Московский государственный университет
имени М. В. Ломоносова

Факультет Вычислительной Математики и Кибернетики

Кафедра Математических Методов Прогнозирования

ДИПЛОМНАЯ РАБОТА СТУДЕНТА 517 ГРУППЫ

«О корректном понижении значности данных в задачах распознавания»

Выполнил студент 5 курса
517 группы:

Березин Алексей Андреевич

Научный руководитель:
доктор ф.-м. наук, доцент

Дюкова Елена Всеволодовна

Москва, 2014

Содержание

Введение

Для решения прикладных задач классификации, распознавания и прогнозирования, возникающих в различных плохо формализованных областях, успешно применяются методы распознавания образов, в частности методы, основанные на обучении по прецедентам. Один из подходов к задаче распознавания по прецедентам базируется на применении аппарата дискретной математики (логических и алгебро-логических методов анализа данных). Подход особенно эффективен в случае целочисленной информации низкой значности.

Сложными являются вопросы применения дискретного подхода в случае веществennозначной информации. Веществennозначная информация часто рассматривается как целочисленная высокой значности. Один из способов понижения ее значности состоит в преобразовании исходной выборки путем разбиения множества значений каждого признака на интервалы порогами. Значения признака, попавшие в один интервал, считаются близкими и кодируются одним числом. Однако, при произвольном выборе кодирующих порогов обучающие объекты, принадлежащие разным классам, могут стать неразличимыми. При данном способе преобразования информации важным является понятие корректного перекодирования данных, т.е. такого преобразования обучающей информации, при котором объекты из разных классов остаются различимыми. Корректное перекодирование основано на рассмотрении только таких порогов, которые содержат «различающую» информацию. Задача сводится к поиску кодирующего покрытия (покрытия специального вида для булевой матрицы, в которой число столбцов равно числу порогов). Построение наилучшей в смысле качества распознавания корректной перекодировки — труднорешаемая оптимизационная дискретная задача. Актуальными являются вопросы, связанные с выбором функционала, характеризующего качество перекодировки, и вопросы сокращения временных затрат.

Различают алгоритмы «частичного перекодирования» и «полного перекодирования».

В статье [1] предложен алгоритм CodeOne, основанный на идее «частичного перекодирования». В этом алгоритме на каждой итерации выбирается очередной объект из обучающей выборки и строится множество порогов, позволяющих отличить выбранный объект от объектов из других классов. При этом перекодируется только та часть обучающей выборки, которая образованна выбранным объектом и всеми объектами из других классов. Задача сводится к поиску кодирующего покрытия булевой матрицы, размер которой существенно меньше, чем при полном перекодировании. Частичное перекодирование позволяет рассматривать перекодировки, значность которых не более трех. Данных об экспериментальном исследовании алгоритма CodeOne в статье [1] не приведено.

В статьях [2] и [3] предложены алгорит2мы корректного понижения значности данных КОД1 и КОД2, основанные на «полном перекодировании» всей обучающей выборки. В алгоритме КОД1 качество перекодировки оценивается числом «типичных» значений признаков в перекодированной выборке. Для сокращения перебора предложена процедура, основанная на независимом перекодировании каждого признака. Тем не менее, время работы алгоритма КОД1 быстро растет с ростом размера задачи.

В алгоритме КОД2 из [3] предложен более эффективный функционал качества перекодирования по сравнению с функционалом качества алгоритма КОД1. Этот функционал учитывает число единиц в каждом столбце кодирующего покрытия и длину кодирующего покрытия. Для поиска оптимального кодирующего покрытия в КОД2 используется генетический подход, что позволяет снизить время работы алгоритма по сравнению с алгоритмом КОД1. Таким образом, алгоритм КОД2 превосходит КОД1 как по качеству перекодирования, так и по скорости работы.

В статье [4] предложен другой подход к использованию логических алгоритмов распознавания в случае вещественнозначной информации. Вводится понятие логической закономерности. Логическая закономерность

представляет собой конъюнкцию нескольких предикатов, каждый из которых задан на одном из признаков. Поиск логических закономерностей сводится к решению задачи целочисленного линейного программирования.

Целями данной работы являются:

1) развитие и исследование методов частичного корректного перекодирования данных;
2) сравнение двух подходов к обработке веществennознозначной информации логическими процедурами распознавания: подхода, основанного на корректном перекодировании исходных данных, и подхода, основанного на поиске логических закономерностей;
3) разработка нового функционала качества кодирующего покрытия;
4) разработка методик, позволяющих сократить временные затраты за счет сокращения числа кодирующих порогов.

В ходе выполнения дипломной работы были решены следующие задачи:

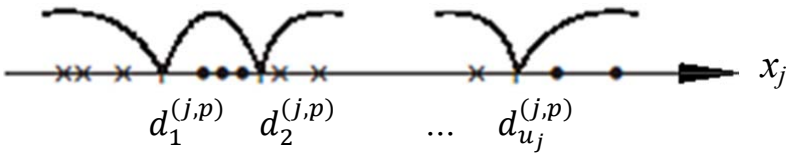
- разработан новый функционал качества перекодирования, более эффективный, чем существующие функционалы;
- разработан и реализован новый алгоритм CodeOne+, работающий по схеме алгоритма CodeOne и отличающийся от CodeOne отбором на каждой итерации оптимальной частичной перекодировки;
- разработана новая схема частичного перекодирования «один класс против всех остальных», и реализован алгоритм CodeClass, работающий по данной схеме;
- реализован алгоритм КОД2;
- проведено тестирование алгоритмов КОД2, CodeOne+, CodeClass на реальных задачах;
- разработана процедура предварительного отбора малоинформативных порогов;
- реализован и протестирован алгоритм предварительного отбора малоинформативных порогов MinD.

