МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Сибирский государственный аэрокосмический университет
имени академика М.Ф. Решетнева»

На правах рукописи

Кузьмич Роман Иванович

МОДИФИЦИРОВАННЫЙ МЕТОД ЛОГИЧЕСКОГО АНАЛИЗА ДАННЫХ ДЛЯ ЗАДАЧ КЛАССИФИКАЦИИ

Специальность 05.13.01 – Системный анализ, управление
и обработка информации
(информатика, вычислительная техника и управление)

Диссертация на соискание ученой степени
кандидата технических наук

Научный руководитель:
кандидат физико-математических наук,
доцент И.С. Масич

Красноярск 2016

СОДЕРЖАНИЕ

ВВЕДЕНИЕ 4

1 АНАЛИЗ ЛОГИЧЕСКИХ АЛГОРИТМОВ КЛАССИФИКАЦИИ 10

1.1 Основные понятия логических алгоритмов классификации 10

1.2 Алгоритмы поиска закономерностей в форме конъюнкций 13

1.3 Анализ основных логических алгоритмов классификации и способов их построения 18

1.3.1 Решающие списки 18

1.3.2 Решающие деревья 21

1.3.3 Алгоритмы простого и взвешенного голосования правил 28

1.4 Анализ программных систем для решения задач классификации 35

Выводы 42

2 МЕТОД ЛОГИЧЕСКОГО АНАЛИЗА ДАННЫХ И ЕГО МОДИФИКАЦИИ 44

2.1 Описание подхода 44

2.2 Бинаризация признаков 45

2.3 Построение опорного множества 48

2.4 Формирование закономерностей 51

2.5 Построение классификатора 55

2.6 Модификации для метода логического анализа данных 57

2.7 Решение задач псевдобулевой оптимизации 64

Выводы 68

3 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ И ЭКСПЕРИМЕНТАЛЬНЫЕ ИССЛЕДОВАНИЯ НА ПРАКТИЧЕСКИХ ЗАДАЧАХ 71

3.1 Программная реализация метода логического анализа данных и особенности использования программной системы 71

3.2 Результаты экспериментальных исследований метода логического анализа данных и разработанных для него модификаций на практических задачах классификации 77

3.3 Настройка параметров метода логического анализа данных с учетом специфики решаемых задач 93

3.4 Сравнительный анализ метода логического анализа данных с другими алгоритмами классификации на практических задачах 96

Выводы 105

ЗАКЛЮЧЕНИЕ 107

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 109

ПРИЛОЖЕНИЕ А (Справочное) Названия полей базы данных и расшифровка их значений 122

ПРИЛОЖЕНИЕ Б (Справочное) Признаки с нулевой и максимальной важностью для задачи прогнозирования осложнений инфаркта миокарда 130

ВВЕДЕНИЕ

В настоящее время при решении задач распознавания образов, помимо требования высокой точности, часто возникает необходимость в интерпретируемости и обоснованности получаемых решений. Особенно интерпретируемость и обоснованность являются ключевыми факторами при решении тех практических задач, в которых потери от принятия неверного решения могут быть велики. Поэтому система поддержки принятия решений, используемая для таких задач, должна обосновывать возможные решения и интерпретировать результат.

Для создания такой системы потребуются алгоритмы классификации данных, которые помимо самого решения предоставляют в явном виде решающее правило, то есть выявляют знания из имеющихся данных. Это справедливо для логических алгоритмов классификации, принцип работы которых состоит в выявлении закономерностей в данных и формализации их в виде набора правил, т.е. набора закономерностей, описываемых простой логической формулой.

Процесс формирования логических правил сопровождается решением задач выбора наилучших альтернатив в соответствии с некоторым критерием. В предлагаемом методе логического анализа данных формализация процесса формирования логических правил осуществляется в виде ряда задач комбинаторной оптимизации, что формирует гибкий и эффективный алгоритм логического анализа для классификации данных. Объединив некоторое количество закономерностей в композицию, получаем классификатор, который решает поставленную задачу.

Однако в настоящее время существует ряд проблем, связанных с применением метода логического анализа данных при решении практических задач классификации. Одной из них является построение оптимизационных моделей для формирования информативных закономерностей. При

рассмотрении данного вопроса, прежде всего, необходимо определиться с теми критериями и ограничениями, которые лежат в основе этих оптимизационных моделей. Другой проблемой исследуемого метода является построение классификатора, который смог бы верно отнести новое наблюдение, т.е. наблюдение, не принимавшее участие при его построении, к тому или иному классу. Основной задачей на данном этапе метода является повышение интерпретируемости классификатора и качества классификации новых наблюдений, т. е. улучшение обобщающих способностей классификатора.

Таким образом, разработка модификаций для метода логического анализа данных, позволяющих улучшить интерпретируемость и обобщающие способности классификатора, является актуальной научно-технической задачей.

Следует отметить, что большой вклад в развитие логических алгоритмов классификации внесли следующие ученые: Ю. И. Журавлев, К. В. Рудаков, К. В. Воронцов, Н. Г. Загоруйко, Г. С. Лбов, Е. В. Дюкова, О. В. Сенько, В. И. Донской, P. L. Hammer, G. Alexe, S. Alexe, Y. Freund, R. E. Schapire.

Цель диссертационной работы состоит в повышении точности решения задач классификации и улучшении интерпретируемости классификатора, основанного на логических закономерностях.

Поставленная цель определила необходимость решения следующих задач:

1. Провести анализ существующих логических алгоритмов классификации, алгоритмов поиска информативных закономерностей для них, и основных программных систем, решающих практические задачи классификации.
2. Разработать алгоритмическую процедуру выбора базовых наблюдений для формирования закономерностей в методе логического анализа данных.

3. Разработать алгоритмическую процедуру улучшения закономерностей для повышения их информативности и усиления обобщающих способностей классификатора, построенного на базе данных закономерностей.

4. Создать модель оптимизации для формирования закономерностей, покрывающих существенно различные подмножества наблюдений обучающей выборки в методе логического анализа данных.

5. Разработать алгоритмическую процедуру построения классификатора, учитывающую информативность закономерностей, для метода логического анализа данных.

6. Модифицировать метод логического анализа данных на основе разработанных алгоритмических процедур.

7. Алгоритмизировать и реализовать метод логического анализа данных в виде программной системы, провести его апробацию и сравнительный анализ по точности с другими алгоритмами классификации на практических задачах.

Методы исследования. В диссертационной работе использовались методы системного анализа, теория множеств, теория вероятностей, комбинаторика, методы оптимизации.

Новые научные результаты, выносимые на защиту:

1. Разработана алгоритмическая процедура выбора базовых наблюдений для формирования закономерностей, отличающаяся от известных целенаправленным выбором базовых наблюдений, получаемых путем применения алгоритма «k-средних» к множеству наблюдений обучающей выборки, позволяющая сократить количество правил в классификаторе и снизить трудоемкость его построения при сохранении высокой точности.

2. Разработана алгоритмическая процедура наращивания закономерностей, полученных на базе оптимизационной модели с максимальным покрытием наблюдений обучающейся выборки, позволяющая повысить информативность правил, тем самым, способствуя увеличению точности принимаемых классификатором решений.

3. Создана модель оптимизации для формирования закономерностей, отличающаяся от известных наличием в целевой функции весового коэффициента покрываемого наблюдения, а также возможностью захвата наблюдений другого класса, позволяющая формировать правила, которые выделяют существенно различные подмножества наблюдений обучающей выборки.

4. Разработана алгоритмическая процедура построения классификатора как композиции информативных закономерностей, отличающаяся от известных совместным использованием критерия бустинга для оценки информативности закономерностей и новой итеративной процедуры выбора порога информативности, позволяющая сократить количество правил в классификаторе при сохранении высокой точности.

5. Модифицирован метод логического анализа данных на основе разработанных алгоритмических процедур, позволяющих повысить интерпретируемость классификатора, сокращая количество правил в нем, и сохранить при этом высокую точность при решении практических задач классификации.

