы.

Значения $1, 0$ или -1 в табл. 4 (табл. 3) указывают на то, что на задаче алгоритм МОН (АМОН) работает соответственно лучше, сравнимо или хуже по сравнению с другим указанным алгоритмом.

Для обозначения алгоритмов в табл. 3, 4 и 5 приняты следующие сокращения:

1) ЛДФ – линейный дискриминант Фишера,
2) АВО – алгоритм вычисления оценок,
3) ГМ – алгоритм построения решающих деревьев генетическим метод,
4) НС – нейронная сеть,
5) МБРД – метод бинарного решающего дерева,
6) МБС – метод k ближайших соседей,
7) ГТТ – стохастический алгоритм голосования по тупиковым тестам,
8) C5.0 – алгоритм построения решающих деревьев,
9) C5.0 boost – C5.0 с использованием бустинга,
10) C5.0 prun – C5.0 с использованием отсечения,
11) AGI.Bias, AGI.La.sum – алгоритмы из [8],
12) LAD Tree – бустинг над решающими деревьями,
13) Random Forest – баггинг над решающими деревьями,

Таблица 2: Результаты работы МОН, АМОН и LOBAGA

Задача	**МОН**	**АМОН**	**LOBAGA**
botwinimmuno	65,63%	68,75%	**69,32%**
botwinSt	**48,72%**	45,24%	47,54%
dorovski	46,32%	**58,64%**	57,32%
ech_r	56,88%	54,89%	**57,21%**
eco_l	46,35%	**47,14%**	46,74%
Hepatit	66,53%	72,43%	**73,89%**
input	98,06%	**98,46%**	98.03%
manelis1	75,59%	75,50%	**76,45%**
manelis2	61,79%	64,60%	**66,39%**
manelis3	75,41%	**81,02%**	80,44%
manelis4	75,88%	79,90%	**80,35%**
matchak2	63,73%	62,06%	**64,05%**
matchak3	**57,62%**	55,24%	56,75%
matchak4	54,39%	55,47%	**55,98%**
oil	60,60%	51,28%	**62,46%**
patomorfoz	70,89%	**76,35%**	75,40%
SARComa	47,50%	**50,00%**	**50,00%**
sigapur	50,00%	50,00%	**50,03%**
surv	47,58%	56,38%	**57,67%**
botwinklbl	50,00%	50,00%	**50,24%**
echu	75,96%	74,77%	**78,65%**
heartUni	75,42%	74,58%	**75,76%**
stupenexper	58,22%	**64,04%**	62,09%
wineUni	57,97%	59,53%	**60,48%**

14) Simple CART – аналог алгоритма CART построения решающих деревьев,

15) Simple CART prun – Simple CART с использованием отсечения,

16) J48 – аналог C4.5.

Тестирование алгоритмов МОН, АМОН и LOBAGA показало:

1) результаты АМОН на трети тестовых задач лучше, чем результаты МОН, и почти на всех остальных тестовых задачах результаты МОН и АМОН сравнимы;

2) алгоритм LOBAGA, благодаря сложным конструктивным особенностям, на всех задачах либо сравним с дающим лучшие результаты алгоритмом, либо работает лучше МОН и АМОН. При этом строящиеся корректоры LOBAGA состоят из наборов со значительно меньшим числом эл.кл. по сравнению со строящимися корректорами МОН и АМОН.

Так как на одних задачах хорошие результаты дает корректор МОН, а на других — корректор АМОН, и корректор LOBAGA позволяет с меньшим числом эл.кл. достичь сравнимых результатов, то по-видимому, целесообразно модифицировать схему построения корректора LOBAGA таким образом, чтобы в семейства корректных наборов эл.кл. входили как монотонные наборы, так и антимонотонные наборы.

Сравнение алгоритмов МОН и АМОН с другими алгоритмами на целочисленных задачах показало:

Таблица 3: Сравнение МОН с другими алгоритмами

Алгоритм \№ задачи	7	8	9	10	11	12	13	14
ЛДФ	0	1	-1	1	-1	1	1	0
АВО	0	1	-1	0	-1	1	1	1
AGI.bias	0	-1	0	-1	0	1	0	0
ГМ	0	0	-1	0	-1	1	1	1
НС	0	0	0	-1	-1	0	1	-1
МБРД	0	1	1	1	0	1	1	1
МБС	0	0	-1	1	0	1	0	-1
ГТТ	1	1	0	1	0	-1	-1	-1
C5.0	0	-1	1	1	0	1	1	0
AGI.La.sum	1	-1	0	-1	0	1	1	0
LAD Tree	1	0	-1	-1	0	1	1	1
Random Forest	0	0	-1	-1	-1	1	1	0
Simple CART	1	1	-1	1	-1	1	1	0
C5.0 boost	0	-1	0	0	0	1	1	0
C5.0 prun	0	-1	1	1	0	1	1	0
Simple CART prun	0	0	1	1	-1	1	1	1
J48	0	-1	1	1	0	1	1	1