1. Обзор в области корректного применения логических алгоритмов распознавания в задачах с веществennозначной информацией

1.1. Основные понятия и обозначения

Рассмотрим задачу распознавания по прецедентам с l непересекающимися классами $K_1, K_2, \dots, K_l$ и множеством обучающих объектов $S_1, \dots, S_m$. Обучающий объект $S_i, i \in \{1,2, \dots, m\}$, задан описанием в системе признаков $\{x_1, \dots, x_n\}$ и имеет вид $(a_{i1}, \dots, a_{in})$, где $a_{ij} \in \mathbb{R}$, $\mathbb{R}-$ множество действительных чисел. Пусть $T = (a_{ij})_{m \times n}$ - таблица обучения. Тогда каждому столбцу таблицы T соответствуют один из признаков $x_1, x_2, \dots, x_n$, а каждой строке — один из обучающих объектов $S_1, \dots, S_m$.

Рассмотрим класс K_p, $p \in \{1,2, \dots, l\}$ и признак x_j, $j \in \{1,2, \dots, n\}$. Пусть $S_{i_1} = (a_{i_1 1}, \dots, a_{i_1 n})$ и $S_{i_2} = (a_{i_2 1}, \dots, a_{i_2 n})$ - обучающие объекты такие, что $S_{i_1} \in K_p$, $S_{i_2} \notin K_p$. Число $(a_{i_1 j} + a_{i_2 j})/2$ назовем порогом для пары (x_j, K_p), если для любого i из $\{1,2, \dots, m\}$ имеет место один из двух случаев: 1) $a_{ij} < \min(a_{i_1 j}, a_{i_2 j})$ или 2) $a_{ij} > \max(a_{i_1 j}, a_{i_2 j})$.



Через множество *кодирующих порогов* $D^{(j,p)} = \left\{d_1^{(j,p)}, \dots, d_{u_{(j,p)}}^{(j,p)}\right\}$ обозначим множество всех порогов для пары (x_j, K_p), $j \in \{1,2, \dots, n\}$. Суммой двух элементов $a_{i_1 j}$ и $a_{i_2 j}$ таблицы T по порогу $d \in D^{(j,p)}$ назовем число $(a_{i_1 j} \oplus a_{i_2 j}|_d)$ равное 1, если $a_{i_1 j}$ и $a_{i_2 j}$ лежат по разные стороны от порога d, и равное 0 в противном случае.

Через $П_p(T)$ будем обозначать последовательность всех порогов

$$d_1^{(1,p)}, \dots, d_{u_{(1,p)}}^{(1,p)}, d_1^{(2,p)}, \dots, d_{u_{(2,p)}}^{(2,p)}, \dots, d_1^{(n,p)}, \dots, d_{u_{(n,p)}}^{(n,p)}$$

Суммой двух строк таблицы T с номерами i_1 и i_2 по последовательности порогов $\Pi_p(T)$ назовем строку

$$\left(a_{i_1 1} \oplus a_{i_2 1}|_{d_1^{(1,p)}}, \dots, a_{i_1 1} \oplus a_{i_2 1}|_{d_{u_{(1,p)}}^{(1,p)}}, \right.$$
$$a_{i_1 2} \oplus a_{i_2 2}|_{d_1^{(2,p)}}, \dots, a_{i_1 2} \oplus a_{i_2 2}|_{d_{u_{(2,p)}}^{(2,p)}},$$
$$\dots$$
$$\left. a_{i_1 n} \oplus a_{i_2 n}|_{d_1^{(n,p)}}, \dots, a_{i_1 n} \oplus a_{i_2 n}|_{d_{u_{(n,p)}}^{(n,p)}} \right)$$

Пусть m_p - число обучающих объектов из класса K_p, $p \in \{1,2,\dots,n\}$, . Построим булеву матрицу L_T^p. Матрица L_T^p имеет размеры $h \times N$, где $h = m_p * (\sum_{j \neq p} m_j)$, $N = \left|D^{(1,p)}\right| + \dots + \left|D^{(n,p)}\right|$. Каждая ее строка формируется в результате попарного сложения строк таблицы T, одна из которых описывает объект из класса K_p, а другая из $\overline{K}_p = \{K_1 \cup \dots \cup K_l\} \backslash K_p$, по последовательности порогов $\Pi_p(T)$. Порядок выбора пар может быть задан произвольным образом. Множеству порогов $D^{(j,p)}$, $j \in \{1,2,\dots,n\}$, по построению соответствует группа из $u_{(j,p)}$ столбцов матрицы $L_{T,p}$, обозначаемая через G_j^p

Для каждого признака $j \in \{1,2,\dots,n\}$ определим коэффициент перемешанности признака $q_{(j,p)} = u_{(j,p)} / \left|K_p\right|$

Набор столбцов H матрицы L_T^p будем называть *кодирующим покрытием*, если выполнены следующие два условия:

1) H является покрытием L_T^p, т.е. для любой строки матрицы L_T^p в наборе H можно указать хотя бы один столбец, имеющий 1 на пересечении с этой строкой;

2) $H \cap G_j \neq \emptyset$ при $j = 1, \dots, n$., т.е. для любого признака x_j найдется хотя бы один порог из соответствующего ему набора $D^{(j,p)}$, который будет присутствовать в наборе H.