Теоретическая значимость результатов диссертационного исследования состоит в разработке и исследовании модификаций для метода логического анализа данных, основанных на создании оптимизационных моделей для формирования информативных закономерностей и алгоритмических процедур сокращения количества правил в классификаторе, что является существенным вкладом в теорию интеллектуальных технологий и представления знаний, практики их применения в системах обработки информации и интеллектуального анализа данных.

Практическая значимость. На основе метода логического анализа данных реализована программная система поддержки принятия решений, которая позволяет, используя рекомендации по настройке ее параметров,

широкому кругу специалистов эффективно решать практические задачи классификации.

Материалы диссертационного исследования и разработанная программная система использованы для решения следующих практических задач: классификация результатов радарного сканирования, выявление спама, прогнозирование осложнений инфаркта миокарда.

Достоверность и обоснованность результатов диссертации подтверждается: исследованием существующих логических алгоритмов классификации и алгоритмов поиска информативных закономерностей для них, корректным обоснованием постановок задач, результатами применения предложенных моделей, методов и алгоритмических процедур, сравнительным анализом по точности с существующими алгоритмами классификации на практических задачах.

Реализация результатов работы. Диссертационная работа поддержана Фондом содействия развития малых форм предприятий в научно-технической сфере по программе «У.М.Н.И.К.» («Участник молодежного научно-инновационного конкурса») в рамках НИОКР «Разработка программной системы на базе логических алгоритмов классификации для решения задач медицинской диагностики и прогнозирования» на 2011-2013 гг. Результаты диссертации использовались в гранте Президента РФ МК-463.2010.9 «Комбинаторная оптимизация в задачах распознавания при диагностике и прогнозировании». Разработанная программная система «Логические анализ данных в задачах классификации» зарегистрирована в Реестре программ для ЭВМ 17 марта 2011 г. (свидетельство № 2011612265).

Апробация работы. Основные положения и результаты диссертации докладывались и обсуждались на XIV, XV Международной научной конференции «Решетневские чтения» (г. Красноярск 2010, 2011, 2014); XLIX Международной научной студенческой конференции «Студент и научно-технический прогресс» (г. Новосибирск 2011); III Общероссийской молодежной

научно-технической конференции «Молодежь. Техника. Космос» (г. Санкт-Петербург 2011); XIV Международной научно-технической конференции «Фундаментальные и прикладные проблемы приборостроения и информатики» (г. Москва 2011); Всероссийской молодежной научной конференции с международным участием «Современные проблемы фундаментальных и прикладных наук» (г. Кемерово 2011); Всероссийской научно-технической конференции студентов, аспирантов и молодых ученых «Научная сессия ТУСУР–2013» (г. Томск 2013).

Публикации. По теме диссертационной работы опубликовано 15 работ, из них 5 в изданиях из перечня ВАК, зарегистрирована программная система в Реестре программ для ЭВМ.

Структура работы. Диссертационная работа состоит из введения, трех глав, заключения, списка литературы из 115 источников и 2 приложений. Основной текст диссертации содержит 121 страницу, 10 рисунков, 19 таблиц.

1 АНАЛИЗ ЛОГИЧЕСКИХ АЛГОРИТМОВ КЛАССИФИКАЦИИ

1.1 Основные понятия логических алгоритмов классификации

Пусть φ: X → {0, 1} – некоторый предикат, определённый на множестве наблюдений X. Предикат φ покрывает наблюдение x, если $\varphi(x) = 1$. Предикат называют закономерностью, если он покрывает достаточно много наблюдений одного класса, и практически не покрывает наблюдения других классов [16].

Любая закономерность классифицирует только часть наблюдений из множества X. Объединив определённое количество закономерностей в композицию, можно получить классификатор, способный классифицировать любые наблюдения из множества. Логическими алгоритмами классификации будем называть композиции, состоящие из легко интерпретируемых закономерностей [11].

Класс, на базе наблюдений которого построена закономерность, будем называть своим классом. Чем больше наблюдений своего класса по сравнению с наблюдениями всех других классов покрывает закономерность, тем она более информативна. Наблюдения своего класса называют также положительными, а других – отрицательными. Покрытие отрицательного наблюдения является ошибкой закономерности, непокрытие положительного наблюдения считается ошибкой менее критичной, поскольку от закономерностей не требуется покрывать все наблюдения. Наблюдение, не покрытое одной закономерностью, может быть покрыто другой.

Для определения ε, δ–закономерности вводятся следующие обозначения:

P_k – число наблюдений своего класса «k» в выборке X^ℓ, где $P_k > 1$, $P_k + N_k = \ell$;

$p_k(\varphi)$ – из них число наблюдений, для которых выполняется условие $\varphi(x) = 1$;

N_k – число наблюдений всех остальных классов в выборке X^ℓ, где $N_k > 1$;

$n_k(\varphi)$ – из них число наблюдений, для которых выполняется условие $\varphi(x) = 1$.

Задача построения информативной закономерности состоит в оптимизации по двум критериям: $p_k(\varphi) \rightarrow \max$ и $n_k(\varphi) \rightarrow \min$. Наименее пригодны с точки зрения классификации те закономерности, которые либо покрывают слишком мало наблюдений, либо покрывают положительные и отрицательные наблюдения примерно в той же пропорции, в которой они были представлены во всей выборке.

Далее вводятся обозначения E_k для доли отрицательных среди всех покрываемых наблюдений, и D_k для доли покрываемых положительных наблюдений:

$$E_k(\varphi, X^\ell) = \frac{n_k(\varphi)}{p_k(\varphi) + n_k(\varphi)}, \qquad D_k(\varphi, X^\ell) = \frac{p_k(\varphi)}{p_k(\varphi) + n_k(\varphi)}.$$

Закономерность φ называется логической ε, δ–закономерностью для класса «k», если $E_k(\varphi, X^\ell) \le \varepsilon$ и $D_k(\varphi, X^\ell) \ge \delta$ при заданных достаточно малом ε и достаточно большом δ из отрезка [0, 1] [11].

Закономерность φ называется чистой или непротиворечивой, если $n_k(\varphi)=0$. Если $n_k(\varphi) > 0$, то закономерность φ называется частичной.

Если длина выборки мала или данные практически не содержат шума, тогда лучше искать чистые закономерности для подобного рода задач. Например, классификация месторождений редких полезных ископаемых по данным геологоразведки, где данные стоят дорого, и потому, как правило, тщательно проверяются. Но чаще данные оказываются неполными и неточными, например в медицинских и экономических задачах. Для них вполне допустима незначительная доля ошибок на обучающей выборке. При решении таких задач лучше использовать частичные закономерности. В этом случае, сравнивать и отбирать закономерности приходится по двум критериям одновременно. Для целенаправленного поиска лучших закономерностей удобнее иметь скалярный критерий информативности.

Одним из таких критериев является статистический критерий информативности [11]. Пусть X – вероятностное пространство, выборка X^ℓ – случайная, независимая, одинаково распределённая, y*(x) и φ(x) – случайные величины. Допускается справедливость гипотезы о независимости событий {x: y*(x) = k} и {x: φ(x) = 1}. Тогда вероятность реализации пары (p, n) подчиняется гипергеометрическому распределению [85]:

$$h_{P,N}(p,n) = \frac{C_P^p C_N^n}{C_{P+N}^{p+n}}\ , \qquad (1.1)$$

где $C_m^k = \frac{m!}{k!(m-k)!}$ – биномиальные коэффициенты, $0 \le k \le m$, $0 \le p \le P$, $0 \le n \le N$.

Если вероятность (1.1) мала, а пара (p, n) реализовалась, то гипотеза о независимости должна быть отвергнута. Чем меньше значение вероятности, тем более значимой является связь между y* и φ.

