МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
им. М.В. Ломоносова

ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И КИБЕРНЕТИКИ
КАФЕДРА МАТЕМАТИЧЕСКИХ МЕТОДОВ ПРОГНОЗИРОВАНИЯ

Дипломная работа:

«Логические алгоритмы классификации: проблема переобучения и применение в задачах медицинской диагностики»

Выполнила студентка 517 группы:
Цурко В.В.

Научный руководитель:
к.ф.-м.н. Воронцов К. В.

Москва
2009

Содержание

1 Введение

Алгоритмы классификации широко применимы к задачам медицинской диагностики. Исходная информация, с которой приходится работать специалистам-врачам, как правило, характеризуется большой размерностью (десятки показателей, измеренных во времени), содержит различного рода ошибки, неточности, противоречия и пропуски. Все это затрудняет «ручной» анализ данных и процесс принятия решения. При отсутствии ограничений временных ресурсов, возможности привлечения большого числа компетентных экспертов принципиально возможно решение многих задач высокой сложности и размерности исходных данных. Необходимость в алгоритмах распознавания возникает в случае ограниченности ресурсов, недостатка времени, дефицита экспертов. Именно такая ситуация является типичной для большинства задач принятия решений в медицинской диагностике и лечении.

В работе рассматривается задача предсказания отдаленного результата хирургических операций. Данные задачи представляют собой матрицу объекты-признаки. Объектами являются больные, которым была проведена операция, признаками показатели, измеренные у больных до и после операции, целевым признаком является отдаленный результат операции: положительный исход (успешная операция) или

отрицательный исход (рецедив заболевания). Задача характеризуются большой размерностью, наличием пропусков в данных, неточностью измерения значений признаков, небольшой длиной выборки, маленьким количеством объектов с отрицательным отдаленным результатом операции.

Решается задача анализа и отбора признаков и объектов, прогнозирования исхода операции по данным, измеренным до операции, прогнозирования исхода по данным, измеренным до и непосредственно после операции. Если при построении первого прогноза предсказывается неблагоприятный исход для больного, то, возможно, его надо лучше готовить к операции. Сразу после операции прогноз тоже имеет смысл делать для того, чтобы интенсивнее наблюдать за больными, склонными к неблагоприятному исходу.

Цель работы — предложить метод, точно решающий задачи, характеризующиеся малостью выборки, большим количеством признаков, неточностью данных, наличием пропусков.

Приоритетным является использование логических алгоритмов классификации. Они широко применимы на практике, в их основе лежат интуитивно понятные идеи, эксперт-врач может легко интерпретировать логические правила, из которых построен алгоритм.

В первом разделе настоящей работы обсуждается общая постановка задачи, специфика рассматриваемой задачи, цели исследования.

Во втором разделе дается определение логической закономерности, правила, приводятся стандартные алгоритмы построения логических правил, дается обзор методов выбора наиболее информативных правил. Рассматриваются логические алгоритмы: решающий список, голосование, методы экспериментального оценивания и выбора наилучшего алгоритма.

В третьем разделе описывается предметная область задачи, освещаются особенности данных. Производится анализ данных, выявляются информативные признаки и объекты, произодится фильтрация объектов-выбросов. К задаче применяются стандартные алгоритмы классификации библиотеки WEKA и специально разработанные логические алгоритмы классификации, анализируется результат.

1.1 Задача классификации

Рассматриваемая задача ставится и решается как задача классификации.

Дано:

X — множество объектов;

$Y = \{Y_1, \ldots, Y_M\}$ — множество возможных ответов, M — число классов. Рассматриваемая задача является двухклассовой $Y = \{0, 1\}$, где класс 0 — положительный исход, класс 1 — отрицательный исход.

$X^\ell = \{(x_1, y_1), \ldots, (x_\ell, y_\ell)\}, x_i \in X, y_i \in Y, i = \overline{1, \ell}$ — *обучающая выборка.*

Предполагается, что существует некоторая неизвестная зависимость $y^* \colon X \to Y$, такая что $y^*(x_i) = y_i, i = \overline{1, \ell}$.

Необходимо построить алгоритм $a \colon X \to Y$, приближающий y^*, причем построенный алгоритм должен давать приемлемое качество классификации и вне обучающей выборки.

Опр. 1.1 *Признаком называется отображение $f \colon X \to D_f$, описывающее результат измерения некоторой характеристики объекта, где D_f — заданное множество. В зависимости от множества D_f признаки делятся на следующие типы:*

- *бинарный признак: $D_f = \{0, 1\}$;*

- *номинальный признак: D_f — конечное множество;*
- *порядковый признак: D_f — упорядоченное конечное множество;*
- *количественный признак: $D_f = \mathbb{R}$.*

Пусть имеется набор признаков $f_1, \ldots, f_n$. Вектор $f_1(x), \ldots, f_n(x)$ называют признаковым описанием объекта $x \in X$.

1.2 Постановка задачи

1. Анализ признакового описания задачи, измерение информативностей признаков $f_1, \ldots, f_n$.
2. Анализ объектов, выявление нетипичных объектов задачи.
3. Применение стандартных алгоритмов классификации к данной задаче классификации.
4. Реализация алгоритмов в среде MATLAB, применение к задаче.
5. Сравнительный анализ и оценивание алгоритмов.

2 Логические алгоритмы классификации

2.1 Логические закономерности

2.1.1 Понятие закономерности

Неформально *закономерность* — это предикат $\varphi(x)\colon X \to \{0, 1\}$, на области истинности которого искомую зависимость $y(x)$ можно считать константой. Если $\varphi(x) = 1$, то x принадлежит классу c, где c — фиксированный класс из Y. Если $\varphi(x) = 1$, то говорят, что φ *покрывает* x. Строгое определение будет дано ниже. Наибольшую ценность имеют закономерности, которые покрывают достаточно много объектов класса c и достаточно мало объектов всех остальных классов и имеют легко интерпретируемое выражение.

Чаще всего закономерности строятся в виде конъюнкций простых условий. Этот способ наиболее близок к интуитивному представлению о закономерностях.

Опр. 2.1 *Термами $t(x)$ будем называть предикаты следующего вида:*

- *для номинальных, бинарных и порядковых признаков:*
  - $f_i(x) = \theta$;
  - $f_i(x) \neq \theta$.
- *для количественных признаков:*
  - $f_i(x) \leqslant \theta$;
  - $f_i(x) \geqslant \theta$;
  - $\theta \leqslant f_i(x) \leqslant \theta'$.
- *для признаков любого типа:*
  - *Значение $f_i(x)$ задано;*

– *Значение $f_i(x)$ не задано.*

где θ, θ' — константы, называемые порогами . Допустимые значения порогов для признака f_i будем обозначать $\Theta_i = \{\theta_{ij}\}$. Обозначим через ξ_{ij} интервал значений признака $f_i(\theta_{ij}; \theta_{i,j+1})$.

Опр. 2.2 *Допустимым множеством термов $T_i(\Theta_i)$ на обучающей выборке X^ℓ относительно признака f_i будем называть множество всех возможных термов, пороги в которых берутся из множества Θ_i.*

Опр. 2.3 *Логическая закономерность удовлетворяет требованию интерпретируемости.*

Опр. 2.4 *Правилом φ_c называется закономерность, которая представляется в виде конъюнкции термов. Правилу всегда соответствует класс $c \in Y$, объекты именно этого класса выделяются правилом φ_c.*

Опр. 2.5 *Длиной правила называется количество термов в правиле.*

2.1.2 Эвристическое определение закономерности

Пусть $\varphi(x)$ — некий предикат. Введем обозначения:
G_c — число объектов класса c в выборке X^ℓ;
$g_c(\varphi)$ — из них число объектов, для которых выполняется $\varphi(x) = 1$;
B_c — число объектов всех остальных классов $Y \setminus \{c\}$ в выборке X^ℓ;
$b_c(\varphi)$ — из них число объектов, для которых выполняется $\varphi(x) = 1$.

Для краткости индекс c и аргумент φ будем иногда опускать. Предполагается, что $G \geqslant 1, B \geqslant 1, G + B = \ell$. Самыми информативнями будут закономерности, которые выделяют много объектов ($(b + g)$ – велико), и в тоже время среди выделенных мало принадлежащих «чужим» классам. Наоборот, не являются закономерностями те, которые выделяют мало объектов, или же выделяют их приблизительно в той же пропорции, что и на всей выборке ($b \div g \simeq B \div G$).

Введем обозначение E_c для доли ошибочно покрываемых «чужих» объектов и D_c для доли всех покрываемых объектов:

$$E_c(\varphi, X^\ell) = \frac{b_c(\varphi)}{b_c(\varphi) + g_c(\varphi)}, \quad D_c(\varphi, X^\ell) = \frac{b_c(\varphi) + g_c(\varphi)}{\ell}$$

Опр. 2.6 *Предикат $\varphi(x)$ будем называть (эвристической) закономерностью для класса $c \in Y$, если $E_c(\varphi, X^\ell) \leqslant \varepsilon$ и $D_c(\varphi, X^\ell) \geqslant \delta$ при заданных достаточно малого ε и достаточно большого δ из отрезка [0,1].*

Если $b_c = 0$, то закономерность φ называют *чистой* или *непротиворечивой*. Если $b_c > 0$, то закономерность φ называют *частичной*.

2.2 Критерии информативности

Часто данные оказываются неполными или неточными. Таковы многие медицинские и экономические задачи. Для них незначительная доля ошибок на обучающей выборке вполне допустима. Когда некоторая доля ошибок неизбежна, приходится пользоваться частичными закономерностями. В этих случаях предикаты φ отбираются по двум критериям $g_c(\varphi)$ и $b_c(\varphi)$ одновременно.

Будем рассматривать эвристики — двумерные функции вида $h(g_c(\varphi), b_c(\varphi))$.

Приведем некоторые примеры эвристик, основанных на линейных метриках. Оптимальной целью обучения является найти закономерность, которая бы покрывала все объекты класса c и ни одного объекта других классов. Цель эвристики, оценивающей закономерность — показать, насколько правило близко к идеальному [7, 8].

1. *Простейшие эвристики*

   Стратегией нахождения правила, покрывающего некоторые объекты своего класса и мало объектов других классов, является минимизация числа покрываемых объектов других классов:

   $$b_c(\varphi) \to \min_{\varphi}.$$

   Стремление покрыть много объектов своего класса выражается эвристикой:

   $$g_c(\varphi) \to \max_{\varphi}.$$

2. *Точность (accuracy)*

   Чтобы покрыть много объектов класса c и избежать покрытия многих объектов других классов, надо максимизировать эвристику:

   $$h_{acc} = g_c(\varphi) - b_c(\varphi) \to \max_{\varphi}.$$

   В [11] приводятся варианты эвристик:

   $$g_c(\varphi) - 2b_c(\varphi) \to \max_{\varphi}; \quad g_c(\varphi) - 5b_c(\varphi) \to \max_{\varphi}.$$

3. *Точность, взвещенная по классам (weighted relative accuracy)*

   Максимизация эвристики h_{acc} дает одинаковую значимость своим и чужим объектам. Нормализация приводит к максимизации эвристики:

   $$h_{wra} = \frac{g_c(\varphi)}{G_c} - \frac{b_c(\varphi)}{B_c} \to \max_{\varphi}.$$

Теорема 2.1 *Минимизация числа ошибок по предикату φ_c эквивалентна максимизации эвристики $h_{sqrt} = \sqrt{g_c} - \sqrt{b_c}$.*

Доказательство.

Будем рассматривать задачу классификации с двумя классами.