Кодирующее покрытие назовем *неприводимым*, если никакое его собственное подмножество кодирующим покрытием не является.

Число $\max_{j\in\{1,2,\dots,n\}}\left|H \cap G_j^p\right| + 1$ назовем *значностью* кодирующего покрытия H.

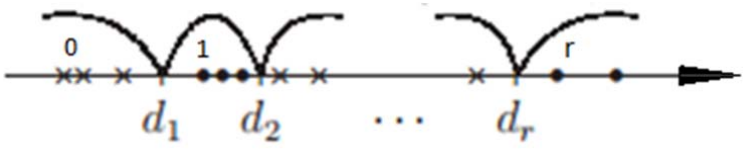
В результате перекодировки описание объекта $S_i, i \in \{1,2,\dots,m\}$, примет вид $(b_{i1}, \dots, b_{in})$, где b_{ij} определяется следующим образом: пусть $\{d_1, \dots, d_r\}$ – пороги из G_j, принадлежащие набору H. Для определенности считаем, что $d_1 < \dots < d_r$. Тогда возможны 3 ситуации:

1) если $a_{ij} \leq d_1$, то $b_{ij} = 0$;

2) если $d_t < a_{ij} \leq d_{t+1},\ t \in \{1,2,\dots,r-1\}$, то $b_{ij} = t$;

3) если $d_r < a_{ij}$, то $b_{ij} = r$.

При таком перекодировании описания объектов из разных классов остаются различимыми.



Пример. Пусть таблица обучения T задана в таблице 1.1.1, и пусть $S_1 \in K_1, S_2 \in K_2, S_3 \in K_2, S_4 \in K_3$. Упорядоченные данные по признакам x_1 и x_2относительно класса K_2 представлены в таблице 1.1.2 и таблице 1.1.3 соответственно. Построим для таблицы T последовательность всех порогов $П_2(T)$ и матрицы L_T^2

Таблица 1.1.1.

объект	признак x_1	признак x_2
S_1	2,5	0,6
S_2	3	0,4
S_3	3,7	0,7
S_4	5	0,5

Таблица 1.1.2

объект	Класс	признак x_1
S_1	$\overline{K}_2$	2,5
S_2	K_2	3
S_3	K_2	3,7
S_4	$\overline{K}_2$	5

Таблица 1.1.3

объект	Класс	признак x_2
S_2	$\overline{K}_2$	0,4
S_4	K_2	0,5
S_1	K_2	0,6
S_3	$\overline{K}_2$	0,7

Тогда $D^{(1,2)} = \{2{,}75; 4{,}35\},\ u_{(1,1)} = 2;\ \ D^{(2,2)} = \{0{,}45; 0{,}65\}, u_{(2,1)} = 2.$

Матрица L_T^2 по всей последовательности порогов представлена в таблице 1.1.4.

Таблица 1.1.4.

объекты	$d_1^{(1)}$	$d_2^{(1)}$	$d_1^{(2)}$	$d_2^{(2)}$
$S_2 \oplus S_4$	0	1	1	0
$S_2 \oplus S_1$	1	0	1	0
$S_3 \oplus S_1$	1	0	0	1
$S_3 \oplus S_4$	0	1	0	1

1.2. Алгоритм корректного понижения значности данных CodeOne

Алгоритм CodeOne, предложенный в [1], основан на схеме «один объект против всех объектов из других классов». Для каждого класса K_p, $p \in \{1,2,\dots,l\}$ поочередно выбирается каждый объект из обучающей выборки $S_i \in K_p$, $\mathrm{S_i} = (a_{i1},\dots,a_{in})$.

По объекту S_i и по всем объектам $S \notin K_p$ строятся множества порогов $D^{(j,p)}$, при этом $\left|D^{(j,p)}\right| < 3, \forall j$. По полученной последовательности порогов строится матрица $L_T^{S_i}$ размера $\overline{K}_p \times \left(\left|D^{(1,p)}\right| + \dots + \left|D^{(n,p)}\right|\right)$. Рассматриваются всевозможные кодирующие покрытия H матрица $L_T^{S_i}$. Каждая перекодировка порождает некоторое множество представительных наборов.

Перекодирование производится для всех распознаваемых объектов и всех объектов $S_j \in \overline{K}_p$. Значность перекодированных данных не более трех.

Во время процедуры распознавания, для каждого объекта из распознаваемой выборки добавляются голоса за класс K_p по всем полученным множествам представительных наборов.

После проведения всех итераций решающее правило присваивает каждому объекту из распознаваемой выборки метку класса с максимальной оценкой.

1.3. Алгоритм корректного понижения значности данных КОД1

Алгоритм полного перекодирования КОД1 предложен в [2].

Пусть T^H – перекодированная таблица обучения T. Рассмотрим $S_i \in K$, $S_i = (b_1,\dots,b_n)$. Значение -го признака b_j называется типичным для класса K, если

$$\frac{v_j(K,b_j)}{|K|} - \frac{v_j(\overline{K},b_j)}{|\overline{K}|} > \varepsilon_j,$$

где ε_j −фиксированный порог, $v_j(K, b_j)$ −число объектов из класса K, у которых значение -го признака после перекодирования равно b_j

Пусть информативность перекодирования, заданного кодирующим покрытием H, равна $I_H = I(K,H)/|K| + I(\overline{K},H)/|\overline{K}|$. Здесь $I(K,H)-$ число типичных значений признаков в классе K.

В алгоритме КОД1 независимо для каждого столбца исходной таблицы T строится множество возможных перекодировок. Перекодировки упорядочиваются по уменьшению числа типичных значений в соответствующем признаке.