Информативность закономерности φ относительно класса «k» по выборке X^ℓ есть:

$$I_k(\varphi, X^\ell) = -\ln h_{Pk,Nk}(p_k(\varphi), n_k(\varphi)). \qquad (1.2)$$

Закономерность φ называется статистической закономерностью для класса «k», если $I_k(\varphi, X^\ell) > I_0$ при заданном достаточно большом I_0. Для каждой задачи порог информативности I_0 выбирается индивидуально.

Альтернативой статистическому является энтропийный критерий информативности, который следует из теории информации [64, 41]. Если имеются два исхода ω_0, ω_1 с вероятностями q_0 и $q_1 = 1 - q_0$, то количество информации, связанное с исходом ω_i, по определению равно $-\log_2(q_i)$.

Энтропия определяется как математическое ожидание количества информации [52]:

$$H(q_0, q_1) = -q_0 \log_2(q_0) - q_1 \log_2(q_1).$$

Следует считать появление наблюдения класса «k» исходом ω_0, а появление наблюдения любого другого класса исходом ω_1. Тогда, подставляя вместо вероятностей частоты, можно оценить энтропию выборки X^ℓ:

$$\hat{H}(P,N) = H\left(\frac{P}{P+N}, \frac{N}{P+N}\right).$$

Стало известно, что закономерность φ выделила p наблюдений из P, принадлежащих классу «k», и n наблюдений из N, не принадлежащих ему. Тогда энтропия выборки – $\{x \in X^{\ell} \mid \varphi(x) = 1\}$ есть $\hat{H}(p,n)$. Вероятность появления наблюдения из этой выборки оценивается как $\frac{p+n}{P+N}$. Аналогично, энтропия выборки – $\{x \in X^{\ell} \mid \varphi(x) = 0\}$ есть $\hat{H}(P-p, N-n)$, а вероятность появления наблюдения из неё оценивается как $\frac{P-p+N-n}{P+N}$. Таким образом, энтропия всей выборки после получения информации φ становится равна:

$$\hat{H}(P,N,p,n) = \frac{p+n}{P+N}\hat{H}(p,n) + \frac{P-p+N-n}{P+N}\hat{H}(P-p, N-n).$$

В итоге уменьшение энтропии составляет [19]:

$$IGain_k(\varphi, X^{\ell}) = \hat{H}(P,N) - \hat{H}_{\varphi}(P,N,p,n).$$

Это есть мера информационного выигрыша – количество информации об исходном делении выборки на два класса «k» и «не k», которое содержится в закономерности φ.

Закономерность φ является закономерностью по энтропийному критерию информативности, если $IGain_k(\varphi, X^{\ell}) > G_0$ при заданном достаточно большом G_0.

Особенностью энтропийного критерия является завышение информативности малых закономерностей, статистический критерий в данном случае работает лучше. На практике, как правило, используют энтропийный критерий информативности, так как он проще с точки зрения его вычисления.

1.2 Алгоритмы поиска закономерностей в форме конъюнкций

Информативные закономерности служат исходными данными для построения логических алгоритмов классификации. Множество предикатов, в

котором следует искать информативные закономерности, называют пространством поиска. Если все исходные признаки являются бинарными, тогда пространство поиска образуется самими признаками и всевозможными булевыми функциями, которые из этих признаков можно построить. Сложнее дело обстоит в тех случаях, когда наблюдения описываются разнотипными признаками: номинальными, порядковыми, количественными. Подобного рода ситуации чаще возникают на практике. Тогда пространством поиска становятся всевозможные бинарные функции от исходных признаков. Процесс построения таких функций называют бинаризацией исходной информации. Способы бинаризации подробно рассмотрены в пункте 2.2 данной работы.

После бинаризации признаков в качестве закономерностей можно брать только конъюнкции этих признаков или их отрицаний. Пусть B – конечное множество предикатов (термов), которые называются элементарными. Рассматривается множество конъюнкций с ограниченным числом термов из B:

$$K_K[B] = \{\varphi(x) = B_1(x) \wedge \ldots \wedge B_k(x) | B_1(x), \ldots, B_k(x) \in B, k \le K\}.$$

Количество термов k в конъюнкции называется её рангом (степенью). Конъюнкции небольшой степени обладают полезным преимуществом – они имеют вид привычных для человека логических высказываний и легко поддаются интерпретации.

Процедура поиска наиболее информативных конъюнкций в общем случае требует полного перебора. Число допустимых конъюнкций может оказаться настолько большим, что полный перебор станет практически неосуществим. На практике используют различные эвристики для сокращённого целенаправленного поиска конъюнкций, близких к оптимальным. Идея всех этих методов заключается в том, чтобы не перебирать огромное количество заведомо неинформативных термов.

Далее приведем обзор наиболее известных алгоритмов синтеза конъюнкций [11].

Наиболее простой из них – «Градиентный» алгоритм синтеза конъюнкций. Суть алгоритма заключается в том, что каждой конъюнкции φ ставиться в соответствие её окрестность – множество конъюнкций $V(\varphi)$, получаемых из φ путём элементарных модификаций: добавлением, удалением или модификацией одного из термов конъюнкции.

Начиная с заданной конъюнкции φ_0 (например, пустой), строится последовательность конъюнкций $\varphi_0, \varphi_1, \ldots, \varphi_t, \ldots$, в которой каждая следующая конъюнкция φ_t выбирается из окрестности предыдущей $V_t = V(\varphi_{t-1})$ по критерию максимума информативности (1.2).

Конъюнкции с высокой информативностью могут допускать много ошибок, хотя они считаются наиболее перспективными с точки зрения дальнейших модификаций. Поэтому на каждом шаге t выделяется «наилучшая» конъюнкция, удовлетворяющая дополнительному условию $E_k(\varphi_t^*) < \varepsilon$, и в общем случае не совпадающая с наиболее перспективной φ_t.

Итерационный процесс сходится за конечное число шагов к некоторой «локально неулучшаемой» конъюнкции, так как множество конъюнкций, различно классифицирующих выборку, конечно.

Ясно, что ни о каком градиенте в прямом смысле слова речь не идёт. Данный алгоритм, по аналогии с градиентным спуском, выбирает на каждой итерации наилучшую из ближайших точек пространства поиска.

Критерий информативности и функция окрестности являются параметрами алгоритма. При формировании окрестности можно применять различные эвристики [11]:

– ограничивать максимальный ранг конъюнкций;

– разрешать только добавления термов;

– чередовать серии добавлений термов с сериями удалений;

– разрешать модификацию нескольких термов одновременно.

Изменяя параметры «градиентного» алгоритма синтеза конъюнкций, можно получать различные процедуры поиска или улучшения информативных конъюнкций. Ниже приведены несколько вариантов этого алгоритма.

Жадный алгоритм синтеза конъюнкции использует только операцию добавления термов. Начальным приближением является конъюнкция, не содержащая термов. Недостаток жадной стратегии в том, что она может уводить в сторону от глобального максимума информативности, поскольку терм, найденный на i-м шаге, перестаёт быть оптимальным после добавления последующих термов. Тем не менее, на некоторых практических задачах данная эвристика способна находить хорошие закономерности.

Случайный локальный поиск [96, 114] начинает с пустой конъюнкции, но применяет весь набор возможных модификаций. Это преимущество по сравнению с жадным алгоритмом, так как появляется возможность удалять и заменять неоптимальные термы. Недостатком является наличие, как правило, очень большой мощности окрестности $|V(\varphi)|$, что затруднит перебор всех допустимых модификаций. Для устранения недостатка в случайном локальном поиске строится не вся окрестность, а только некоторое её случайное подмножество. Максимальная допустимая мощность этого подмножества задаётся как дополнительный параметр алгоритма.

С целью улучшения конъюнкции, построенной жадным наращиванием или случайным локальным поиском, к ней применяют процедуры стабилизации и редукции.