Введем функционал числа ошибок, допускаемых предикатом φ_c на обучающей выборке X^ℓ:

$$Q = \sum_{i=1}^{\ell} [\varphi_c(x_i)[y_i = c] - \varphi_c(x_i)[y_i \neq c] < 0].$$

Целью является минимизация функционала числа ошибок. Решим задачу приближенно. Заменим пороговую функцию непрерывно диференцируемой оценкой сверху. Выбор конкретной аппроксимирующей функции является эвристикой. Наиболее простые выкладки получаются, если воспользоваться оценкой $[z < 0] \leqslant e^{-\alpha z}$. Будем минимизировать функционал Q по параметру α.

Запишем верхнюю оценку $\widetilde{Q}$ функционала Q:

$$Q \leqslant \widetilde{Q} = \sum_{i=1}^{\ell} \exp\left(-\alpha\varphi_c(x_i)[y_i = c] + \alpha\varphi_c(x_i)[y_i \neq c]\right) =$$

$$= \sum_{i=1}^{\ell} \exp\left(-\alpha\varphi_c(x_i)[y_i = c]\right) \exp\left(\alpha\varphi_c(x_i)[y_i \neq c]\right).$$

Воспользуемся тождеством $e^{A\varphi} = (1-\varphi) + \varphi e^{A}$, которое справедливо для любых $A \in \mathbb{R}$ и $\varphi \in \{0,1\}$:

$$\widetilde{Q} = \sum_{i=1}^{\ell} \left(1 - \varphi_c(x_i)[y_i = c] + e^{-\alpha}\varphi_c(x_i)[y_i = c]\right)\left(1 - \varphi_c(x_i)[y_i \neq c] + e^{\alpha}\varphi_c(x_i)[y_i \neq c]\right).$$

Раскроем скобки, воспользуемся $[y_i = c][y_i \neq c] = 0$:

$$\widetilde{Q} = \underbrace{\sum_{i=1}^{\ell} 1}_{\ell} - \underbrace{\sum_{i=1}^{\ell} \varphi_c(x_i)[y_i = c]}_{g} - \underbrace{\sum_{i=1}^{\ell} \varphi_c(x_i)[y_i \neq c]}_{b} +$$
$$+ e^{\alpha} \underbrace{\sum_{i=1}^{\ell} \varphi_c(x_i)[y_i \neq c]}_{b} + e^{-\alpha} \underbrace{\sum_{i=1}^{\ell} \varphi_c(x_i)[y_i = c]}_{g} =$$
$$= \ell - g - b + e^{\alpha} b + e^{-\alpha} g \to \min.$$

Минимум этого выражения достигается при $e^{\alpha} b = e^{-\alpha} g$, отсюда вытекает $\alpha^* = \frac{1}{2}\ln\frac{g}{b}$, если только $b \neq 0$. Подставляя α^* в функционал $\widetilde{Q}$ получаем:

$$\widetilde{Q} = \ell - \left(\sqrt{g} - \sqrt{b}\right)^2.$$

Минимизация числа ошибок предиката эксивалентна максимизации эвристики $\sqrt{g} - \sqrt{b}$, что и требовалось доказать.

■

Эвристика $h_{sqrt} = \sqrt{g} - \sqrt{b}$ используется в алгоритмах бустинга. Она похожа на эвристику точность, также как и эвристика h_{acc} она не учитывает количество объектов каждого класса в выворке. Чтобы произвести балансировку по классам, можно предложить следующую эвристику:

$$h_{wsqrt} = \sqrt{\frac{g}{G}} - \sqrt{\frac{b}{B}}.$$

Часто для выбора лучшей закономерности φ_c класса c используется ROC анализ [8]. ROC анализ представляет собой визуализацию исследуемых закономерностей, их изображение на двумерных графиках зависимости $\frac{b}{B}$(вертикальная ось координат) от $\frac{g}{G}$(горизонтальная ось координат). Для закономерности φ_c: $\frac{g_c}{G_c}$ — доля правильно классифицированных объектов класса c, а $\frac{b_c}{B_c}$ — доля неправильно классифицированных объектов всех остальных классов. Пример ROC графика изображен на рис. 13.

На двумерных графиках идеальная закономерность находится в нижнем правом углу, в точке (1,0), точки (1,1) и (0,0) иллюстрируют закономерности, которые классифицируют все объекты либо к классу c, либо не покрывают ни одного «своего»(класса c) или «чужого»(всех остальных классов $Y \setminus \{c\}$) объекта. Если изобразить все закономерности на ROC графике, то лучшие будут ближе всего лежать к точке (1,0). Стремясь к точке (1,0) мы максимизируем эвристику h_{wra}.

2.3 Алгоритм поиска логических правил

Для количественных признаков $f\colon X \to \mathbb{R}$ будем искать термы в виде:

$$\beta(x) = [f(x) \leqslant \theta], \quad \theta \in D_f \tag{1}$$

Алгоритм 2.3. Построение множества порогов

Вход: обучающая выборка $X^\ell = (x_i, y_i)_{i=1}^\ell$, выделенный класс $c \in Y$, признаки $f_1, \ldots, f_m$;

Выход: $\Theta_j = \theta_j^1, \ldots, \theta_j^r, j = \overline{1,m}$ — строго возрастающие последовательности порогов для каждого признака $f_j, j = \overline{1,m}$.

1: **для** $j = 1, \ldots, m$
2: $\quad\Theta_j := \varnothing$;
3: **для** $j = 1, \ldots, m$
4: $\quad$Упорядочить по возрастанию значения признака f_j на объектах выборки X^ℓ, получить $f_j^{(1)}, \ldots f_j^{(\ell)}$;
5: $\quad$**для** $i = 1, \ldots, \ell$
6: $\quad\quad$**если** $f_j^{(i)} \neq f_j^{(i+1)}$ и $[y_i = c] \neq [y_{i+1} = c]$ **то**
7: $\quad\quad\quad$добавить новый порог $(f_j^{(i)} + f_j^{(i+1)})/2$ в конец последовательности Θ_j;

$$\beta(x) = [f(x) \geqslant \theta], \quad \theta \in D_f \tag{2}$$

Имеет смысл брать такие значения порогов θ, которые по-разному разделяют выборку X^ℓ. Если исключить тривиальные разбиения, обращающие $\varphi(x)$ в 0 или 1 на всей выборке, то таких значений останется не более $\ell - 1$. Можно взять пороги вида

$$\theta_i = \frac{f^{(i)} + f^{(i+1)}}{2}, \quad f^{(i)} \neq f^{(i+1)}, \tag{3}$$

где $f^{(1)}, \ldots f^{(\ell)}$ — последовательность значений признака f на объектах выборки $f(x_1), \ldots f(x_\ell)$, упорядоченная по возрастанию (вариационный ряд). Алгоритм 2.3 дает подробное описание процесса построения множества порогов.

Способ, описанный алгоритмом 2.3, позволяет получить огромное количество предикатов. Для признака f и порога θ можно построить два предиката вида (1) и (2). Для каждого правила производится настройка порогов признаков методом покоординатной оптимизации 2.3.

Правило φ_c класса c представимо в виде конъюнкции термов (1) или (2). Терм вида (1) или (2) полностью определяется парой ⟨значение порога, знак⟩. Чтобы построить правило φ_c надо задать набор признаков $f_1, \ldots, f_s$ и пару ⟨значение порога, знак⟩ для каждого признака.

Проиллюстрируем работу метода покоординатной оптимизации на примере настройки правила φ_c, зависящего от признаков $f_1, \ldots, f_s$. Для каждого признака $f_j, j = \overline{1,s}$ фиксируем начальные значения порогов $\theta_j^1 \in \Theta_j, j = \overline{1,s}$, применяем итерационный процесс, который заключается в том, что для $f_j, j = \overline{1,s}$ находим такое значение порога $\theta_j \in \Theta_j$ и знак из множества знаков $\{\leqslant, \geqslant\}$, при которых правило φ_c имеет максимальную информативность, например, $h_{wra} = \frac{g}{G} - \frac{b}{B}$. Процесс продолжается последовательно для каждого признака, пока выбираемые пороги и знаки не перестанут меняться.

2.4 Отбор наиболее информативных правил

Для разных целей могут быть подходящими разные правила. Для нахождения одного лучшего по фиксированному критерию правилу достаточно выбрать правило, на котором достигается экстремум (максимум или минимум, в зависимости от задачи) критерия.

Если нас интересует набор правил, то правила каждого класса можно изобразить на ROC графиках и выбирать правила, оптимальные по Парето.

Алгоритм 2.3. Метод покоординатной оптимизации. Функция GetRules

Вход: обучающая выборка $X^\ell = (x_i, y_i)_{i=1}^\ell$, класс $c \in Y$, признаки $f_1, \ldots, f_s$, последовательности порогов $\{\Theta_j\}_{j=1}^s$;

Выход: для каждого признака $f_j, j = \overline{1,s}$ найти пару (θ_j,sign) — оптимальные порог и знак.

1: **для** $j = 1, \ldots, s$
2: $\quad\theta_j := \theta_j^1$;
3: $\quad$sign:=$\geqslant$;
4: **пока** хотя бы один порог или знак изменился
5: $\quad$**для** $j = 1, \ldots, s$
6: $\quad\quad$**для** $\theta_j \in \Theta_j$
7: $\quad\quad\quad$**для** sign$\in \{\leqslant, \geqslant\}$
8: $\quad\quad\quad\quad$ θ_j и знак sign, такие что правила, представимые в виде конъюнкции термов (1) или (2) максимизируют эвристику, выбранную для конкретной задачи, например, $h_{wra} = \frac{g}{G} - \frac{b}{B}$;

Алгоритм 2.4. Метод отбора наиболее информативных правил.

Вход: набор логических правил B, глубина окрестности d;

Выход: построить окрестность оптимальных по Парето правил B'.

1: $B' = \varnothing$;
2: **для** $i = 1, \ldots, d$
3: $\quad B'$ — оптимальные по Парето правила из множества правил B;
4: $\quad B = B \setminus B'$;

Опр. 2.7 *Правило φ называется оптимальным по Парето , если не существует правила φ', такого, что выполняются одновременно два условия:*

$$\frac{g(\varphi')}{G(\varphi')} \geqslant \frac{b(\varphi)}{B(\varphi)}$$

$$\frac{g(\varphi')}{G(\varphi')} \leqslant \frac{b(\varphi)}{B(\varphi)}$$

и хотя бы одно неравенство выполнено как строгое, т.е. $\varphi \neq \varphi'$

Иными словами, правила неоптимальные по Парето являются хуже оптимальных хотя бы по одному из критериев: либо доля покрываемых ими «чужих» объектов больше, либо доля покрываемых ими «своих» объектов меньше.

Чтобы расширить набор правил, можно выбирать некоторую «окрестность» оптимальных по Парето правил. Итерационно для набора правил строятся оптимальные по Парето, потом они исключаются из набора и происходит переход на следующую итерацию. Количество итераций называется глубиной окрестности и является входным параметром алгоритма. Подробное описание процесса построения «окрестности» оптимальных по Парето правил — алгоритм 2.4. Изображение окрестности Парето оптимальных правил на рис. 17. Использование окрестности оптимальных правил позволяет покрывать большее количество объектов, рассматривать больший набор правил, использовать правила, близкие к оптимальным.