Полученные перекодировки столбцов объединяются и упорядочиваются по уменьшению суммарной информативности I_H. Полученные перекодировки обучающей таблицы не обязательно корректны. Первая корректная перекодировка является результатом работы алгоритма.

1.4. Алгоритм корректного понижения значности данных КОД2

В [3] предложен алгоритм КОД2, основанный на схеме «полного перекодирования». По всем объектам обучающей выборки строится последовательность порогов $\Pi(T)$ и матрица L_T. В случае, когда число классов больше двух, при построении матрица L_T, имеет размеры $h \times N$, где $h = \sum_{i=1}^{l}\left(m_i \sum_{j=i+1}^{l} m_j\right)$, $N = \left|D^{(1,1)}\right| + \left|D^{(1,l)}\right| \ldots + \left|D^{(n,1)}\right| + \left|D^{(n,p)}\right|$.

Для сокращения перебора при поиске кодирующего покрытия матрицы L_T используется генетический алгоритм, схема которого описана в статье [5]. Особям соответствуют неприводимые кодирующие покрытия, а роль функции приспособленности играет функционал $f\ (H) = \sum_{j \notin H} w_j$, где $w_j, j \in \{1,2,\ldots,n\}$, – число единиц в j −ом столбце матрицы L_T.

После нахождения оптимального кодирующего покрытия H, производится перекодирование обучающей и распознаваемой выборок. В качестве распознающего алгоритма используется алгоритм голосования по

представительным наборам с ограниченной длиной. Для каждого объекта из распознаваемой выборки независимо вычисляются оценки принадлежности к каждому из классов, решающее правило присваивает объекту метку класса с максимальной оценкой.

1.5. Алгоритм распознавания, основанный на поиске логических закономерностей

В статье [4] предложен алгоритм, использующий понятие логической закономерности. Пусть обучающий объект $S = (a_1, \dots, a_n)$, где $a_j \in \mathbb{R}$. Логической закономерностью (ЛЗ) считается функция вида $P(S) = \varphi_{11}(S)\ \&\ \dots\ \&\ \varphi_{1i}(S)\&\ \dots\ \&\ \varphi_{nt}(S)$, где $\varphi_{ji}(S) = [\alpha_{ji} < a_j < \beta_{ji}]$ – i-ый предикат по -му признаку. Ставится задача поиска таких логических закономерностей $P(S)$, что $P(S) = 0$ для $S \in \overline{K}$ и число объектов $S \in K$, $P(S) = 1$ максимально. В случае целочисленной информации задача поиска ЛЗ сводится к задаче поиска представительных наборов с максимальной встречаемостью в своем классе. В геометрической интерпретации задача поиска логической закономерности выглядит как задача поиска по данным обучающей выборки прямоугольного координатного параллелепипеда, лежащего в некотором признаковом подпространстве, содержащим максимальное число объектов из класса K и не содержащее объектов из $\overline{K}$. Поиск логических закономерностей сводится к решению задачи целочисленного линейного программирования.

2. Новые алгоритмы корректного понижения значности данных CodeOne+, CodeClass

2.1. Алгоритм CodeOne+

В данной работе разработан алгоритм CodeOne+. В этом алгоритме используются схема работы, аналогичная схеме алгоритма CodeOne.

В отличие от алгоритма CodeOne, в алгоритме CodeOne+ на каждой итерации рассматриваются не всевозможные кодирующие покрытия, а выбирается оптимальное. Критерием оптимальности служит функционал качества

$$f_3(H) = \frac{1}{|H|}\sum_{j\in H}\frac{1}{w_j},$$

где $w_j, j \in \{1,2,\dots,n\}$, - число единиц в j −ом столбце матрицы $L_T^{S_i}$.

Перекодирование производится для всех распознаваемых объектов и всех объектов $S_j \in \overline{K}_p$. Достигается значность перекодированных данных не более двух.

Во время процедуры распознавания, для каждого объекта из распознаваемой выборки добавляются голоса за класс K_p по множеству представительных наборов, порожденных выбранной оптимальной перекодировкой.

После проведения всех итераций решающее правило присваивает каждому объекту из распознаваемой выборки метку класса с максимальной оценкой.

2.2. Алгоритм CodeClass

Алгоритм CodeClass, разработанный в данной работе, работает по схеме «один класс против всех остальных».

В алгоритме CodeClass на каждой итерации последовательно выбирается каждый из классов K_p. Для выбранного фиксированного класса строится последовательность порогов $\Pi_p(T)$ и матрица L_T^p. При построении последовательности порогов $\Pi_p(T)$ все объекты из $\overline{K}_p = \{K_1 \cup \ldots \cup K_l\} \backslash K_p$ считаются объектами из одного класса и пороги между ними не ставятся.

Для поиска кодирующего покрытия используется схема генетического алгоритма из [5]. Особи соответствует вектор $g = (g_1, \ldots, g_h)$, где $g_i \in \{1,2, \ldots, N\}$ – номер столбца, покрывающего *-ую* строку матрицы L_T^p. Таким образом каждая особь задает набор столбцов H покрытия матрицы L_T^p.

В качестве функции приспособленности используются функционалы

$$f_1(H) = \frac{1}{|H|}, \quad f_2(H) = \sum_{j \notin H} w_j,$$

$$f_3(H) = \frac{1}{|H|} \sum_{j \in H} \frac{1}{w_j}, \quad f_4(H) = \sum_{j \in H} q_j$$

где $w_j, j \in \{1,2, \ldots, n\}$, – число единиц в j −ом столбце матрицы L_T^p, q_j – коэффициент перемешанности признака j −го столбца. Решается задача минимизации функционала $f_2(H)$.