Основной идеей процедуры стабилизации является попытка улучшения конъюнкции путем удаления или замены по одному терму. В отличие от случайного локального поиска, перебираются все возможные удаления и замены. Модификации производятся до тех пор, пока возрастает информативность конъюнкции. В результате стабилизации найденные конъюнкции часто сходятся к одним и тем же локальным максимумам информативности, что положительно сказывается на интерпретируемости

правил. Также процедура стабилизации рекомендуется для настройки порогов в термах.

Целью процедуры редукции является повышение обобщающей способности правила. Ее отличие от стабилизации заключается в том, что термы только удаляются, а информативность вычисляется по независимой контрольной выборке X^k, составленной из наблюдений, не участвовавших в построении конъюнкции. Контрольную выборку формируют до начала обучения, выделяя случайным образом из массива исходных данных примерно 25% наблюдений. При этом наблюдения разных классов распределяются в той же пропорции, что и во всей выборке. Смысл редукции в том, чтобы проверить, не является ли найденная конъюнкция избыточно сложной, и либо упростить её, либо вовсе признать неудачной и удалить из классификатора. Упрощение повышает общность закономерности, поскольку множество покрываемых ей наблюдений расширяется. Недостаток редукции заключается в том, что она оставляет значительную долю данных для контроля, уменьшив представительность обучающей выборки. Однако при приемлемом выборе соотношения ℓ/k поиск правил по X^ℓ с последующей редукцией по X^k может давать лучшие результаты, чем поиск по $X^\ell \cup X^k$ без редукции.

Дальнейшим усовершенствованием случайного локального поиска на основе идей дарвиновской эволюции является генетический алгоритм синтеза конъюнкций. Основное отличие генетического алгоритма от случайного локального поиска в том, что на каждом шаге отбирается не одна наилучшая конъюнкция, а целое множество лучших конъюнкций, называемое популяцией. Из них порождается большое количество конъюнкций–потомков с помощью двух генетических операций – скрещивания и мутации. Скрещивание – образование новой конъюнкции путём обмена термами между двумя членами популяции. В роли мутаций выступают операции добавления, замещения и удаления термов.

Следует отметить, что генетические алгоритмы отличаются большим разнообразием всевозможных эвристик, заимствованных непосредственно из живой природы. Например, в генетический алгоритм легко встроить процедуру селекции или искусственного отбора, порождая потомков только от наилучших конъюнкций, или задавая распределение вероятностей на популяции так, чтобы вероятность стать родителем увеличивалась с ростом информативности. Эти и другие эвристики описаны в литературе по генетическим алгоритмам [12, 21, 54, 85].

Для поиска конъюнктивных закономерностей можно также приспособить методы отбора признаков, описанные в [23]: метод последовательного сокращения признаков (алгоритм Del [104]), метод последовательного добавления признаков (алгоритм Add [8]), алгоритм Add–Del [24], метод случайного поиска с адаптацией (алгоритм СПА [42]). Для этого достаточно заменить в них функционал качества – вместо минимизации средней ошибки искать максимум информативности.

1.3 Анализ основных логических алгоритмов классификации и способов их построения

1.3.1 Решающие списки

Решающий список [103] – логический алгоритм классификации a: X → Y, который задаётся набором закономерностей $\varphi_1(x), \ldots, \varphi_T(x)$, приписанных к классам $k_1,\ldots, k_T$ соответственно, и вычисляется согласно алгоритму 1 [11]:

Алгоритм 1. Классификация наблюдения решающим списком

1. t = 1;
2. Если $\varphi_t(x) = 1$, то вернуть k_t;
   иначе t = t + 1;

3. если $t \le T$, то на 2;

иначе вернуть k_0.

Согласно алгоритму 1, при классификации наблюдения x закономерности проверяются последовательно для всех t = 1,…,T, пока для некоторого t не выполнится условие $\varphi_k^t(x) = 1$. Ответ k_0 означает отказ алгоритма от классификации наблюдения x. Часто такие наблюдения приписывают классу, имеющему минимальную цену ошибки.

Для формирования решающего списка используется алгоритм 2 [11]:

Алгоритм 2. Жадный алгоритм построения решающего списка

1: $U = X^\ell$;

2: для всех $t = 1,\ldots,T_{\max}$

3: $k = k_t$ – выбрать класс, для которого будет строиться правило;

4: найти наиболее информативное правило при ограничении на долю ошибок: $\varphi_t = \arg\max_{\varphi \in \Phi'} I_k(\varphi, U)$, где $\Phi' = \{\varphi \in \Phi : E_k(\varphi, U) \le E_{\max}\}$

5: если $I_k(\varphi_t, U) < I_{\min}$, то выход;

6: исключить из выборки наблюдения, выделенные правилом φ_t: $U = \{x \in U : \varphi_t(x) = 0\}$;

7: если $|U| \le \ell_0$, то выход,

где X^ℓ – обучающая выборка, Φ – семейство предикатов, из которого выбираются закономерности, $T_{\max}$ – максимальное допустимое число правил в списке, $I_{\min}$ – минимальная допустимая информативность правил в списке, $E_{\max}$ – максимальная допустимая доля ошибок на обучающей выборке, ℓ_0 – максимальное допустимое число отказов.

На каждой итерации алгоритм 2 строит ровно одно правило φ_t, покрывающее максимальное число наблюдений некоторого класса k_t и минимальное число наблюдений других классов. Для этого производится поиск

наиболее информативного правила на шаге 4 алгоритма, допускающего относительно мало ошибок на обучающей выборке. После построения правила φ_t покрытые им наблюдения удаляются из выборки и алгоритм переходит к поиску следующего правила φ_{t+1} по оставшимся наблюдениям. В итоге выборка оказывается покрытой множествами вида $\{x: \varphi_t(x) = 1\}$. По этой причине решающий список называют также покрывающим набором закономерностей или машиной покрывающих множеств [109].

В алгоритме 2 одновременно работают три критерия останова:

– построение заданного числа правил $T_{\max}$;

– покрытие всей выборки, за исключением не более ℓ_0 наблюдений;

– невозможность найти правило с информативностью выше $I_{\min}$ по остатку выборки.

Для нахождения баланса между точностью классификации обучающей выборки и длиной списка используется в алгоритме параметр $E_{\max}$. Уменьшение $E_{\max}$ приводит к снижению числа ошибок на обучении, но усложняет отбор правил, способствует уменьшению числа наблюдений, покрываемых отдельными правилами, и увеличению длины списка. Правила, покрывающие слишком мало наблюдений, статистически не надёжны и могут допускать много ошибок на независимых контрольных данных. Таким образом, увеличение длины списка при одновременном малом покрытии правил может приводить к эффекту переобучения. Учитывая это, значение $E_{\max}$ должно быть примерно равно доле ошибок, которую мы ожидаем получить как на обучающей выборке, так и контрольной. На практике параметр $E_{\max}$ подбирается экспериментально.

Важным моментом Алгоритма 2 является выбор класса на шаге 3. Для реализации этого предлагается два альтернативных варианта.

Первый вариант состоит в поочередном построении набора правил для каждого класса. Классы берутся в порядке убывания важности или цены ошибки. Преимущество данного варианта в том, что правила оказываются

независимыми – в пределах своего класса их можно переставлять местами. Это улучшает интерпретируемость правил.

Второй вариант заключается в совмещении шагов 3 и 4 алгоритма и выборе пары (φ_t, k_t), для которой информативность $I_{k_t}(\varphi_t, U)$ максимальна. Тогда правила различных классов могут следовать неупорядоченно. Доказано, что списки такого типа реализуют более широкое множество функций [103]. При этом улучшается разделяющая способность списка, но ухудшается его интерпретируемость. На практике первый вариант часто оказывается более удобным.

Особенностью решающих списков является решение проблемы пропущенных данных. Если для вычисления правила φ_t не хватает данных, то считается, что $\varphi_t(x) = 0$, и обработку наблюдения x берут на себя следующие правила в списке. Это относится и к стадии обучения, и к стадии классификации.