Алгоритм 2.5.1. Алгоритм построения решающего списка

Вход: обучающая выборка $X^\ell = (x_i, y_i)_{i=1}^\ell$, множество классов Y, желаемая длина решающего списка p, $\{h \to \max\}$ — критерий качества правила, выбираемый под конкретную задачу;

Выход: $\mathrm{DL} = \{(\beta_1, c_1), (\beta_2, c_2), \ldots, (\beta_p, c_p), c_0\}$ — решающий список.

1: Инициализировать решающий список $\mathrm{DL} = \varnothing$;
2: Инициализировать множество объектов обучения $U = X^\ell$;
3: **пока** $\mathrm{DL} < p$
4: $\quad \Phi = \mathsf{GetRules(U)}$;
5: $\quad$ **если** $\Phi = \varnothing$ **то**
6: $\quad\quad$ Вернуть DL
7: $\quad \beta = \max(h)$;
8: $\quad$ Добавить β в DL;
9: $\quad$ Удалить из U объекты, покрытые правилом β: $U = \{x \colon \beta(x) = 0\}$
10: Вернуть DL

2.5 Алгоритмы классификации

Обычно трудно найти закономерность, которая полностью отделит один класс от другого, поэтому для построения алгоритмов классификации используют композицию закономерностей.

2.5.1 Решающий список

Опр. 2.8 *Решающий список задается последовательностью правил $\varphi_1(x), \ldots, \varphi_p(x)$, с соответствующими им ответами $c_1, \ldots, c_p, c_0$. Получив на вход объект x, алгоритм проверяет правила последовательно, и как только находится правило $\varphi_i(x) = 1$, возвращает ответ c_i. Если ни одно из правил не смогло классифицировать объект, то возвращается ответ c_0, означающий отказ от классификации объекта x.*

На практике объекты, которые не смогло покрыть ни одно правило, приписываются классу, имеющему минимальную цену ошибки.

Рассмотрим первое и второе правило в решающем списке. Первое правило должно выделять объекты класса c_0, тогда это правило будет «хорошим», если покроет много объектов класса c_0 и мало объектов класса c_1. Второе правило решающего списка выделяет объекты класса c_1. Аналогично, оно будет «хорошим», если покроет много объектов класса c_1 и мало объектов класса c_0, но ничто не мешает ему покрывать объекты, уже покрытые первым правилом. Получается, что если $c_0 \neq c_1$, то второе правило может не являться «хорошей» закономерностью само по себе. Возникает вопрос о выборе стратегии установления последовательности классов при создании решающего списка. Существует два простейших варианта.

Первый вариант — сначала строятся все правила первого класса, потом все правила второго и.т.д. Классы берутся в порядке убывания важности или цены ошибки.

Второй вариант — выбирать предикат, у которого информативность максимальна. Тогда правила различных классов могут следовать вперемежку.

Достоинствами решающих списков являются: интерпретируемость и простота классификации, возможность обработки разнотипных данных и данных с пропусками.

Среди недостатков можно отметить то, что алгоритм требует больших вычислительных затрат, т.к. для построения каждого правила φ_i необходимо заново строить весь набор правил и выбирать из него наилучшее. Если множество Φ выбрано

Алгоритм 2.5.2. Алгоритм синтеза правил для голосования по большинству

Вход: обучающая выборка $X^\ell = (x_i, y_i)_{i=1}^\ell$, множество классов Y;
Выход: $\text{Vote} = \{(\varphi_1, c_1), (\varphi_2, c_2), \ldots, (\varphi_p, c_p)\}$ — список голосования по большинству.

```
1: Инициализировать список Vote = ∅;
2: Инициализировать множество объектов обучения U = X^ℓ;
3: Φ = GetRules(U);
4: если Φ = ∅ то
5:    Вернуть Vote;
6: для φ ∈ Φ
7:    если φ оптимальная закономерность то
8:       Добавить φ в Vote;
```

не удачно, то алгоритм может не построиться. Каждый объект классифицируется только одним правилом, что не позволяет правилам компенсировать неточности друг друга.

Чтобы избежать больших вычислительных затрат можно строить решающий список последовательно из всех оптимальных по Парето правил. Если оптимальные правила не покрыли всех объектов, тогда переходим к следующей итерации: генерируем набор правил, выбираем из них оптимальные по Парето, последовательно включаем их в решающий список.

2.5.2 Голосование по большинству

В качестве альтернативы алгоритма решающий список предлагается рассмотреть алгоритм голосования по большинству.

Строится весь набор логических правил. Для правил каждого класса строятся ROC графики. На графиках по горизонтальной оси отложено количество покрытых «своих» объектов ($\frac{g}{G}$), по вертикальной оси — количество покрытых «чужих» объектов ($\frac{b}{B}$).

По ROC графикам каждого класса выбираются оптимальные закономерности (можно выбирать правила оптимальные по Парето или лежащие в окрестности оптимальных по Парето).

Опр. 2.9 *Голосование по большинству задается набором правил $\varphi_1(x), \ldots, \varphi_p(x)$ и соответствующими им классами $c_1, \ldots, c_p$. Решение о принадлежности объекта x какому-либо классу принимается так:*

$$a(x) = \arg\max_{y \in Y} \frac{1}{N} \sum_{i=1}^{p} \varphi_i(x)[c_i = y],$$

где $N = \sum_{i=1}^{p} [c_i = y]$ — количество правил класса $y \in Y$.

Алгоритм голосования по большинству прост и легко интерпретируемый. Он не требует больших вычислительных затрат, т.к. самая трудоемкая операция — генерация набора правил, выполняется один раз. Решение о принадлежности объекта классу «принимается» всеми правилами, покрывающими этот объект, таким образом, недостатки одних правил могут быть скомпенсированы другими правилами.

2.6 Оценивание и выбор моделей

Задача выбора моделей состоит в следующем. Имеется T моделей алгоритмов. Возникает вопрос: какой алгоритм выбрать?

В типичных случаях эта задача возникает, когда априорные предпочтения для использования какой-либо модели отсутствует, а хочется выбрать ту модель, которая обладает лучшей обобщающей способностью.

Пусть задано конечное множество $X^L = \{x_1, \ldots, x_L\}$, называемое *полной* или *генеральной* выборкой, множество классов Y и множество алгоритмов $A\colon X^L \to Y$. Допустим, что для любого объекта $x \in X^L$ и для любого алгоритма $a \in A$ можно сказать является ли ответ $a(x)$ ошибочным, т.е. существует бинарная функция

$$I(a, X) = [a(x) \neq y^*(x)],$$

называемая *индикатором ошибки*. Если $I(a, x) = 1$, то говорят, что алгоритм a допускает ошибку на объекте x.

Числом ошибок алгоритма a на выборке $X \subseteq X^L$ называется величина

$$n(a, X) = \sum_{x \in X} I(a, x).$$

Частотой ошибок или *эмпирическим риском* алгоритма a на выборке X называется величина

$$\nu(a, X) = \frac{1}{|X|} n(a, X).$$

Она принимает значения из отрезка $[0, 1]$.

Опр. 2.10 *Методом обучения называется отображение $\mu\colon X^\ell \to a$, которое конечной выборке $X^\ell \in X^L$ ставит в соответствие алгоритм $a \in A$.*

Выбор модели производится по одному из следующих критериев.

2.6.1 Критерий средней ошибки на контрольных данных

Будем считать, что полная выборка $X^L = \{x_1, \ldots, x_L\}$ некоторым образом разделена на обучающую и контрольную части $X^L = X^\ell \cup X^k$. Критерий средней ошибки метода обучения μ на контрольных данных:

$$\text{HoldOut}(\mu, X^L) = \nu(\mu(X^\ell), X^k).$$

В англоязычной литературе этот критерий называют *ошибкой на отложенных данных* (hold-out error).

На практике полную выборку данных X^L разбивают на обучение и контроль случайным образом. В контрольной выборке, как правило, оставляют от четверти до половины объектов.

2.6.2 Критерий скользящего контроля

Этот критерий является обобщением предыдущего. Чтобы результат не зависел от способа разбиения, берут несколько различных разбиений исходной выборки X^L на обучение и контроль $X^L = X^\ell_n \cup X^k_n, n = 1, \ldots, N$, и среднюю ошибку на контроле усредняют по разбиениям. Этот функционал называется *ошибкой скользящего контроля* (cross validation error CV):

$$\text{CV}(\mu, X^L) = \frac{1}{N} \sum_{n=1}^{N} \nu(\mu(X^\ell_n), X^k_n).$$

В зависимости от способа формирования разбиений различают несколько видов скользящего контроля.

Полный скользящий контроль (complete CV) строится по всем $N = C_L^k$ разбиениям. Это число становится слишком большим уже при $k > 2$, поэтому на практике чаще используют другие виды скользящего контроля.

Контроль по отдельным объектам (leave-one-out CV) является частным случаем полного скользящего контроля при $k = 1$:

$$\text{LOO}(\mu, X^L) = \frac{1}{L}\sum_{i=1}^{L} \nu(\mu(\mathbb{X} \setminus \{x_i\}), \{x_i\}).$$

Преимущества LOO в том, что каждый объект ровно один раз участвует в контроле, а длина обучающих выборок всего на единицу меньше длины полной выборки. Недостатком LOO является большая ресурсоемкость, т.к. обучаться приходится L раз.

Контроль по q блокам (q-fold CV). Выборка разбивается на q одинаковых (или почти одинаковых) блоков длины $l_1, \ldots, l_q$:

$$X^L = X_1^{l_1} \cup \cdots \cup X_q^{l_q}, l_1 + \cdots + l_q = L.$$

Каждый блок по очереди становится контрольной подвыборкой, при этом обучение проводится по остальным $q - 1$ блокам. Критерий определяется как средняя по всем блокам ошибка на контроле:

$$\text{Qfold}(\mu, X^L) = \frac{1}{q}\sum_{n=1}^{q} \nu(\mu(X^L \setminus X_n^{l_n}), X_n^{l_n}).$$

Это компромисс между LOO и hold-out. С одной стороны, обучение проводится q раз вместо L. С другой стороны, длина обучающих подвыборок $L\frac{q-1}{q}$ (с точностью до округления) не сильно отличается от длины полной выборки.

Если ввести дополнительное требование разбивать выборку на непересекающиеся блоки, такие что любой объект входит в единственный блок, то каждый объект $q - 1$ раз попадет в обучение и 1 раз в контроль. Тогда можно в явном виде посчитать число ошибок классификации на контроле и на обучении:

$$\text{Test} = \sum_{n=1}^{q} n(\mu(X^L \setminus X_n^{l_n}), X_n^{l_n}));$$

$$\text{Train} = n(\mu(X^L), X^L).$$

При решении практических задач восстановления зависимостей приходится регулярно сталкиваться с проблемой *переобучения*, когда алгоритм выдает правильные ответы на обучении и демонстрирует низкое качество на новых объектах, не входивших в состав обучения.

Этот эффект принято связывать с избыточной сложностью алгоритма. Чем больше у алгоритма свободных параметров, тем меньшего числа ошибок можно добиться путем оптимизации. Однако по мере нарастания сложности модели «оптимальные» алгоритмы начинают слишком хорошо подстраиваться под конкретные данные, улавливая не только черты восстанавливаемой зависимости, но и ошибки измерения обучающей выборки, и погрешность самой модели.

Оценить эффект переобучения можно, сравнивая число ошибок на контроле и обучении. Если число ошибок на контроле в несколько раз больше числа ошибок на обучении, то алгоритм слишком сильно подстроился под данные, переобучение велико.