После нахождения оптимального кодирующего покрытия H, производится перекодирование обучающей и распознаваемой выборок. В качестве распознающего алгоритма используется алгоритм голосования по представительным наборам с ограниченной длиной. Для каждого объекта из распознаваемой выборки независимо вычисляются оценки принадлежности к каждому из классов, решающее правило присваивает объекту метку класса с максимальной оценкой.

2.3. Тестирование алгоритмов КОД2, CodeOne+, CodeClass на реальных данных

Эксперименты проводились на 29 реальных прикладных задачах из репозитория системы «Распознавание» [6] и "UCI Machine Learning Repository" [7]. Параметры задач представлены в Таблице 2.3.1. В качестве распознающего алгоритма использовалась процедура голосования по представительным наборам с ограничением по длине набора ([8], [9]). Деление выборки на обучающую и тестируемую осуществлялось методом кросс-валидации с пятью группами.

Введем обозначения:

КОД0 – алгоритм полного перекодирования, в котором используется одно кодирующее покрытие, содержащее все кодирующие пороги; RКОД0 – алгоритм распознавания, примененный к данным, перекодированным алгоритмом КОД0;

RCodeOne+ – алгоритм распознавания, примененный к данным, перекодированным алгоритмом CodeOne+;

RCodeClass - алгоритм распознавания, примененный к данным, перекодированным алгоритмом CodeClass,

RКОД2 – алгоритм распознавания, примененный к данным, перекодированным алгоритмом КОД2.

Качество алгоритмов сравнивалось по точности распознавания, т.е. по проценту правильно распознанных объектов тестируемой выборки . Так же учитывалось время работы алгоритмов.

Сравнение работы алгоритмов RКОД0 и RCodeOne+ представлены в таблице 2.3.2. По быстроте работы алгоритм RКОД0 превосходит RCodeOne+ на задачах с числом признаков больше 20. RКОД0 дал выигрыш в точности распознавания по сравнению с RCodeOne+ на 6 задачах, в которых примерно половина признаков имеет небольшую (меньше 5) значность.

Сравнение работы алгоритмов RCodeClass и RКОД2 представлено в таблице 2.3.3. В алгоритме RCodeClass тестировались новые функционалы

$$f_1(H) = \frac{1}{|H|}, \quad f_2(H) = \sum_{j \notin H} w_j, \quad f_4(H) = \sum_{j \in H} q_j$$

и функционал $f_3(H) = \frac{1}{|H|} \sum_{j \in H} \frac{1}{w_j}$, использующийся в RКОД2. Здесь H-кодирующее покрытие булевой матрицы L_T, $w_j -$ число единиц в $j -$ ом столбце L_T, $q_j = u_j / |K_p|$ – отношение числа порогов -го признака в p-ом классе к числу объектов в p-ом классе.

По таблице 2.3.3. видно, что на задачах с числом классов более двух, алгоритм RКОД2 работает быстрее, но алгоритм RCodeClass лучше по точности. Выигрыш по точности при использовании алгоритма RCodeClass увеличивается с увеличением числа классов. На задачах Oil ,Melanoma и hea_l использование функционала $f_4(H) = \sum_{j \in H} q_j$ дало выигрыш 4%, 3% и 2% в точности распознавания, по сравнению с точностью распознавания, получаемой при использовании функционалов $f_1(H)$, $f_2(H)$ и $f_3(H)$. На задачах с двумя классами, алгоритмы RКОД2 и RCodeClass показали примерно одинаковые результаты независимо от выбираемого функционала.

Сравнение алгоритмов RCodeClass и RCodeOne+ представлено в таблице 2.3.4. Из рассмотрения таблицы видно, что на большинстве задач алгоритм RCodeClass значительно превосходит алгоритм RCodeOne+ по точности распознавания. Наименьший выигрыш достигается на трех задачах с небольшим числом кодирующих порогов D (задачи input, manelis1, manelis3 со значением коэффициента перемешанности 0.01÷0.2). Время работы алгоритма RCodeOne+ меньше, чем у RCodeClass.

В таблице 2.3.5 представлено сравнение алгоритма RCodeClass и алгоритма поиска логических закономерностей. Алгоритм RCodeClass показывает результаты лучше на шести задачах с небольшим числом признаков (менее 10) и объектов (менее 100). Соответственно, алгоритм логических закономерностей показывает лучшие результаты на пяти задачах

с большим числом признаков. На остальных задачах результаты работы алгоритмов сравнимы.

Таким образом, по точности распознавания наилучшие результаты показали алгоритм поиска логических закономерностей и RCodeClass; по времени работы наилучшие результаты показали алгоритм поиска логических закономерностей и RCodeOne+.