К преимуществам решающих списков можно отнести:

– Интерпретируемость правил;

– Возможность обработки данных с пропусками.

К недостаткам решающих списков можно отнести:

– Список может не построиться, если множество предикатов выбрано неудачно. При этом возможен высокий процент отказов от классификации.

– Список плохо интерпретируется, если он длинный и правила различных классов следуют неупорядоченно.

– Каждое наблюдение классифицируется только одним правилом, что не позволяет правилам компенсировать ошибки друг друга. Данный недостаток устраняется путём голосования правил.

1.3.2 Решающие деревья

Решающее дерево – логический алгоритм классификации, отличающийся от решающего списка тем, что при синтезе все конъюнкции строятся одновременно.

Деревом называется конечный связный ациклический граф с выделенной вершиной $v_0 \in V$, называемой корневой вершиной (корнем) дерева [18]. Вершина, не имеющая выходящих рёбер, называется терминальной или листом. Остальные вершины называются внутренними. Дерево называется бинарным, если из любой его внутренней вершины выходит ровно два ребра. Выходящие рёбра связывают каждую внутреннюю вершину с левой дочерней вершиной L_v и с правой дочерней вершиной R_v.

Бинарное решающее дерево – это алгоритм классификации, задающийся бинарным деревом, в котором каждой внутренней вершине $v \in V$ приписан предикат $\beta_v : X \to \{0, 1\}$, каждой терминальной вершине $v \in V$ приписано имя класса k_v. При классификации наблюдения $x \in X$ он проходит по дереву путь от корня до некоторого листа, в соответствии с алгоритмом 3 [11]:

Алгоритм 3. Классификация наблюдения решающим деревом

1: $v := v_0$;
2: пока вершина v внутренняя:
   если $\beta_v(x) = 1$, то $v := R_v$ (переход вправо);
   иначе $v := L_v$ (переход влево);
3: вернуть k_v.

Основными элементами алгоритмов синтеза решающих деревьев являются: выбор критерия ветвления для поиска атрибута во внутреннюю вершину решающего дерева, выбор критерия остановки для прекращения ветвления с целью получения терминальных вершин, и определение имени класса, приписываемого листу.

Признак обладает свойством полной отделимости, если логическое правило, построенное на базе этого признака, разделяет исходное множество наблюдений на два подмножества: подмножество A, которое содержит наблюдения только одного класса; подмножество B, которое содержит

наблюдения только одного класса и классы наблюдений из A и B различны. Признак обладает свойством частичной отделимости, если подмножество A или подмножество B содержит наблюдения только одного класса [18].

Большая часть эвристических алгоритмов синтеза решающих деревьев предпочитают признаки со свойством полной или частичной отделимости. Считается, что такие признаки формируют решающее правило, которое допускает минимальное число ошибок на наблюдениях экзаменующей выборки [77, 82].

В зависимости от выбора алгоритма разбиения по данным наблюдениям обучающей выборки может быть построено несколько решающих деревьев, позволяющих верно классифицировать эти наблюдения. В этом случае необходимо выбрать решающее дерево, которое лучше остальных позволит правильно классифицировать наблюдения, не участвовавшие в обучении. Как правило, таким деревом считают наиболее простое решающее дерево, точное на всех обучающих наблюдениях.

На практике применяют различные эвристики, нацеленные на построение как можно более простого дерева, обладающего как можно лучшим качеством классификации. Придумано огромное количество различных алгоритмов синтеза бинарных решающих деревьев по обучающей выборке.

Сначала следует рассмотреть простой жадный алгоритм ID3, основанный на принципе «разделяй и властвуй» [105].

Идея алгоритма заключается в последовательном разбиении выборки на две части до тех пор, пока в каждой части не окажутся наблюдения только одного класса. Проще всего записать этот алгоритм в виде рекурсивной процедуры LearnID3, которая строит дерево по заданной подвыборке U [11]:

Алгоритм 4. Алгоритм синтеза бинарного ID3

1: ПРОЦЕДУРА LearnID3 (U);
2: если все наблюдения из U лежат в одном классе k то:

создать новый лист v;

$k_v = k$;

вернуть (v);

3: найти предикат с максимальной информативностью: $\beta = \arg\max_{\beta \in B} I(\beta, U)$;

4: разбить выборку на две части $U = U_0 \cup U_1$ по предикату β:

$U_0 = \{x \in U : \beta(x) = 0\}$;

$U_1 = \{x \in U : \beta(x) = 1\}$;

5: если $U_0 = Ø$ или $U_1 = Ø$ то:

создать новый лист v;

k_v = класс, в котором находится большинство наблюдений из U;

6: иначе:

создать новую внутреннюю вершину v;

$\beta_v := \beta$;

L_v := LearnID3 (U_0); (построить левое поддерево);

R_v := LearnID3 (U_1); (построить правое поддерево);

7: вернуть (v).

На шаге 3 алгоритма 4 выбирается предикат β из заданного семейства B, задающий максимально информативное ветвление дерева – разбиение выборки на две части $U = U_0 \cup U_1$.

На практике применяются различные критерии ветвления. В данном случае критерий, ориентированный на отделение заданного класса «k»:

$$I(\beta, U) = \max_{k \in Y} I_k(\beta, U).$$

Задача выбора наиболее информативного предиката на шаге 3 алгоритма 4 легко решается полным перебором, если мощность множества предикатов не велика. Иначе необходимо применять эвристические процедуры направленного поиска.

Среди преимуществ алгоритма ID3 можно выделить [11]:

– Простота и интерпретируемость классификации. Алгоритм способен не только классифицировать наблюдение, но и выдать объяснение классификации в терминах предметной области. Объяснение строится путём выписывания последовательности условий, проверенных для данного наблюдения на пути от корня дерева до листа. Данные условия образуют конъюнкцию, то есть легко интерпретируемое логическое правило.

– Трудоёмкость алгоритма линейна по длине выборки.

– Не бывает отказов от классификации, в отличие от решающих списков.

– Алгоритм очень прост для реализации и легко поддаётся различным усовершенствованиям. Можно использовать различные критерии ветвления и критерии останова, вводить редукцию, и т. д.

Среди недостатков алгоритма ID3 можно выделить:

– Жадность. Локально оптимальный выбор предиката не является глобально оптимальным. В случае неудачного выбора алгоритм не способен вернуться на уровень вверх и заменить неудачный предикат.

– Чем дальше вершина расположена от корня дерева, тем меньше длина подвыборки, по которой принимается решение о ветвлении в данной вершине, тем менее статистически надёжным является выбор предиката.

– Высокая чувствительность к составу выборки. Изменение данных в 1–2 наблюдениях часто приводит к радикальному изменению структуры дерева.

– Алгоритм ID3 переусложняет структуру дерева, и, как следствие, склонен к переобучению. Его обобщающая способность относительно невысока.

ID3 не позволяет обрабатывать большие массивы информации, характеризующиеся противоречивыми данными, т.е. при наличии в выборке двух и более одинаковых наблюдений с указанной принадлежностью к разным классам; пропусками в данных; признаками, принимающими значения из непрерывного интервала. Решение этих проблем привело к созданию нового поколения алгоритмов обучения, основанных на построении решающих

деревьев. Наиболее известным из них является алгоритм C4.5 [95, 106]. Он использует в качестве критерия ветвления оценку прироста (информационного выигрыша) – теоретико-информационную меру, которая оценивает пригодность признака по относительной величине прироста информации. Во внутреннюю вершину решающего дерева выбирается признак с максимальной оценкой прироста. Наиболее подробно алгоритм C4.5 описан в главе 3.4 диссертационного исследования.