3 Прогнозирование отдаленных результатов хирургических операций

3.1 Описание предметной области

Проблема облитерирующего атеросклероза артерий считается одной из наиболее важных в современной медицинской науке [13]. В возрасте 50 лет в той или иной степени от этого заболевания страдают до 90% населения, а в возрасте свыше 60 лет — 100%. В структуре смертности атеросклероз и его проявления занимают основное место, являясь причиной 80% летальных исходов. Тем не менее, до сих пор не выявлены ключевые механизмы возникновения атеросклероза.

Сужение и тромбоз артерий нижних конечностей в большинстве случаев не удается вылечить без операции. Хирургическое вмешательство заключается в установке шунта в артерию, но и это не всегда приносит положительный эффект, т.к. иногда происходит повторное сужение артерии на том же или другом уровне. После операции болезнь может прогрессировать и приводить к инвалидности или смерти. От чего зависит результат операции? Можно ли, зная набор показателей каждого больного до операции, предсказать отдаленный результат операции? В данной работе эта задача ставится и решается как задача обучения классификации. Каждый больной рассматривается в качестве прецедента в задаче классификации, измеренный набор показателей выступает в роли признакового описания. Обучающую выборку составляет группа больных, которые наблюдались длительное время, и для каждого был определен исход операции: положительный или отрицательный (отрицательным исходом является повторное сужение сосудов).

3.2 Особенности медицинских данных

Исходные данные медицинской задачи — выборка из 115 прецедентов, накопленных в клинике факультетской хирургии РГМУ. Они представляют собой совокупность гемодинамических и иммунологических показателей, всего 178 признаков, измеренных у больного до операции и через некоторые периоды времени после операции. Все признаки кроме целевого являются количественными, а целевой признак — номинальный. Целевым признаком являлся отдаленный результат операции: 0 — успешная операция, 1 — плохой результат (повторное сужение сосудов), 2 — смерть от других причин.

Данные собирались экспертами в два этапа: на первом этапе была собрана информация о 72 больных (будем называть эти данные старыми), на втором этапе данные дополнялись новыми признаками и объектами. Старые данные содержат меньше 5% процентов пропусков, являются более аккуратно собранными. Основная задача состоит в классификации всего набора объектов. Но при проведении исследования будем применять алгоритмы классификации и к старым данным.

Опишем подробнее структуру данных задачи. Признаки задачи делятся на две качественные группы: иммунологические и гемодинамические признаки, и 5 групп по времени наблюдения: до операции, непосредственно после операции, спустя неделю, спустя месяц, спустя 2 месяца.

Чем дальше от операции происходит измерение, тем у меньшего количества больных оно производится. Прогноз отдаленного результата операции важен до операции, т.е. прогноз строиться по всем объектам с признаками, измеренными до операции. Если предсказывается неблагоприятный исход для больного, то, возможно, его надо лучше готовить к операции. Сразу после операции прогноз тоже имеет смысл делать для того, чтобы интенсивнее наблюдать за больными, склонными к неблагоприятному исходу. Чем дальше момент прогнозирования удален от операции (неделя, месяц, 2 месяца после операции), тем более точен может быть прогноз, но менее полезен прогноз с медицинской точки зрения.

Признаковое описание задачи разделено на две группы: иммунологические и гемодинамические признаки. Отклонения от нормы гемодинамических и иммунологических показателей определяют разные врачебные тактики. Эксперты склоняются к несмешиванию двух групп признаков в одном логическом правиле. Соответственно, правила надо строить по всем признакам, но каждое правило должно содержать признаки только одной группы.

Классы задачи (успешная операция, плохой отдаленный результат) не являются равнозначными. Приписать здоровому больному отрицательный исход операции не так страшно, как отнести человека с отрицательным отдаленным исходом к классу больных, у которых операция прошла успешно. При настройке алгоритма классификации первостепенной задачей является не минимизация суммарного процента ошибок классификации, а уменьшение процента пропущенных отрицательных исходов: количества больных с отдаленным отрицательным исходом, которых алгоритм отнес к положительному классу.

Данная задача классификации характеризуются большой размерностью (число признаков 178, число объектов 115), наличием пропусков в данных (более 30% значений признаков не заполнены), неточностью измерения значений признаков, небольшой длиной выборки, маленьким количеством объектов с плохим результатом операции (18 прецедентов из 115 принадлежат классу 1). Специфика рассматриваемой задачи накладывает ограничения на применимые алгоритмы, заставляет модифицировать некоторые стандартные методы логических алгоритмов.

3.3 Предварительная обработка данных

Были предоставлены данные о 115 больных. Признаковое описание больных было неоднородно и содержало большое число пропусков. Общее число признаков составляло 178. Данные требовали предобработки.

Были исключены прецеденты, у которых отсутствовала информация об исходе или была указана «смерть от других причин».

Для классификации использовались данные, отражающие состояние больного непосредственно до операции и сразу после операции. В результате фильтрации были удалены признаки, отражающие состояние больного позднее 11 дня после операции. Также из задачи исключались признаки, измеренные только у объектов одного класса. Логические правила, построенные на таких признаках заведомо не будут покрывать объекты «чужих» классов и будут отражать лишь особенности измерения данных.

В выборке оказались объекты, у которых было заполнено менее двух (возраст и исход) из выбранных 73 признаков. Такие прецеденты были удалены из выборки.

В процессе анализа данных также были обнаружены несоответствия в размерностях гемодинамических данных, измеренных на разных аппаратах. Это было вызвано различной калибровкой измерительных приборов. Совместно с врачами-экспертами абсолютные значения гемодинамических признаков были приведены к относитель-

ным. Если в исходных данных были даны показатели скоростей крови по вене до и после операции, то из них получали процентное изменение скорости крови. Переход к относительным показателям исключил найденные несоответствия.

Проведенные преобразования данных:

1. Удаление объектов с неизвестным значением целевого признака.
2. Удаление объектов с исходом «смерть от других причин».
3. Удаление признаков, измеренных позднее 11 дня после операции.
4. Удаление признаков, измеренных у объектов одного класса.
5. Удаление объектов с числом пропусков более 98%.
6. Модификация гемодинамической группы признаков.

Модифицированная выборка была представлена в виде матрицы 100×52 с 20% пропусков данных. Из 52 признаков иммунологической группе принадлежат 22, а гемодинамической — 29, признак возраст не принадлежит группам. Охарактеризуем каждую из групп признаков и дадим краткое описание признаков задачи.

1. AGE — Возраст.
2. ИСХОД — Отдаленный результат операции: 0 — хороший, 1 — плохой.

   Иммунологические признаки характеризуют защитную систему организма, которая участвует в поддержании постоянства внутренней среды и борьбе с внешними агрессивными воздействиями.

   Изменение уровня Т киллеров и других CD показателей свидетельсвует о их защитном воздействии на сосудистую стенку.
3. CD8 ДО — Т киллеры до операции.
4. CD8 ПО — Т киллеры после операции.
5. CD18 ДО — В лимфоциты до операции.
6. CD18 ПО — В лимфоциты после операции.
7. CD11b ДО — Молекулы адгезии до операции.
8. CD11b ПО — Молекулы адгезии после операции.

   Повышение нейтрофилов происходит в ответ на воспаление в области шунта и борьбу с ним.
9. ФИН ДО — Нейтрофилы до операции.
10. ФИН ПО — Нейтрофилы после операции.

    Увеличение моноцитов и ЦИК показывает общий характер ответа организма на операционную травму
11. ФИМ ДО — Моноциты до операции.
12. ФИМ ПО — Моноциты после операции.

13. ЦИК3 ДО — Циркулирующие иммунные комплексы 3% до операции.
14. ЦИК3 ПО — Циркулирующие иммунные комплексы 3% после операции.
15. ЦИК4 ДО — Циркулирующие иммунные комплексы 4% до операции.
16. ЦИК4 ПО — Циркулирующие иммунные комплексы 4% после операции.

    Повышенная концентрация ФНО указывает на усиление воспалительных и иммунных процессов. ФНО, ИФ, ТФР, ИЛ8 обеспечивают мобилизацию организма при возпалительном процессе.

17. ФНО ДО — Фактор некроза опухоли до операции.
18. ФНО ПО — Фактор некроза опухоли после операции.
19. ИФ ДО — Интерферон гамма до операции.
20. ИФ ПО — Интерферон гамма после операции.
21. ТФР ДО — Транс фактор роста до операции.
22. ТФР ПО — Транс фактор роста после операции.
23. ИЛ8 ДО — Интерлейкин 8 до операции.
24. ИЛ8 ПО — Интерлейкин 8 после операции.

Гемодинамические признаки характеризуют скорость кровотока в венозной и артриальной системе конечности. По изменению этих данных можно судить о проходимости сосудов или их сужении. Контроль данных показателей в динамике (до и после операции) является объективным признаков эффективности вмешательства.

25. VАДО — соотношение максимальной скорости кровотока до нагрузки до операции с максимальной скоростью кровотока после нагрузки до операции.
26. VАПО — соотношение максимальной скорости кровотока до и после нагрузки после операции.
27. VA=(VАПО-VАДО)*100%/ VАПО.
28. VАДН — соотношение максимальной скорости кровотока до и после операции до нагрузки.
29. VАПН — соотношение максимальной скорости кровотока до и после операции после нагрузки.
30. VIДО — соотношение минимальной скорости кровотока до и после нагрузки до операции.
31. VIПО — соотношение минимальной скорости кровотока до и после нагрузки после операции.
32. VI=(VIПО-VIДО)*100%/ VIПО.
33. VIДН — соотношение минимальной скорости кровотока до и после операции до нагрузки.

34. VIIIH — соотношение минимальной скорости кровотока до и после операции после нагрузки.

35. ТДО — соотношение средней скорости кровотока до и после нагрузки до операции.

36. ТПО — соотношение средней скорости кровотока до и после нагрузки после операции.

37. Т=(ТПО-ТДО)*100%/ТПО.

38. ТДН — соотношение средней скорости кровотока до и после операции до нагрузки.

39. ТПН — соотношение средней скорости кровотока до и после операции после нагрузки.

40. RДО — соотношение индекса резистивности вены до и после нагрузки до операции.

41. RПО — соотношение индекса резистивности вены до и после нагрузки после операции.

42. R=(RПО-RДО)*100%/RПО.

43. RДН — соотношение индекса резистивности вены до и после операции до нагрузки.

44. RПН — соотношение индекса резистивности вены до и после операции до нагрузки.

45. РДО — соотношение индекса пульсивности вены до и после нагрузки до операции.

46. РПО — соотношение индекса пульсивности вены до и после нагрузки после операции.

47. Р=(РПО-РДО)*100%/РПО.

48. РДН — соотношение индекса пульсивности вены до и после операции до нагрузки.

49. РПН — соотношение индекса пульсивности вены до и после операции до нагрузки.

50. ПРОКСПРО — скорость кровотока по оперированной артерии выше начала шунта.

51. ПРОКСДИС — скорость кровотока по оперированной артерии ниже начала шунта.

52. ДИСТПРО — скорость кровотока по оперированной артерии выше конца шунта.

53. ДИСТДИСТ — скорость кровотока по оперированной артерии ниже конца шунта.