Таблица 2.3.1 Параметры задач

	Задача	объектов	признаков	классов	распределение объектов по классам	Перемешанность признаков q Min÷max	Исходное число порогов		
							max	med	sum
1	wineUni	178	13	3	60, 70, 48	0.07÷0.32	54	38	470
2	eco_l	144	7	4	76, 33, 24, 11	0.01÷0.21	36	22	131
3	melanoma	70	33	3	29, 20, 21	0.1÷0.5	41	29	946
4	oil	115	5	3	60, 15, 40	0.17÷0.28	33	32	148
5	surv	77	8	2	52,25	0.03÷0.4	31	22	173
6	Hepatit	155	19	2	32,123	0.01÷0.30	46	2	197
7	input	344	9	2	218,126	0.01÷0.02	9	7	66
8	manelis1	145	35	2	38,107	0.007÷0.2	35	1	290
9	manelis2	107	35	2	25,72	0.01÷0.2	29	1	244
10	manelis3	73	35	2	38,35	0.01÷0.2	29	1	237
11	manelis4	110	35	2	38,72	0.01÷0.24	27	1	249
12	matchak2	132	24	2	30,102	0.01÷0.34	45	2	202
13	credit_I	342	15	2	152,190	0.003÷0.4	137	10	466
14	patomorfoz	77	7	2	47,30	0.09÷0.42	33	23	157
15	sarcoma	80	18	2	40,40	0.03÷0.42	44	29	472
16	sigapur	58	15	2	11,47	0.10÷0.15	9	9	116
17	echu	131	9	2	89,42	0.01÷0.35	46	28	192
18	heartUni	270	13	2	120,150	0.004÷0.24	93	3	283
19	stupenexper	61	18	2	39,22	0.08÷0.47	29	19	317
20	botwinSt	196	17	2	23,173	0.03÷0.17	34	12	219
21	dorovski	33	12	2	16,17	0.03÷0.57	21	12	146
22	ech_r	71	8	2	48,23	0.01÷0.42	30	18	140
23	eco_r	179	7	8	66,44,28,24,14,1,1,1	0.01÷0.19	35	27	150
24	hea_r	136	13	5	78,24,14,15,5	0.007÷0.27	54	3	177
25	hea_l	167	13	5	86,34,22,30,8	0.006÷0.14	27	2	115
26	image_l	210	16	7	30,30,30,30,30,30,30	0.005÷0.22	50	32	462
27	houm_l	242	13	5	16,91,93,27,15	0.02÷0.08	30	7	144
28	houm_r	264	13	5	8,98,117,25,16	0.004÷0.06	16	6	91
29	sclif	116	41	4	36,36,30,14	0.08÷0.36	42	10	568

Таблица 2.3.2.Сравнение точности распознавания с использованием алгоритма RCodeOne+ и RKOД0

Задача		Время работы, сек		Точность распознавания, %		
№	имя	RKOД0	RCodeOne+	RKOД0	RCodeOne+	Разность RCodeOne+ и RKOД0
2	eco_l	4	2	47	61	14
27	houm_l	39	44	18	31	13
3	melanoma	28	48	36	45	9
20	botwinSt	18	27	64	73	9
28	houm_r	46	57	24	29	5
8	manelis1	59	114	78	82	4
7	input	28	10	94	97	3
23	eco_r	7	6	37	40	3
26	image_l	55	47	18	21	3
31	soybean	10	35	98	100	2
6	Hepatit	18	24	79	80	1
30	lung cancer	10	51	42	43	1
11	manelis4	39	65	76	76	0
19	stupenexper	6	3	59	59	0
22	ech_r	1	0.4	64	63	-1
9	manelis2	35	63	68	66	-2
10	manelis3	19	32	68	66	-2
21	dorovski	1	0.4	50	48	-2
13	credit_I	71	56	83	79	-4
24	hea_r	11	14	54	44	-10
25	hea_l	18	23	50	40	-10
5	surv	1	0.7	49	35	-14
12	matchak2	20	34	73	57	-16
18	heartUni	33	23	80	63	-17

Таблица 2.3.3. Сравнение точности распознавания с использованием алгоритмов RCodeClass и RКОД2

Параметры задач					Время работы, сек.		Точность распознавания, %		
№	имя	объектов	признаков	классов	RCodeClass	RКОД2	RCodeClass	RКОД2	RCodeClass – RКОД2
26	image_l	210	16	7	80	23	86	71	15
27	houm_l	242	13	5	97	34	56	43	13
4	Oil*	115	5	3	3	2	65	54	11
3	Melanoma*	70	33	3	194	60	60	50	10
28	houm_r	264	13	5	101	38	64	57	7
2	eco_l	144	7	4	11	5	93	87	6
23	eco_r	179	7	8	34	7	66	63	3
1	wineUni	178	13	3	60	32	95	92	3
25	hea_l*	167	13	5	92	29	57	55	2
31	soybean	47	35	4	142	46	100	98	2
24	hea_r	136	13	5	47	16	59	58	1
30	lung cancer	32	55	3	184	102	49	48	1

Таблица *2.3.4. Сравнение* алгоритмов RCodeClass и RCodeOne+

Параметры задач					Времемя работы, сек		Точность распознавания		
№	имя	объектов	признаков	классов	RCodeClass	RCodeOne+	RCodeClass	RCodeOne+	RCodeClass – RCodeOne+
4	oil	115	5	3	3	2	65	22	43
14	patomorfoz	77	7	2	21	5	84	45	39
2	eco_l	144	7	4	11	2	93	61	32
23	eco_r	179	7	8	34	6	66	45	21
12	matchak2	132	24	2	140	34	73	57	16
15	sarcoma	80	18	2	24	7	66	50	16
3	melanoma	70	33	3	194	48	60	45	15
5	surv	77	8	2	3	1	50	35	15
24	hea_r	136	13	5	47	12	59	44	15
22	ech_r	71	8	2	3	1	76	63	13
18	heartUni	270	13	2	89	23	82	73	9
30	lung cancer	32	55	3	184	52	49	43	6
20	botwinSt	196	17	2	86	27	88	83	5
9	manelis2	107	35	2	253	63	70	66	4
13	credit_I	342	15	2	198	56	81	79	2
31	soybean	47	35	4	142	35	100	100	0
7	input	344	9	2	48	10	96	97	-1
8	manelis1	145	35	2	458	114	78	82	-4
10	manelis3	73	35	2	129	32	67	74	-7