Для использования обучающих множеств большого объема невозможного в рамках ID3 Куинланом [105] предложен подход, называемый методом «окна» (windowing method). «Окном» называется произвольное подмножество наблюдений всего обучающего множества. Суть метода состоит в произвольном выборе «окна» из обучающего множества и построении по нему решающего дерева. В случае если «окно» правильно классифицирует оставшиеся наблюдения обучающего множества, работа метода завершается. Иначе, в «окно» добавляются неправильно классифицированные наблюдения обучающего множества, и по ним снова строится решающее дерево. Процесс продолжается до полного исчерпания неправильно классифицированных наблюдений.

В работе [42] предложены эвристические алгоритмы формирования логических решающих функций с выделением признаковых предикатов. В алгоритме CORAL при классификации наблюдения в каждой вершине дерева проверяется истинность высказывания, являющегося конъюнкцией простых высказываний. На этапе обучения и классификации возможны пропуски значений признаков. В алгоритме DW при классификации нового наблюдения в каждой вершине дерева проверяется истинность простого высказывания. Алгоритм DW приводит к локально-оптимальному решению и предназначен для задач двухклассовой классификации. Для корректной работы алгоритма DW пропуски значений признаков должны отсутствовать.

В работах [77, 101, 102] описан эвристический критерий разбиения, используемый в алгоритме CART. Критерий разбиения основан на индексе *Gini*, базирующемся на идее уменьшения неопределенности в узле. Наиболее подробно алгоритм CART описан в главе 3.4 диссертационного исследования.

Главная причина недостатков рассмотренных алгоритмов – неоптимальность жадной стратегии наращивания дерева. Для решения этой проблемы применяют различные эвристические приемы, например, редукцию или построение совокупности деревьев – решающего леса.

Ключевой идеей редукции является удаление поддеревьев, имеющих недостаточную статистическую надёжность. При этом дерево перестаёт корректно классифицировать обучающую выборку, зато качество классификации новых наблюдений, как правило, улучшается.

Придумано огромное количество эвристик для проведения редукции, хотя ни одна из них не гарантирует улучшения качества классификации. Наиболее известные стратегии редукции:

– редукция на основе сокращенной ошибки [77, 87];

– редукция на основе пессимистической ошибки [77, 88];

– редукция на основе критического значения [78];

– редукция на основе оценки цены-сложности [77];

– редукция, основанная на минимальной ошибке [97].

Подробнее с ними можно познакомиться в [20]. К наиболее простым вариантам редукции относятся предредукция и постредукция [11].

Суть предредукции (pre-pruning) заключается в том, что она досрочно прекращает дальнейшее ветвление в вершине дерева, если информативность $I(\beta,U)$ для всех предикатов не дотягивает до заданного порогового значения I_0. Предредукция считается не самым эффективным способом избежать переобучения, так как жадное ветвление по-прежнему остаётся глобально неоптимальным. Более эффективной считается стратегия постредукции.

При использовании постредукции (post-pruning) просматриваются все внутренние вершины дерева и заменяются отдельные вершины либо одной из дочерних вершин (при этом вторая дочерняя удаляется), либо терминальной вершиной. Критерием замены является сокращение числа ошибок на контрольной выборке, отобранной заранее, и не участвовавшей в обучении дерева. Процесс замен продолжается до тех пор, пока в дереве остаются вершины, удовлетворяющие критерию замены.

1.3.3 Алгоритмы простого и взвешенного голосования правил

Пусть для каждого класса $k \in Y$ построено множество логических закономерностей (правил), необходимых для классификации наблюдений данного класса: $R_k = \{\varphi_k^t : X \to \{0, 1\} \mid t = 1,\ldots,T_k\}$. Следует отметить, что если $\varphi_k^t(x) = 1$, то правило φ_k^t относит наблюдение $x \in X$ к классу «k». Если же $\varphi_k^t(x) = 0$, то правило φ_k^t воздерживается от классификации наблюдения x.

Алгоритм простого голосования подсчитывает долю правил в наборах R_k, относящих наблюдение x к каждому из классов [11]:

$$\Gamma_k(x) = \frac{1}{T_k}\sum_{t=1}^{T_k}\varphi_k^t(x),$$

и относит наблюдение x к тому классу, за который подана наибольшая доля голосов: $a(x) = \arg\max_{k \in Y} \Gamma_k(x)$.

Если доля голосов одинакова для нескольких классов, выбирается тот, для которого цена ошибки меньше. Нормирующий множитель $1/T_k$ вводится для того, чтобы наборы с большим числом правил не перетягивали наблюдения в свой класс.

Алгоритм взвешенного голосования учитывает возможность правил иметь различную ценность. По этой причине каждому правилу φ_k^t приписывается вес $\alpha_k^t \geq 0$, и при голосовании берётся взвешенная сумма голосов [11]:

$$\Gamma_k(x) = \sum_{t=1}^{T_k} \alpha_k^t \varphi_k^t(x).$$

Веса нормируются на единицу: $\sum_{t=1}^{T_k} \alpha_k^t = 1$, для всех $k \in Y$. Исходя из этого, простое голосование является частным случаем взвешенного, когда веса одинаковы и равны $1/T_k$.

Вес правила помимо его информативности должен зависеть и от уникальности правила. Например, если имеется несколько хороших, но одинаковых или почти одинаковых правил, их суммарный вес должен быть сравним с весом столь же хорошего правила, не похожего на все остальные. Отсюда следует, что веса должны учитывать не только ценность правил, но и их различность.

Общий подход к настройке весов заключается сначала в поиске набора правил $\{\varphi_k^t(x)\}$, затем принятии их за новые (бинарные) признаки и построении в этом новом признаковом пространстве линейной разделяющей поверхности (кусочно-линейной при $|Y| > 2$). С этой целью можно применить логистическую регрессию, однослойный персептрон или метод опорных векторов.

Далее приводятся основные алгоритмы простого и взвешенного голосования правил: КОРА, ТЕМП, бустинг.

Алгоритм комбинаторного распознавания КОРА, предложенный М.М.Бонгардом и реализованный М.Н.Вайнцвайгом, строит набор конъюнктивных закономерностей [10].

Алгоритм базируется на следующих эвристических предположениях:

– множество термов подобрано так удачно, что среди конъюнкций ранга 2 или 3 уже находится достаточное количество информативных закономерностей;

– для поиска закономерностей можно применить полный перебор, так как ранг конъюнкций ограничен сверху числом 3;

–интерес представляют непротиворечивые закономерности.

Данный алгоритм формирует для каждого класса $k \in Y$ свой список конъюнкций R_k. Каждая конъюнкция содержит не более K термов, выбираемых из множества предикатов. Основой алгоритма является рекурсивная процедура наращивания конъюнкции, которая добавляет термы в конъюнкцию любыми способами и заносит в список закономерностей только наилучшие конъюнкции – удовлетворяющие критериям отбора $D_k(\varphi) \geq D_{\min}$ и $E_k(\varphi) \leq E_{\max}$, где $D_k(\varphi)$ – доля покрываемых положительных наблюдений, а $E_k(\varphi)$ – доля отрицательных наблюдений, среди всех покрытых.

Для перебора конъюнкций используют метод поиска в глубину. В процессе перебора избегают повторного просмотра конъюнкций, отличающихся только порядком следования термов, для этого конъюнкции наращиваются так, чтобы номера предикатов возрастали.

Для сокращения перебора процедура наращивания использует два приёма, основанных на критериях отбора конъюнкций [11].

Во-первых, конъюнкция φ перестаёт наращиваться, если она покрывает слишком мало наблюдений своего класса, $D_k(\varphi) < D_{\min}$, так как с увеличением числа термов количество покрываемых наблюдений может только уменьшиться.

Во-вторых, конъюнкция, удовлетворяющая критериям отбора и добавленная в список R_k, больше не наращивается. Таким образом, предпочтение отдаётся более коротким конъюнкциям.

На количество получаемых конъюнкций сильно влияют значения параметров $D_{\min}$ и $E_{\max}$. Если критерии отбора заданы слишком жёстко, алгоритм может вообще не найти ни одной конъюнкции. В противном случае, когда критерии слишком слабы, алгоритм будет тратить время на перебор и оценивание большого количества малоинформативных конъюнкций. На практике значения данных параметров подбирают экспериментальным путём.