Таблица 4: Сравнение АМОН с другими алгоритмами

Алгоритм \№ задачи	7	8	9	10	11	12	13	14
ЛДФ	0	1	0	1	-1	0	0	0
АВО	0	1	-1	1	-1	1	1	1
AGI.bias	0	-1	1	0	0	1	0	0
ГМ	0	0	0	1	0	1	1	1
НС	0	0	0	0	-1	0	0	-1
МБРД	1	1	1	1	0	1	1	1
МБС	0	0	-1	1	1	1	0	-1
ГТТ	1	1	0	1	1	-1	-1	-1
C5.0	1	-1	1	1	0	0	1	1
AGI.La.sum	1	-1	1	0	0	0	1	1
LAD Tree	1	0	-1	0	0	1	1	1
Random Forest	1	0	-1	0	0	1	0	1
Simple CART	1	1	-1	1	0	1	1	1
C5.0 boost	0	-1	1	1	0	0	0	0
C5.0 prun	1	-1	1	1	0	0	0	0
Simple CART prun	1	0	1	1	0	1	1	1
J48	0	-1	1	1	1	0	1	1

1) алгоритм МОН на большинстве задач работает хуже, чем НС, показал результаты, сравнимые с МБС и ГТТ, и относительно остальных алгоритмов отработал лучше на большинстве задач;

Таблица 5: Сравнение МОН и АМОН с другими алгоритмами в совокупности

	МОН			АМОН				МОН			АМОН		
	+1	-1	0	+1	-1	0		+1	-1	0	+1	-1	
ЛДФ	4	2	2	2	1	5	AGI.La.sum	3	2	3	4	1	3
АВО	4	2	2	5	2	1	LAD Tree	4	2	2	4	1	3
AGI.bias	1	2	5	2	1	5	Random Forest	2	3	3	3	1	4
ГМ	3	2	3	4	0	4	Simple CART	5	2	1	6	1	1
НС	1	3	4	0	2	6	C5.0 boost	2	1	5	2	1	5
МБРД	6	0	2	7	0	1	C5.0 prun	4	1	3	3	1	4
МБС	2	2	4	3	2	3	Simple CART prun	5	1	2	6	0	2
ГТТ	3	3	2	4	3	1	J48	5	1	2	5	1	2
C5.0	4	1	3	5	1	2							

2) алгоритм АМОН на большинстве задач отработал хуже, чем НС, и в сравнении с остальными алгоритмами отработал лучше на большинстве задач.

Заключение

В статье рассмотрен алгебро-логический подход к построению корректных распознающих алгоритмов, основанных на голосовании по корректным наборам эл.кл. Введено новое понятие *антимонотонного* корректного набора эл.кл., и построен новый логический корректор АМОН. Проведено тестирование логического корректора АМОН, а также предложенных ранее логических корректоров МОН и LOBAGA, на реальных данных. Лучшие результаты дает логический корректор LOBAGA. На целочисленных задачах логические корректоры МОН и АМОН, как правило, работают лучше других тестируемых в работе алгоритмов.

Литература

[1] *Журавлев Ю. И* Об алгебраическом-подходе к решению задач распознавания или классификации // Пробл. кибер. М: Наука, 1978. Вып. 33. С. 5–68.

[2] *Дюкова Е. В., Журавлев Ю. И., Рудаков К. В.* Об алгебро-логическом синтезе корректных процедур распознавания на базе элементарных алгоритмов // ЖВМ и МФ. 1996. Т. 36.

[3] *Dyukova E. V., Prokofjev P. A.* Models of recognition procedures with logical correctors // Pattern Recognition and Image Analysis. 2013. Vol. 23, No. 2. Pp. 235–244.

[4] *Dyukova E. V., Zhuravlev Yu. I., Sotnezov M. R.* Construction of an ensemble of logical correctors on the basis of elementary classifiers // Pattern Recognition and Image Analysis. 2011. Vol. 21, No. 4. Pp. 599–605.

[5] *Sotnezov R. M.* Genetic algorithms for problems of logical data analysis in discrete optimization and image recognition // Pattern Recognition and Image Analysis. 2009. Vol. 19, No. 3. Pp. 469–477.

[6] *Журавлев Ю. И., Рязанов В. В., Сенько О. В.* «Распознавание» Математические методы. Программная Система. Практические применения М.: Фазис, 2006. С. 176.

[7] *Дюкова Е. В., Сизов А. В., Сотнезов Р. М.* О корректном понижении значности данных в задачах распознавания // Доклады Всероссийской конференции «Математические методы распознавания образов» (ММРО-15). Петрозаводск, 11–17 сентября 2011 г. С. 80–83.

[8] *Genrikhov I. E.* Synthesis and analysis of recognizing procedures on the basis of full decision trees // Pattern Recognition and Image Analysis. 2011. Vol. 21, No. 1. Pp. 45–51.