Таблица 2.3.5. Сравнение точности распознавания с использованием алгоритмов RCodeClass и Логических Закономерностей

№	имя	объектов	признаков	классов	перемешанность	RCodeClass	Логические закономерности	CodeClass -RЛЗ
17	echu	131	9	2	0.01÷0.35	70	50	20
22	ech_r	71	8	2	0.01÷0.42	76	66	10
24	hea_r	136	13	5	0.01÷0.19	59	50	9
19	stupenexper	61	18	2	0.08÷0.47	91	84	7
25	hea_l	167	13	5	0.007÷0.27	57	50	7
21	dorovski	33	12	2	0.03÷0.57	66	61	5
3	melanoma	70	33	3	0.1÷0.5	60	58	2
18	heartUni	270	13	2	0.004÷0.24	82	80	2
13	credit_l	342	15	2	0.003÷0.4	81	80	1
20	botwinSt	196	17	2	0.03÷0.17	88	87	1
1	wineUni	178	13	3	0.07÷0.32	95	95	0
6	Hepatit	155	19	2	0.01÷0.30	80	80	0
7	input	344	9	2	0.01÷0.02	96	96	0
2	eco_l	144	7	4	0.01÷0.21	93	94	-1
4	oil	115	5	3	0.17÷0.28	65	68	-3
8	manelis1	145	35	2	0.007÷0.2	78	81	-3
16	sigapur	58	15	2	0.10÷0.15	91	94	-3
14	patomorfoz	77	7	2	0.09÷0.42	84	88	-4
11	manelis4	110	35	2	0.01÷0.24	76	83	-7
10	manelis3	73	35	2	0.01÷0.2	67	75	-8
27	houm_l	242	13	5	0.005÷0.22	56	67	-11
28	houm_r	264	13	5	0.02÷0.08	64	76	-12

3. Предварительный отбор малоинформативных порогов

3.1. Алгоритм MinD

Поиск корректного перекодирования данных можно вести не только по всему множеству порогов, но и по подмножеству наиболее информативных порогов. Предварительный отбор малоинформативных порогов позволяет уменьшить перебор при поиске кодирующего покрытия H матрицы L_T^p, снизить влияние шумовых данных и улучшить интерпретацию признаков.

Пусть $S = (a_1, \dots, a_n)$ - объект из обучающий выборки. Зафиксируем произвольный класс K_p, $p \in \{1,2, \dots, l\}$. Рассмотрим предикаты вида $\varphi_{\alpha\beta}(S) = [\alpha < a_j < \beta]$, определяемые двумя порогами из множества порогов $j-$го признака для класса K_p: $\alpha, \beta \in D^{(j,p)}$. Через $g(\varphi_{\alpha\beta})$ обозначим число объектов $S_i \in K_p$, для которых $\varphi_{\alpha\beta}(S_i) = 1$. Через $b(\varphi_{\alpha\beta})$ обозначим число объектов $S_i \in \overline{K}_p$ для которых $\varphi_{\alpha\beta}(S_i) = 1$.

Обозначим за $l(\varphi_{\alpha\beta})$ длину отрезка $[\alpha, \beta] = \beta - \alpha$.

Предикат $\varphi_{\alpha\beta}$ информативен, если он выделяет много объектов из класса K_p и мало объектов из класса $\overline{K}_p$. Критериями информативности могут служить различные отношения между $g(\varphi_{\alpha\beta})$ и $b(\varphi_{\alpha\beta})$. В простейшем случае это $I_1(\varphi) = g(\varphi_{\alpha\beta}) - b(\varphi_{\alpha\beta})$. При сильно неравномерном распределении объектов между классами или в многоклассовой задаче, можно ввести критерии с поправкой на размер классов: $I_2(\varphi) = \frac{g(\varphi_{\alpha\beta})}{|K_p|} - \frac{b(\varphi_{\alpha\beta})}{|\overline{K_p}|}$ и $I_3(\varphi) = \sqrt{\frac{g(\varphi_{\alpha\beta})}{|K_p|}} - \sqrt{\frac{b(\varphi_{\alpha\beta})}{|\overline{K_p}|}}$. Преимущества критерия $I_3(\varphi)$ видны на примере данных таблицы 3.1.1

Таблица 3.1.1

$\frac{g(\varphi_{\alpha\beta})}{\lvert K_p\rvert}$	$\frac{b(\varphi_{\alpha\beta})}{\lvert\overline{K_p}\rvert}$	$I_2(\varphi)$	$I_3(\varphi)$
4	0	4	2
8	4	4	0.8

Алгоритм предварительного отбора малоинформативных порогов MinD может быть выполнен после построения последовательности порогов $\Pi_p(T)$.

Все признаки упорядочиваются по уменьшению коэффициента перемешанности $q_{(j,p)}$. Отбор порогов производится для γ процетов наиболее перемешанных признаков , γ является параметром алгоритма.