Среди достоинств алгоритма комбинаторного распознавания можно отметить:

– конъюнкции, состоящие из 2-3 термов, легко интерпретируются в терминах предметной области;

– если короткие информативные конъюнкции имеются, они обязательно будут найдены, поскольку алгоритм осуществляет полный перебор.

Среди недостатков алгоритма комбинаторного распознавания можно выделить:

– если множество термов выбрано неудачно, то коротких информативных конъюнкций может просто не существовать;

– возможна невысокая обобщающая способность алгоритма, так как он не стремится к различности конъюнкций и обеспечению равномерного покрытия наблюдений выборки.

В алгоритме КОРА применяется стратегия перебора конъюнкций поиск в глубину, а в алгоритме ТЭМП, предложенном Г. С. Лбовым [42, 43], – поиск в ширину. Преимущество поиска в ширину заключается в том, что он работает немного быстрее, и в него легче встраивать различные эвристики, сокращающие перебор.

Суть алгоритма ТЭМП заключается в следующем. Процесс поиска закономерностей начинается с построения конъюнкций ранга 1. Для этого отбираются не более T_1 самых информативных термов из базового множества термов. Затем к каждому из отобранных термов добавляется по одному терму из оставшегося множества термов всеми возможными способами. Получается не более $T_1|B|$ конъюнкций ранга 2, из которых снова отбираются T_1 самых информативных, и так далее. На каждом шаге процесса делается попытка добавить один терм к каждой из имеющихся конъюнкций. Наращивание конъюнкций прекращается либо при достижении максимального ранга K, либо когда ни одну из конъюнкций не удаётся улучшить путём добавления терма.

Множество логических закономерностей, необходимых для классификации наблюдений конкретного класса, может содержать конъюнкции

различного ранга, поскольку лучшие конъюнкции, собранные со всех шагов, заносятся в данное множество закономерностей.

Компромисс между качеством и скоростью работы алгоритма обеспечивает параметр T_1. При $T_1 = 1$ алгоритм ТЭМП работает исключительно быстро и строит единственную конъюнкцию, добавляя термы по очереди. Фактически, он совпадает с жадным алгоритмом. При увеличении T_1 пространство поиска расширяется, алгоритм начинает работать медленнее, но находит больше информативных конъюнкций. Если задать $T_1 = \infty$, то алгоритм выполнит полный перебор. На практике выбирают максимальное значение параметра T_1, при котором поиск занимает приемлемое время. Недостатком алгоритма является жадная стратегия поиска – термы оптимизируются отдельно, и при подборе каждого терма учитываются только предыдущие, но не последующие термы.

С целью улучшения конъюнкций к ним применяют эвристические методы стабилизации и редукции.

При применении стабилизации конъюнкции становятся локально неулучшаемыми. Алгоритм в целом становится более устойчивым. Устойчивость алгоритма проявляется в том, что при незначительных изменениях в составе обучающей выборки он чаще находит одни и те же закономерности, следовательно, улучшается его способность обобщать эмпирические данные.

Достоинства алгоритма ТЭМП:

– алгоритм ТЭМП эффективнее алгоритма КОРА, так как он решает поставленную задачу за $O(KT_1|B|\ell)$ операций, а КОРА имеет трудоёмкость $O(|B|^K\ell)$;

– параметр T_1 позволяет обеспечивать компромисс между качеством конъюнкций и скоростью работы алгоритма;

– алгоритм ТЭМП можно применять в качестве генератора конъюнкций в других алгоритмах.

Недостатки алгоритма ТЭМП:

– отсутствие гарантии поиска самых лучших конъюнкций, особенно при малых значениях параметра T_1;

– возможна невысокая обобщающая способность алгоритма, так как он не стремится к различности конъюнкций и обеспечению равномерного покрытия наблюдений выборки.

Алгоритмы КОРА и ТЭМП имеют общий недостаток – они не стремятся увеличивать различность конъюнкций. Данная проблема решается в алгоритме бустинга. Бустинг предложили американские учёные Freund Y., Schapire R. E. как универсальный метод построения выпуклой комбинации классификаторов [89, 90].

Стратегия алгоритма бустинга заключается в том, что закономерности строятся последовательно, и после построения очередной закономерности веса покрытых ею наблюдений изменяются – уменьшаются у положительных и увеличиваются у отрицательных наблюдений. В результате каждая следующая закономерность стремится покрыть «наименее покрытые» наблюдения, оказавшиеся «сложными» для предыдущих закономерностей. Это способствует повышению различности закономерностей, более равномерному покрытию наблюдений и повышению обобщающей способности выпуклой комбинации закономерностей.

Приведенная выше стратегия напоминает алгоритм построения решающего списка. Разница в том, что алгоритме решающего списка единожды покрытое наблюдение исключается из дальнейшего рассмотрения, а в алгоритме бустинга каждое покрытие только изменяет вес наблюдения. Для реализации этой идеи остаётся понять, как именно должны вычисляться веса наблюдений и веса закономерностей на каждом шаге алгоритма. С этой целью приводится задача классификации с двумя классами, $Y = \{-1,+1\}$ и алгоритм взвешенного голосования [11]:

$$a_T(x) = sign\left(\Gamma_{+1}(x) - \Gamma_{-1}(x)\right) = sign\left(\sum_{t=1}^{T_{+1}} \alpha_{+1}^t \varphi_{+1}^t(x) - \sum_{t=1}^{T_{-1}} \alpha_{-1}^t \varphi_{-1}^t(x)\right), \alpha_k^t > 0, k \in Y.$$

Число ошибок алгоритма $a_{T+1}(x)$ после добавления закономерности φ_k на обучающей выборке X^ℓ:

$$Q_{T+1}(\varphi_k, \alpha) = \sum_{i=1}^{\ell} [y_i = k]\left[\Gamma_{y_i}(x_i) - \Gamma_{-y_i}(x_i) + \alpha\varphi_k(x_i) < 0\right] +$$

$$+ \sum_{i=1}^{\ell} [y_i \neq k]\left[\Gamma_{y_i}(x_i) - \Gamma_{-y_i}(x_i) - \alpha\varphi_k(x_i) < 0\right].$$

Минимум функционала $Q_{T+1}(\varphi_k, \alpha)$ достигается при:

$$\varphi_k^* = \arg\max_{\varphi} J_k^w(\varphi, X^\ell) \quad , \qquad J_k^w(\varphi, X^\ell) = \sqrt{p_k^w(\varphi)} - \sqrt{n_k^w(\varphi)};$$

$$\alpha^* = \frac{1}{2} \frac{p_k^w(\varphi_k^*)}{\max\{n_k^w(\varphi_k^*), \lambda\}},$$

где $\lambda \in (0,1)$, $p_k^w(\varphi) = \sum_{i=1}^{\ell} w_i [y_i = k][\varphi(x_i) = 1]$, $n_k^w(\varphi) = \sum_{i=1}^{\ell} w_i [y_i \neq k][\varphi(x_i) = 1]$, w_i – неотрицательные веса наблюдений.

После нахождения закономерности $\varphi_k(x)$ и вычисления соответствующего ей коэффициента α, нужно пересчитать веса наблюдений для следующего шага алгоритма:

$$w_i' = \begin{cases} w_i, & \varphi_k(x_i) = 0; \\ w_i e^{-\alpha}, & \varphi_k(x_i) = 1 \; u \; y_i = k; \\ w_i e^{\alpha}, & \varphi_k(x_i) = 1 \; u \; y_i \neq k. \end{cases}$$

Следовательно, при покрытии положительного наблюдения его вес уменьшается в e^α раз, а при покрытии отрицательного наблюдения увеличивается во столько же раз.