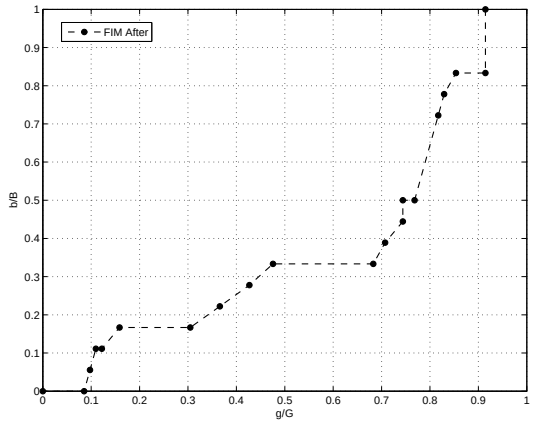

Рис. 1: Признак ФИМ ПО

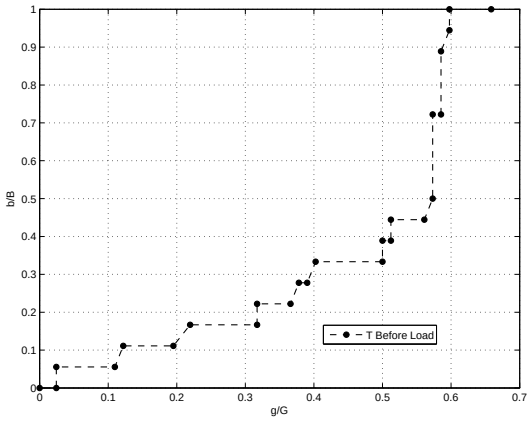

Рис. 2: Признак ТДН

3.4 Одномерный анализ признаков

Данные задачи прогнозирования отдаленного исхода хирургических операций состоят из 100 объектов и 52 признаков. Признаки выбирались врачами-экспертами, возможно, некоторые из них не влияют на исход операции или влияют в незначительной степени, от других, наоборот, целевой признак зависит очень сильно.

Проведем одномерный анализ признаков. Попробуем ответить на вопрос: по каким признакам можно отделить объекты класса 0 от объектов класса 1 наиболее точно?

Для каждого признака f_j с помощью алгоритма 2.3 строится набор порогов $\Theta_j = \{\theta_j^1, \ldots, \theta_j^r\}$. Порог $\theta_j^i, i = \overline{1,r}$ порождает правила вида

$$\varphi(x) = [f_i \leqslant \theta_j^i]$$

$$\varphi(x) = [f_i \geqslant \theta_j^i].$$

Рассмотрим правила класса 0. Для правил вычислим g — количество покрытых «своих» объектов, b — количество покрытых «чужих» объектов. Правила, построенные для фиксированного признака и знака ($\leqslant$ или $\geqslant$), изобразим на одном ROC графике. ROC графики правил, построенных для одного и того же признака, но разных знаков, будут симметричны относительно диагонали. Чем ближе правила к «идеальной» закономерности, находящейся в точке (1,0) графика, тем точнее можно разделить классы.

На ROC графиках 1–6 точки обозначают правила, построенные для фиксированного признака при варьировании порога. По горизонтальной оси отложена доля правильно классифицированных объектов класса 0, по вертикальной — доля неправильно классифицированных. Рисунки 1–4 иллюстрируют хорошую разделимость классов по признаку, правила больше приближены к точке (1,0), чем на рисунках 5–6, иллюстрирующих плохую разделимость.

Одномерный анализ признаков показал, что лучше разделяют классы признаки, измеренные после операции. Правила, построенные на этих признаках покрывают больше объектов «своего» класса, чем «чужого», лежат на ROC графиках ниже диагонали. См. рис. 1–4.

Значения признаков, измеренных до операции, размыты. Соотношение покрытых правилами «чужих» и «своих» объектов практически одинаково, правила, изображенные на ROC гафиках лежат близко к диагонали. См. рис. 5–6.

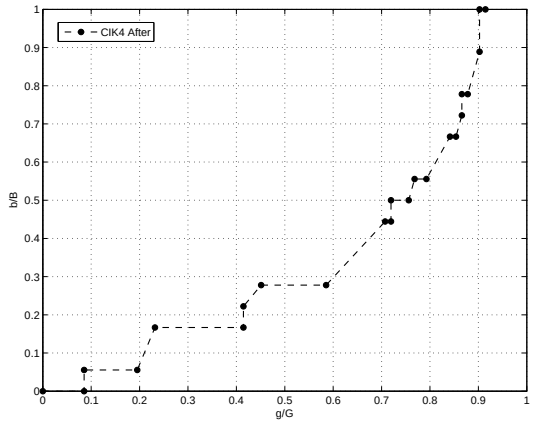

Рис. 3: Признак ЦИК4 ПО

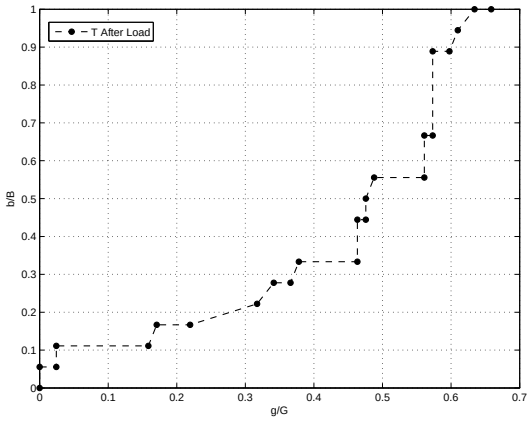

Рис. 4: Признак ТПН

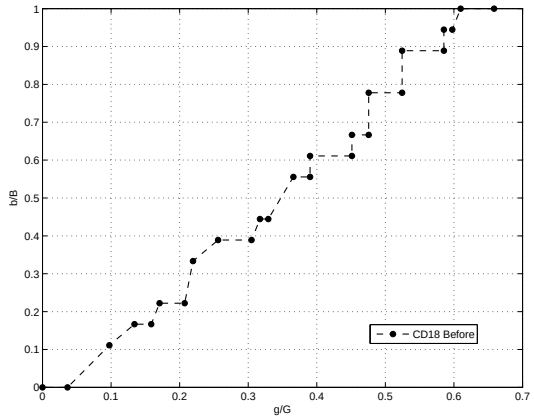

Рис. 5: Признак CD8 ДО

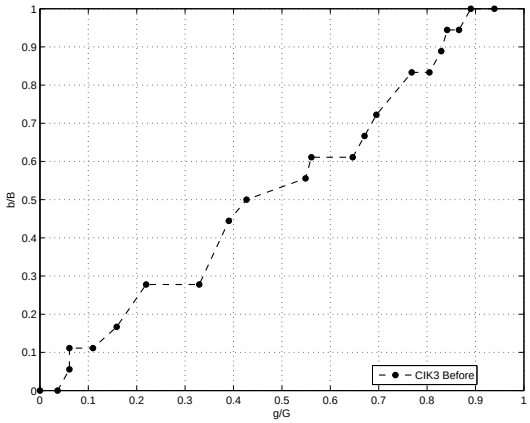

Рис. 6: Признак ЦИК3 ДО

3.5 Линейная классификация. Информативность признаков и объектов

Одномерный анализ данных показал низкое качество классификации правилами длины один. Нельзя предсказать исход операции, опираясь на какой-то один признак. Надо использовать наборы признаков, выявлять верные сочетания и закономерности. Для анализа информативности сочетаний признаков использовались линейные классификаторы.

Для всевозможных сочетаний признаков по одному, по два, по три, по четыре строились линейные разделяющие гиперплоскости. Размер перебора был ограничен сверху допустимым временем работы алгоритма построения линейных классификаторов. Классификаторы, построенные на двух признаках будем называть двумерными, на трех — трехмерными и.т.д.

Линейные классификаторы характеризуются количеством правильно классифицированных объектов класса 0 (положительный исход операции) и класса 1 (отрицательный исход), количеством ошибок классификации для класса 0 и 1. По аналогии с логическими закономерностями обозначим через G — количество объектов класса 0, B — количество объектов класса 1, g – количество правильно классифицированных объектов класса 0, b — количество объектов класса 1, которых классификатор ошибочно отнес к классу 0. Тогда количество правильно классифицированных

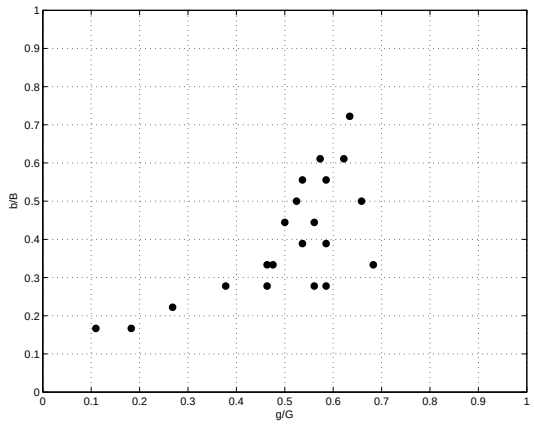

Рис. 7: Оптимальные одномерные классификаторы

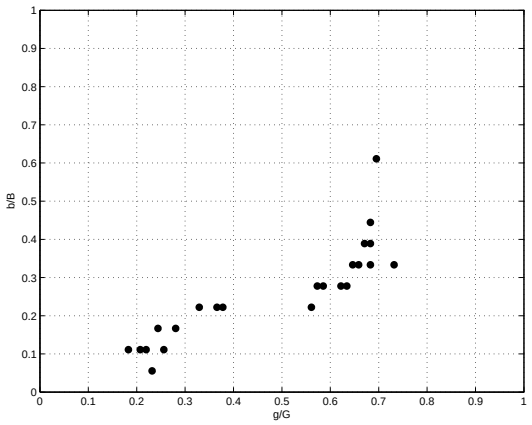

Рис. 8: Оптимальные двумерные классификаторы

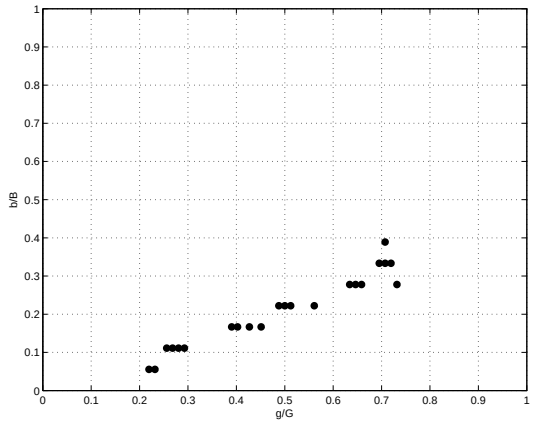

Рис. 9: Оптимальные трехмерные классификаторы

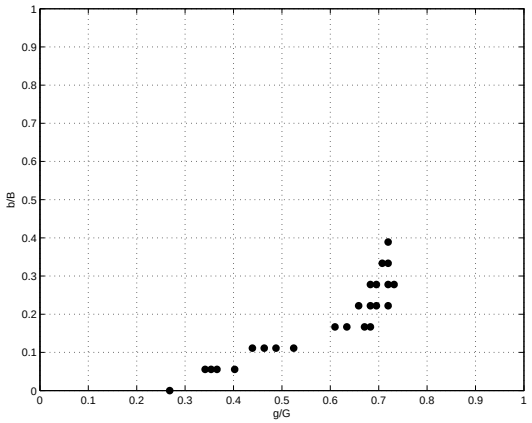

Рис. 10: Оптимальные четырехмерные классификаторы

объектов класса 1 выражается через введенные величины, как $B - b$, а количество неправильно классифицированных объектов класса 0 равно $G - g$. Следовательно, оценивая качество классификатора можем опираться только на значения величин g и b. Максимизируя g, мы уменьшаем количество ошибок на объектах класса 0, а минимизируя b, увеличиваем количество правильно классифицированных объектов класса 1. Для линейных классификаторов можно строить ROC графики зависимости $\frac{b}{B}$ от $\frac{g}{G}$, анализировать графики, выбирать оптимальные по Парето классификаторы, которые будут обладать наилучшим качеством.