Для каждого из выбранных признаков независимо осуществляется проход по всем четверкам соседних порогов d_1, d_2, d_3, d_4, которые на интервале $[d_2, d_3]$ содержат объекты из класса K_p. Пара порогов d_2 и d_3 помечается как малоинформативная, если

1. $l(\varphi_{d_2 d_3}) < \lambda * \min\left(l(\varphi_{d_1 d_2}), l(\varphi_{d_3 d_4})\right)$, где -параметр алгоритма.
2. $I\left(\varphi_{d_2 d_3}\right) < \mu * \left(-I\left(\varphi_{d_1 d_4}\right)\right)$, где -параметр алгоритма

Удаляется δ процентов пар порогов, дающих максимальный выигрыш выигрыша информативности после удаления, т.е. $\mu * \left(-I\left(\varphi_{d_1 d_4}\right)\right) - I\left(\varphi_{d_2 d_3}\right)$.

Параметры $\gamma, \lambda, \mu, \delta$ влияют на осторожность алгоритма при удалении порогов.

3.2. Тестирование алгоритма MinD

Тестирование алгоритма MinD производилось на реальных задачах из репозиториев [6] и [7]. Для перекодирования использовался алгоритм КОД0, для распознавания использовался алгоритм RКОД0 (см. п. 2.3).

Тестирование проводилось с различными значениями параметров $\gamma, \lambda, \mu, \delta$. В таблице *3.2.1.* представлены результаты счета с параметрами $\lambda = 1, \gamma = 0.5,\ \mu = 0.5,\ \delta = 0.5$. При этих значениях точность распознавания с использованием алгоритма MinD оказалась не ниже, чем без этого алгоритма.

Результаты тестирования показывают, что при указанных параметрах происходит отсеивание до 30% наименее информативных порогов. Время счета уменьшается на 10%-15%.

Таблица 3.2.1. Результаты тестирования алгоритма MinD.

	Звдача	Исходное число порогов			Итоговое число порогов			Время работы, с		Процент найденных малоинформативных порогов
		max	med	sum	max	med	sum	КОД0	КОД0 и MinD	
1	wineUni	54	38	470	48	35	416	47	42	11%
3	melanoma	41	29	946	29	22	716	28	23	24%
4	oil	36	28	145	28	21	113	3	3	22%
5	surv	31	22	173	22	16	123	1	1	29%
9	manelis2	29	1	244	21	1	216	35	31	11%
12	matchak2	45	2	202	28	2	134	20	17	34%
14	patomorfoz	33	23	157	31	19	143	12	11	9%
19	stupenexper	29	19	317	20	13	245	6	5	23%
24	hea_r	54	3	177	48	3	141	11	9	20%
29	Sclif	62	16	756	58	14	694	61	57	8%

Заключение

1) В работе построены и исследованы новые методы корректного понижения значности целочисленных данных. Реализованы и протестированы алгоритмы частичного корректного понижения значности целочисленных данных CodeOne+ и CodeClass. Лучшие результаты по точности распознавания показал алгоритм RCodeClass, по скорости работы RCodeOne+.
2) Проведено сравнение двух подходов к обработке веществennозначной информации логическими процедурами распознавания: подхода, основанного на корректном перекодировании исходных данных, и подхода, основанного на поиске логических закономерностей. По скорости работы лучшие результаты показал алгоритм поиска логических закономерностей, а по точности распознавания алгоритм RCodeClass оказался сравним с алгоритмом поиска логических закономерностей.
3) Проведено тестирование различных функционалов качества кодирующего покрытия с использованием алгоритма RCodeClass. Показано, что лучшая точность распознавания достигается при использовании предложенного в работе функционала $f_4(H) = \sum_{j \in H} q_j$ (где q_j- коэффициент перемешанности j-го признака).
4) Разработан алгоритм предварительного отбора малоинформативных порогов MinD. Данный алгоритм позволяет уменьшить число порогов при поиске корректной перекодировки исходных данных до 30 процентов, что позволяет уменьшить время работы алгоритма RКОД0 на 10-15 процентов.

Литература

1. Дюкова Е.В., Карнеева И.Л. Модели распознающих алгоритмов, основанные на различных способах перекодировки исходной информации // Математические методы в распознавании образов и дискретной оптимизации. М.: ВЦ АН СССР. 1990. С. 43-56.
2. Дюкова Е.В., Журавлев Ю.И., Песков Н.В., Сахаров А.А. Обработка вещественнозначной информации логическими процедурами распознавания // Искуственный интеллект. НАН Украины, 2004. №2. С. 80-85.
3. Дюкова Е.В., Сизов А.В., Сотнезов Р.М. Об оптимальном корректном перекодировании целочисленных данных в распознавании // Инфоратика и её применения. М.:«ТОРУС ПРЕСС» 2012. Т.6, №4 С. 61-65.
4. Рязанов В.В. Логические закономерности в задачах распознавания (параметрический подход) // Журнал вычислительной математики и математической физики. М.: Наука 2007. Т.47, №10 С. 1793-1808.
5. Sotnezov R.M. Genetic Algorithms for Problems of Logical Data Analysis in Discrete Optimization and Image Recognition // Pattern Recognition and Image Analysis – Pleiades Publishing 2009. Vol.19, No.3. Pg. 469-477.
6. Журавлев Ю.И., Рязанов В.В., Сенько О.В. «Распознавание». Математические методы. Программная Система. Практические применения. // М.:Фазис, 2006. 176 с.
7. UC Irvine Machine Learning Repository // http://archive.ics.uci.edu/ml/ .
8. Дюкова Е.В. Дискретные (логические) процедуры распознавания: принципы конструирования, сложность реализации и основные модели // М.: МПГУ, 2003. 29 с.
9. Баскакова Л.В., Журавлёв Ю.И. Модель распознающих алгоритмов с представительными наборами и системами опорных множеств // Журнал вычислительной математики и математической физики , 1981. Т. 21, №5. С. 1264-1275.