На графиках 7–10 изображены окрестности оптимальных по Парето классификаторов (рассматривается окрестность грубины 4). Чем больше размерность пространства, в котором строились классификаторы, тем большей точности можно достичь. Классификаторы размерности четыре на ROC графиках лежат ближе к точке (1,0).

Изобразим классификаторы размерностей 1–4 на одном ROC графике и выберем из них те, которые лежат в окрестности оптимальных по Парето правил, для краткости назовем их просто оптимальными. Для каждого признака вычислим количество классификаторов, в которые он входит. Таким образом, можно определить значимость, информативность признака: информативность признака — это число оп-

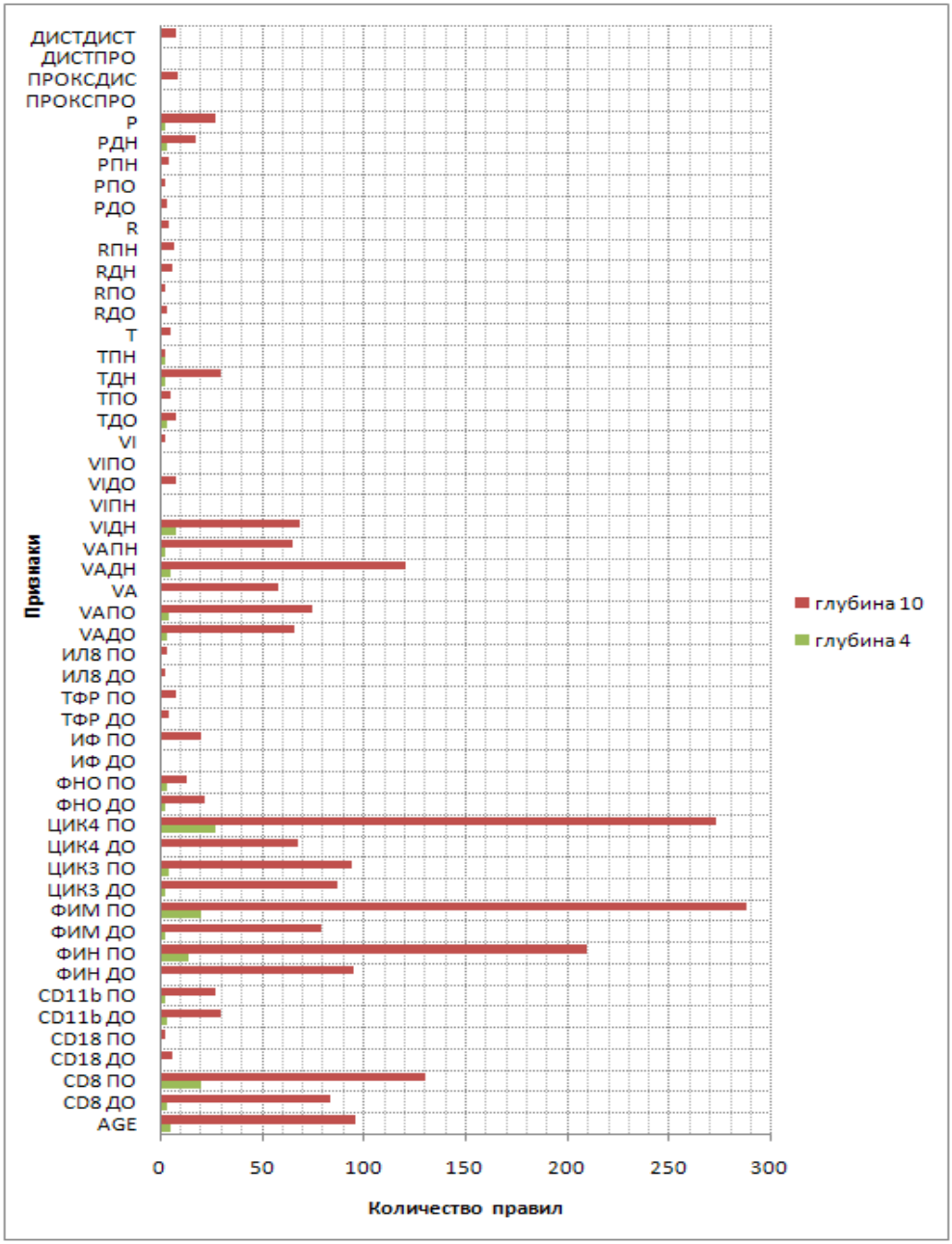

Рис. 11: Информативность признаков

тимальных классификаторов, в которые этот признак входит. Рисунок 11 иллюстрирует значимость признаков, по вертикальной оси перечислены признаки, по горизонтальной отложено количество оптимальных классификаторов, в которые входит признак. Величина значимости признаков зависит от выбора оптимальных правил. Для сравнения рассматривались окрестности оптимальных правил глубины 4 и 10. Анализ графика показал, что признаки, измеренные после операции более информативны, информативность иммунологической группы признаков выше, чем гемодинамической, в гемодинамической группе наиболее значимы показатели скорости кровотока по вене.

Информативность объектов задачи также оценивалась по оптимальным классификаторам. Для каждого объекта вычислялось количество неверных классификаций и количество раз, которое данный объект был классифицирован, бралось отношение этих величин. Оно означало долю ошибочных классификаций объекта. Объекты, для которых это число ошибок велико, являются объектами-выбросами. На рис. 12

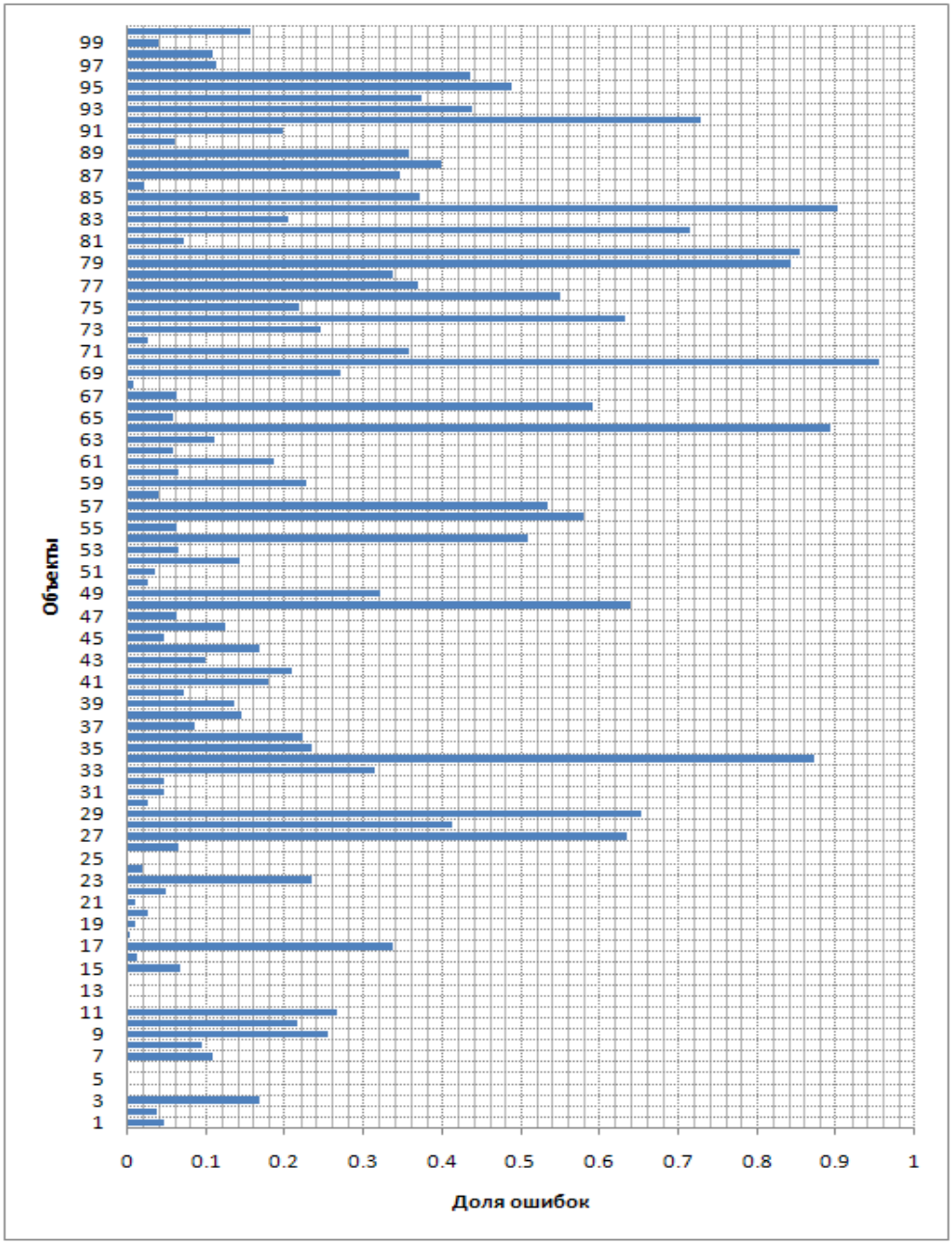

Рис. 12: Ошибки классификации для объектов задачи

изображена доля ошибочных классификаций для каждого объекта, по вертикальной оси — объекты, по горизонтальной — взвешенное количество ошибок, доля ошибок. Объекты на графике отсортированы по принадлежности классам: первые 82 объекта относяться к классу 0, последние 18 объектов к классу 1.

Производить фильтрацию объектов-выбросов можно, устанавливая разные пороги допустимой доли ошибок. В данной задаче будем считать выбросами объекты, у которых доля ошибок больше 0.8 (6 объектов, среди них один объект класса 1).

3.6 Применение стандартных алгоритмов классификации

Для решения задачи была выбрана библиотека алгоритмов машинного обучения WEKA(Waikato Environment for Knowledge Analysis). WEKA — это открытый программный продукт. Программное обеспечение написано целиком на языке Java. WEKA предоставляет прямой доступ к библиотеке реализованных в ней алгорит-

мов. Исходные данные должны быть представлены в виде матрицы признаковых описаний объектов.

С помощью системы WEKA можно легко провести классификацию имеющихся данных, оценить качество с помощью скользящего контроля, присвоить веса объектам, выбрать лучший алгоритм или лучшее семейство алгоритмов классификации.

Задача прогнозирования отдаленного исхода операций была разделена на несколько подзадач:

1. данные до и после операции, все признаки
2. данные до и после операции, иммунологическая группа признаков и возраст
3. данные до и после операции, гемодинамическая группа признаков и возраст
4. данные до операции, все признаки
5. старые данные

Подзадача 4 не делилась на иммунологическую и гемодинамическую группы в силу малого количества гемодинамических признаков, измеренных отдельно до операции.

К задачам классификации 1-5 применялся 10-fold скользящий контроль. Каждый объект попадал в обучение 9 раз, а в контроль один раз. Оценивалось количество неправильно классифицированных на контроле объектов класса 0 и класса 1. Обозначим через Test_0 — количество объектов класса 0, которые алгоритм отнес к классу 1, Test_1 — количество объектов класса 1, которые алгоритм отнес к классу 0. Ошибки классификации разделялись, т.к. классы являются неравнозначными: приписать здоровому больному отрицательный исход операции (допустить ошибку Test_0) не так страшно, как отнести человека с отрицательным отдаленным исходом к классу больных, у которых операция прошла успешно (допустить ошибку Test_1).

В таблице представлены результаты классификации.

Алгоритм	Тип ошибки	Задача 1	Задача 2	Задача 3	Задача 4	Задача 5
Conjuctive Rule	Test_0	3	5	5	6	5
	Test_1	18	17	18	16	16
Decision Table	Test_0	0	0	0	0	4
	Test_1	18	18	18	18	18
Random Tree	Test_0	8	9	11	10	19
	Test_1	14	14	15	17	12
Random Forest	Test_0	0	1	2	0	5
	Test_1	18	18	18	18	17
AdaBoost	Test_0	11	11	10	9	9
	Test_1	16	18	17	17	15
Multilayer Perceptron	Test_0	12	13	11	12	13
	Test_1	14	13	16	13	10
RBF NETWork	Test_0	0	0	0	0	0
	Test_1	18	18	18	18	18
Simple Logistic	Test_0	0	3	0	0	3
	Test_1	18	17	18	18	16
Bayes Net	Test_0	0	0	0	0	2
	Test_1	18	18	18	18	18
Naive Bayes	Test_0	13	11	14	11	11
	Test_1	13	13	15	14	13

Из-за небольшого количества объектов плохого класса в выборке (18%), алгоритмы показали низкое качество классификации объектов этого класса. Но объекты класса 1 имеют большую стоимость ошибки, нежели объекты класса 0, следует сбалансировать выборку путем введения весов классов. Класс 1 в четыре раза меньше, чем класс 0, поэтому обосновано использование веса, равного 4. Введем обозначения: w_0 — вес класса 0, w_1 — вес класса 1. Исследуем, как меняется качество классификации при использовании весов: $w_0 = 1$, $w_1 = 3, 4, 5, 7$. В следующей таблице представлены результаты классификации. Жирным выделены несколько лучших результатов классификации с точки зрения минимизации взвешенной по классам.

Алгоритм	w_1	Ошибка	Задача 1	Задача 2	Задача 3	Задача 4	Задача 5
Conjuctive Rule	3	Test_0	13	19	18	20	23
		Test_1	15	13	14	12	15
	4	Test_0	17	18	23	22	37
		Test_1	14	13	11	9	7
	5	Test_0	30	25	28	27	37
		Test_1	11	12	9	9	7
	7	Test_0	31	49	23	**33**	38
		Test_1	12	6	10	**8**	8
Decision Table	3	Test_0	11	19	10	24	22
		Test_1	18	13	14	14	16
	4	Test_0	15	24	12	17	24
		Test_1	12	15	10	15	12
	5	Test_0	22	34	**22**	**17**	23
		Test_1	11	14	**9**	**9**	12
	7	Test_0	**30**	33	30	25	19
		Test_1	**8**	12	8	8	10
Random Tree	3	Test_0	5	4	7	2	13
		Test_1	17	18	17	17	12
	4	Test_0	4	2	5	3	13
		Test_1	18	16	18	18	12
	5	Test_0	4	2	4	2	14
		Test_1	17	18	17	18	12
	7	Test_0	4	4	3	4	15
		Test_1	17	18	16	17	12
AdaBoost	3	Test_0	1	14	13	16	13
		Test_1	18	17	13	15	15
	4	Test_0	16	15	**18**	19	**11**
		Test_1	15	16	**11**	15	**12**
	5	Test_0	22	20	15	22	20
		Test_1	13	15	12	15	14
	7	Test_0	25	25	21	26	17
		Test_1	11	17	10	13	13
Multilayer Perceptron	3	Test_0	**13**	13	14	17	14
		Test_1	**12**	13	14	14	11
	4	Test_0	13	13	16	17	**13**
		Test_1	13	12	15	18	**11**
	5	Test_0	18	**13**	16	14	14
		Test_1	13	**11**	14	15	11
	7	Test_0	15	13	17	**15**	12
		Test_1	12	13	14	**12**	12

Алгоритм	w_1	Ошибка	Задача 1	Задача 2	Задача 3	Задача 4	Задача 5
RBF NETWork	3	Test_0	10	22	13	7	38
		Test_1	13	14	15	16	9
	4	Test_0	**16**	25	20	20	50
		Test_1	**10**	13	11	15	3
	5	Test_0	46	44	61	57	54
		Test_1	5	8	2	6	1
	7	Test_0	56	53	70	78	54
		Test_1	3	4	1	0	1
Simple Logistic	3	Test_0	12	17	6	8	15
		Test_1	14	13	14	18	15
	4	Test_0	14	18	14	11	22
		Test_1	15	13	14	17	11
	5	Test_0	46	34	24	18	23
		Test_1	5	11	11	18	12
	7	Test_0	44	41	40	74	27
		Test_1	10	9	9	6	13
Naive Bayes	3	Test_0	19	14	21	20	13
		Test_1	13	12	13	14	11
	4	Test_0	20	14	23	20	**13**
		Test_1	13	12	12	13	**11**
	5	Test_0	20	20	26	24	13
		Test_1	15	12	12	13	11
	7	Test_0	20	24	27	32	16
		Test_1	13	12	12	13	11

Применим лучшие алгоритмы классификации с подобранными весами класса 1 к задачам с произведенной фильтрацией объектов. Выберем объекты, у которых доля ошибок меньше 0.8 (см. рис. 12). Не будем классифицировать пять объектов класса 0 и один объект класса 1. К результатам классификации будут прибавляться ошибки на объектах-выбросах, т.е. к Test_0 будет прибавляться ошибка на 5 неклассифицированных объектах, а к Test_1 будет прибавляться ошибка на одном объекте-выбросе.

Номер задачи	Алгоритм	w_1	Ошибка	Результат
1	Decision Table	7	Test_0 Test_1	22 11
1	Multilayer Perceptron	3	Test_0 Test_1	17 8
1	RBF NETWork	4	Test_0 Test_1	28 10
2	Multilayer Perceptron	5	Test_0 Test_1	16 10
2	Decision Table	5	Test_0 Test_1	27 7
3	AdaBoost	4	Test_0 Test_1	28 12
4	Conjuctive Rule	7	Test_0 Test_1	49 9
4	Decision Table	5	Test_0 Test_1	29 8
4	Multilayer Perceptron	7	Test_0 Test_1	21 12
5	AdaBoost	4	Test_0 Test_1	12 10
5	Multilayer Perceptron	4	Test_0 Test_1	15 10
5	Naive Bayes	4	Test_0 Test_1	13 11

Фильтрация объектов вместе с настройкой весов класса 1 позволили значительно улучшить качество классификации.

Лучший классификатор для задачи 1 (данные до и после операции, все признаки), Multilayer Perceptron, на контроле допустил ошибки: 21% (17/82) для объектов класса 0 и 44% (8/18) для объектов класса 1, общая ошибка классификации составила 25%.

Для задачи 2 (данные до и после операции, иммунологическая группа признаков и возраст) классификатор Multilayer Perceptron на контроле допустил ошибки: 20% (16/82) для объектов класса 0 и 56% (10/18) для объектов класса 1, общая ошибка классификации составила 26%, классификатор Decision Table допустил меньше ошибок на объектах класса 1 (39%), но больше ошибался на объектах касса 0 (33%), общая ошибка составила 34%.

Алгоритм AdaBoost, примененный к задаче 3 (данные до и после операции, гемодинамическая группа признаков и возраст), на контроле допустил ошибки: 34% (28/82) для объектов класса 0 и 67% (12/18) для объектов класса 1, общая ошибка классификации составила 40%.

Для задачи 4 (данные до операции, все признаки) алгоритм Decision Table на контроле допустил ошибки: 35% (29/82) для объектов класса 0 и 44% (8/18) для объектов класса 1, общая ошибка классификации составила 37%.

Лучший классификатор для задачи 5 (старые данные), AdaBoost, на контроле допустил ошибки: 22% (12/54) для объектов класса 0 и 55% (10/18) для объектов класса 1, общая ошибка классификации составила 22%.

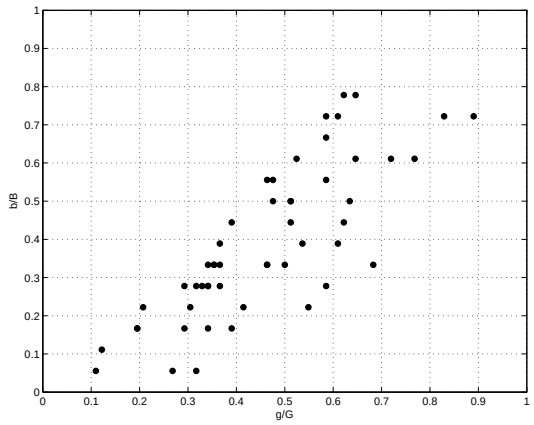

Рис. 13: Одномерные правила класса 0

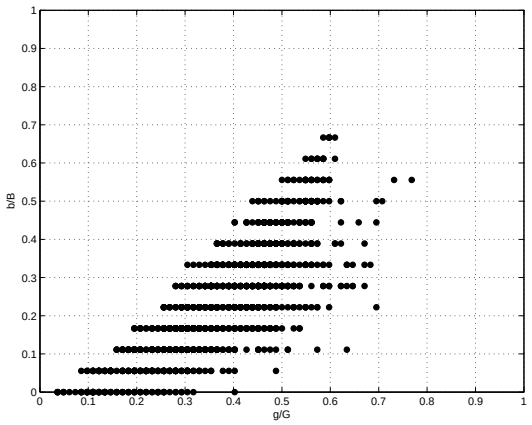

Рис. 14: Двумерные правила класса 0

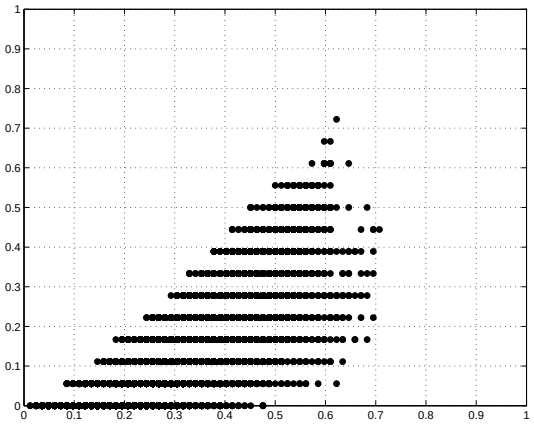

Рис. 15: Трехмерные правила класса 0

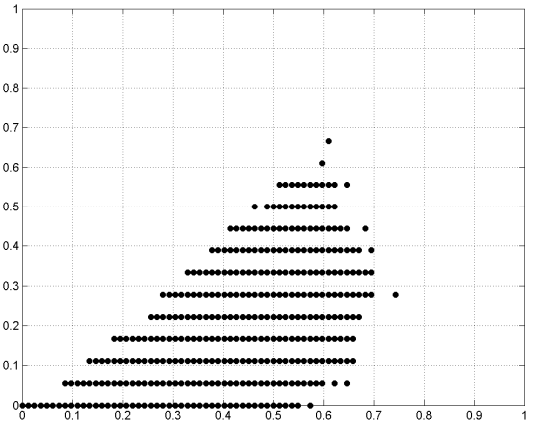

Рис. 16: Четырехмерные правила класса 0

3.7 Применение разработанной библиотеки логических алгоритмов

Для решения поставленной задачи прогнозирования отдаленных результатов хирургических операций использовалисть логические алгоритмы: голосование и решающий список.

Строились правила, представляющие собой конъюнкции всевозможных комбинаций из двух, трех, четырех термов разных признаков. Настройка производилась алгоритмом покоординатной оптимизации 2.3, пороги и знак определялись из соображений максимизации эвристики $h_{wra} = \frac{g}{G} - \frac{b}{B}$ или эвристики $h_{wsqrt} = \sqrt{\frac{g}{G}} - \sqrt{\frac{b}{B}}$. Были выбраны эвристики, взвещенные по классам, т.к. предыдущие исследования задачи показали существенную несбалансированность классов. На ROC графиках 13–16 изображены правила класса 0, отвечающие эвристике h_{wra}. Последовательно представлены правила длины 1–4. Чем больше длина правила, тем большей точности можно достичь. Правила размерности четыре на ROC графиках лежат ближе к точке (1,0).

Для правил строились ROC графики, находились оптимальные по Парето правила и правила, лежащие в окрестности оптимальных по Парето. На графиках 17–18 — правила, максимизирующие эвристику h_{wra}, на 19–20 — правила, максимизирую-

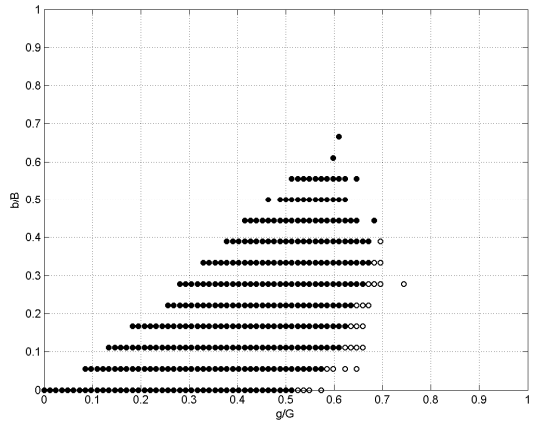

Рис. 17: Оптимальные правила класса 0, максимизирующие h_{wra}

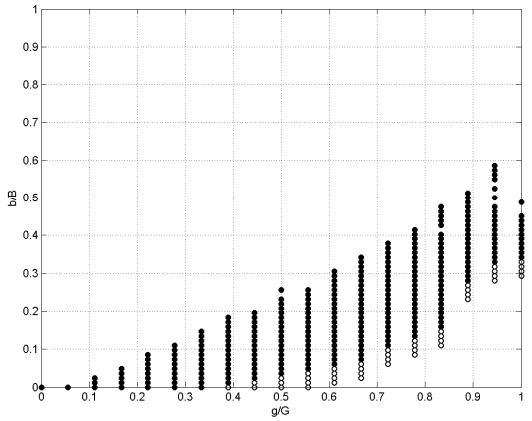

Рис. 18: Оптимальные правила класса 1, максимизирующие h_{wra}

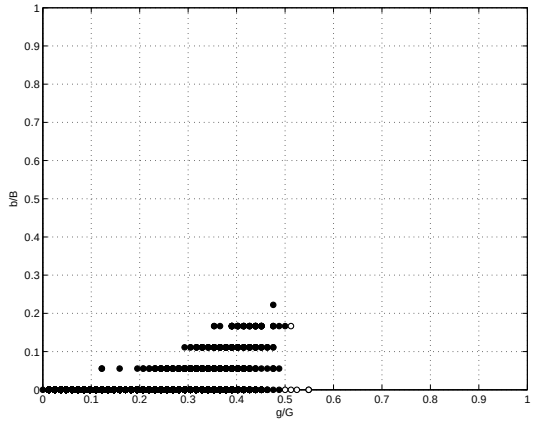

Рис. 19: Оптимальные правила класса 0, максимизирующие h_{sqrt}

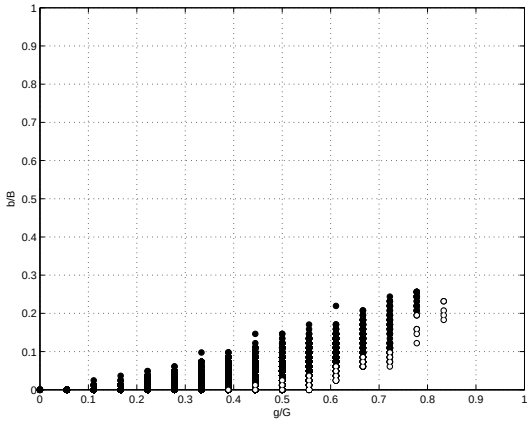

Рис. 20: Оптимальные правила класса 0, максимизирующие h_{sqrt}

щие эвристику h_{wsqrt}, закрашенные точки — все правила, незакрашенные точки — окрестности оптимальных по Парето правил глубины 4.

К задачам 1–5, с проведенной фильтрацией выбросов, применялись логические алгоритмы: голосование (алгоритм 2.5.2) и решающий список (алгоритм 2.5.1).

Алгоритмы строились из правил длины 3 и 4. Выбирались оптимальные по Парето правила, и правила, лежащие в окрестности оптимальных по Парето глубины 4. Оценивалось количество неправильно классифицированных на контроле и на обучении объектов класса 0 и класса 1. Как и в предыдущем разделе, обозначим через Test_0 — количество объектов класса 0, которые алгоритм отнес к классу 1 на контроле, Test_1 — количество объектов класса 1, которые алгоритм отнес к классу 0 на контроле, Train_0 — количество объектов класса 0, которые алгоритм отнес к классу 1 на обучении, Train_1 — количество объектов класса 1, которые алгоритм отнес к классу 0 на обучении. Сравнивая ошибки на обучении и контроле, экспериментально оценим насколько алгоритм переобучен.

В таблице результатов введены следующие обозначения: Vote — алгоритм голосования, DL — решающий список, 3D, 4D — длины правил (правила длины 3 и 4 соответственно), из которых строились алгоритмы, wra — в функции построения

правил GetRules использовалась эвристика h_{wra}, wsqrt — в функции построения правил GetRules использовалась эвристика h_{wsqrt}, глубина — глубина окрестности оптимальных по Парето правил.

Результаты классификации представлены в таблице.

Алгоритм	Глубина	Ошибка	Задача 1	Задача 2	Задача 3	Задача 4	Задача 5
Vote3D wra	4	Train_0	15	17	20	16	15
		Test_0	16	20	25	19	15
		Train_1	5	5	8	6	4
		Test_1	9	8	9	10	7
Vote3D wsqrt	4	Train_0	19	15	18	17	13
		Test_0	19	20	25	20	15
		Train_1	4	5	5	6	3
		Test_1	8	5	5	7	5
Vote4D wra	4	Train_0	13	13	14	15	10
		Test_0	13	15	17	17	11
		Train_1	5	7	6	7	4
		Test_1	6	6	6	8	7
Vote4D wsqrt	4	Train_0	9	12	19	15	12
		Test_0	15	15	21	19	14
		Train_1	4	7	10	6	3
		Test_1	5	6	11	7	5
DL3D wra	1	Train_0	20	21	25	20	16
		Test_0	23	23	27	21	18
		Train_1	7	8	9	8	6
		Test_1	8	10	11	9	9
DL3D wsqrt	1	Train_0	19	21	20	25	18
		Test_0	20	25	26	27	20
		Train_1	6	9	10	7	6
		Test_1	8	8	15	8	9
DL4D wra	1	Train_0	15	16	19	15	13
		Test_0	20	21	29	20	15
		Train_1	4	8	9	10	5
		Test_1	6	10	11	9	8
DL4D wsqrt	1	Train_0	13	14	17	13	10
		Test_0	17	18	20	17	14
		Train_1	5	6	10	8	9
		Test_1	5	8	12	9	9

Реализованные логические алгоритмы показывают хорошее качество классификации. Количество ошибочных классификаций объектов меньше, чем у алгоритмов библиотеки WRA. Качество классификации задачи 3 (данные до и после операции, гемодинамическая группа признаков и возраст) хуже, чем задачи 2 (данные до и после операции, иммунологическая группа признаков и возраст), что согласуется с измеренными информативностями признаков: иммунологическая группа признаков более информативна, чем гемодинамическая. Благодаря выбору эвристик, сбалансированных по классам, ошибка на объектах класса 1 (Test_1) не так велика. Сравнение ошибок на обучении и контроле показало несильную переобученность алгоритмов, но переобученность усиливается при построении правил большей размерности.

4 Заключение

В данной работе рассматривалась задача прогнозирования отдаленного результата хирургических операций. В ходе анализа данных был разработан удобный инструмент измерения информативностей признаков и объектов. Произведен анализ признакового описания задачи, выявлено, что признаки, измеренные после операции, более информативны, чем признаки, измеренные до операции, иммунологическая группа признаков информативнее гемодинамической. Анализ объектов задачи выявил 6 нетипичных объектов, их исключение из обучающей выборки позволило улучшить качество классификации.

К данным задачи были применены стандартные алгоритмы классификации, оценено качество.

Были разработаны специальные алгоритмы классификации, позволившие учесть все особенности задачи: наличие пропусков в данных (20% значений признаков не заполнены), неточность измерения значений признаков, небольшая длина выборки (100 объектов), маленькое количеством объектов с отрицательным отдаленным результатом операции (18%). Алгоритмы были реализованы в среде MATLAB. Произведены вычислительные эксперимены, оценено качество классификации на обучающей и контрольной выборках. Реализованные алгоритмы допускают ошибки порядка 20% на объектах класса 0 и 28% на объектах класса 1, они хорошо применимы к размытым данным и данным с большим количеством пропусков.

Список литературы

[1] Воронцов К.В. Комбинаторные оценки качества обучения по прецедентам // Докл. РАН. — 2004. — Т. 394, No 2. — С. 175–178.

[2] Воронцов К.В. Комбинаторный подход к оценке качества обучемых алгоритмов // Математические вопросы кибернетики / Под редакцией О.Б. Лупанов. — М.: Физматлит, 2004. — Т. 13 — С. 5–36.

[3] Воронцов К.В. Обзор соременных исследований по проблеме качества обучения алгоритмов // Таврический вестник информатики и митематики. — 2004.

[4] Воронцов К.В. Лекции по логическим алгоритмам классификации.

[5] Воронцов К.В. Лекции по методам оценивания и выбора моделей.

[6] Воронцов К.В. Лекции по алгоритмическим композициям.

[7] Furnkranz J. Modeling rule precision // LWA / Ed. by A. Abecker, S. Bickel, U. Brefeld, I. Drost, N. Henze, O. Herden, M. Minor et al. — Humbold-Universitat Berlin, 2004. — Pp. 147–154.

[8] Furnkranz J., Peter F. ROC 'n' rule learning — towards a better understanding of covering algorithms // Machine Learning. — 2005. — Vol. 58, no. 1. — Pp. 39–77.

[9] Janssen F., Furnkranz J. On meta-learning rule learning heuristics// LWA / Ed. by A. Hinneburg. — Martin-Luther-University Halle-Wittenberg, 2007. — Pp. 167–174.

[10] Ивахненко А.А. Методы улучшения обобщающей способности логических алгоритмов классификации.

[11] Загоруйко Н.Г., Елкина В.Н., Лбов Г.С Алгоритмы обнаружения эмпирических закономерностей. — Новосибирск, Наука, 1985.

[12] Васин А.А, Краснощеков П.С., Морозов В.В. Исследование операций. — М., Академия, 2008

[13] Кузнецов М.Р., Туркин П.Ю., Воронцов К.В., Дьяконов А.Г., Ивахненко А.А., Сиваченко Е.А. Прогнозирование результатов хирургического лечения атеросклероза на основе анализа клинических и иммунологических данных. — М., МАКС Пресс, 2007.