В ходе работы алгоритма имеет смысл проанализировать распределение весов наблюдений. Наблюдения с наибольшими весами являются самыми «сложными» для всех построенных закономерностей, поскольку почти не покрываются ими. Возможно, это выбросы – наблюдения, в описании которых допущены грубые ошибки. Исключение таких наблюдений, называемое

цензурированием выборки, как правило, повышает качество классификации. После процедуры цензурирования выборки построение закономерностей лучше начать заново, так как присутствие выбросов в обучающей выборке могло помешать поиску наиболее информативных закономерностей.

Следует отметить, что бустинг используется для построения набора конъюнкций в алгоритме SLIPPER (Simple Learner with Iterative Pruning to Produce Error Reduction), который является одним из эффективных алгоритмов классификации, основанных на взвешенном голосовании конъюнкций [82].

Среди достоинств алгоритма бустинга можно отметить:

– Обобщающая способность алгоритма достигается за счёт равномерного покрытия наблюдений закономерностями и различности правил.

– Интерпретируемость. Набор правил легко интерпретируется, если правил немного, и они имеют вид конъюнкций. Веса правил всегда положительны и показывают степень их важности для классификации.

– Возможность поиска выбросов в обучающей выборке.

Среди недостатков алгоритма бустинга можно отметить возможность эффекта переобучения, если на шаге построения закономерностей не удаётся находить достаточно хорошие закономерности. В таком случае резко возрастает число правил, необходимых для обеспечения корректности и их комбинация теряет свойство интерпретируемости.

1.4 Анализ программных систем для решения задач классификации

Следует отметить, что в настоящее время существует два пути создания программных средств, решающих практические задачи классификации. Первый путь связан с узкоспециализированными пакетами интеллектуального анализа данных, которые направлены на небольшой круг решаемых задач, а их алгоритмической базой является какой-либо один из альтернативных подходов, применяющий метод опорных векторов, метод потенциальных функций, и т.п.

Такие пакеты имеют определенные недостатки. Одним из них является тот факт, что использующиеся в пакетах методы не универсальны относительно размерностей задач, структурированности данных, наличия выбросов, противоречивости данных, и т.п. Вторым недостатком является то, что разработанные и настроенные на решение конкретных задач методы могут оказаться совершенно непригодными для остальных задач в данной области [22].

Другой путь создания программных средств основан на включении основных существующих алгоритмов. В данном случае повышаются шансы выбора из имеющихся алгоритмов такого, который обеспечит наиболее точное решение интересующих исследователя задач.

Приведем обзор основных программных систем для решения задач классификации.

Среди российских программных систем можно выделить: «Распознавание», «MultiNeuron».

«Распознавание» – универсальная программная система интеллектуального анализа данных [22]. Идеи универсальности и интеллектуальности положены в основу требований к системе. Под универсальностью системы понимается возможность ее использования на большом множестве задач, отличающихся друг от друга по размерностям, типу, качеству и структуре данных. Под интеллектуальностью понимается присутствие элементов самонастройки и способности успешного автоматического решения задач неквалифицированным пользователем. Для достижения данных требований проведены работы по объединению различных методов и алгоритмов в рамках единой системы, в частности, по унификации обозначений, форматов, пользовательских интерфейсов, единых форм представления результатов обработки данных.

Практическая реализация системы «Распознавание» выполнена в ООО «Центр технологий анализа и прогнозирования «Решения»» при поддержке

Фонда содействия развитию малых форм предприятий в научно-технической сфере. Система адаптирована для решения задач прогнозирования дискретных событий по выборкам многомерных данных.

В рамках системы «Распознавание» разработана библиотека программ, реализующих линейные, комбинаторно-логические, статистические, нейросетевые, гибридные методы прогноза, классификации и извлечения знаний из прецедентов, а также коллективные методы прогноза и классификации [22]. Например, многослойный персептрон, линейный дискриминант Фишера, метод опорных векторов, логические закономерности, алгоритмы вычисления оценок и т.д. Причем, используя графический интерфейс программной системы, можно легко добавлять или удалять любой метод при решении конкретной задачи.

Основные отличительные черты созданной программной системы от имеющихся аналогов состоят в следующем [22]:

– разнообразие имеющихся подходов и более широкие их практические возможности;

– способность автоматической обработки больших массивов данных;

– возможность решения задач прогноза и распознавания редких или уникальных событий и процессов;

– способность обработки разнотипных, частично-противоречивых и неполных данных.

Вся программная система разбита на модули, которые реализуют разработанный интерфейс. Основным модулем программы является Engine, отвечающий за управление всеми процессами, происходящими в системе, и связь между модулями. Вторым важным модулем является GUI (graphical user interface), который отвечает за графический интерфейс пользователя, то есть меню, панели инструментов, графическое представление данных и результатов работы методов. Завершают список группа однотипных модулей обработки данных, реализующих различные математические методы распознавания и

кластеризации, и группа модулей загрузки данных. Общая структура системы представлена на рисунке 1.1 [22].

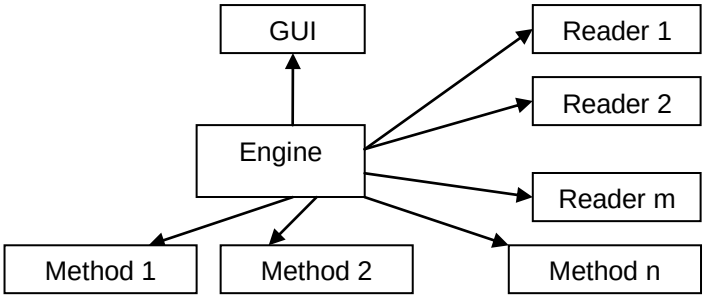

Рисунок 1.1 – Структура системы

Такая структура позволяет, во-первых, проводить независимую модификацию различных частей программы, и, во-вторых, добавлять математические методы и новые источники данных. Кроме того, фиксированный интерфейс обмена данных между модулями предоставляет конечному пользователю возможность написания своих методов и присоединения их к системе без участия ее разработчиков.

Программа «MultiNeuron» реализует методику обучения компьютерных нейронных сетей. Данная программа создана сотрудниками лаборатории моделирования неравновесных систем ВЦК СО РАН г. Красноярск [15, 55] и группы «НейроКомп».

Для решения задач классификации можно применить две из имеющихся в арсенале «MultiNeuron» программы – «классификатор» и «потенциатор». «Классификатор» использует для своей работы алгоритм двойственного функционирования back–propagation [15], «потенциатор» – метод потенциальных функций [2]. «Классификатор» хорош тем, что абсолютно весь процесс обучения проводится в автоматическом режиме. На входные нейроны нейронной сети подаются числовые сигналы, характеризующие входные данные. На выходные нейроны подаются ответы – выходные данные. Однако, «классификатор» дает невысокие результаты в случае существенного различия в количестве наблюдений каждого класса [15], поэтому в таких случаях лучше

применять «потенциатор». Обучение с использованием метода потенциальных функций предусматривает управление процессом двумя способами. Во-первых, может быть изменен параметр, отвечающий за изменение максимальной глубины потенциальных ям возле точек обучающегося множества. Во-вторых, можно варьировать соотношение весов классов, что приведет к изменению потенциала в точках этого класса.

В программе «MultiNeuron» применен способ прогнозирования с созданием нейросетевого консилиума [55, 61, 110, 111]. Этот метод предусматривает обучение нескольких нейросетей, решающих одну и ту же задачу. Окончательное решение принимается «большинством голосов» путем суммирования ответов всех нейросетей.

Программа «MultiNeuron» базируется на методике обучения компьютерных нейронных систем, модели которых обладают существенным недостатком – представляют модель черного ящика и не предоставляют исследователю информацию о том, каким образом найдено решение. Точность классификации с использованием нейронных сетей варьируется от 60% до 90% в зависимости от задачи и настройки параметров сети [14].

Среди зарубежных программных систем можно выделить: WEKA, RapidMiner.

Программа «WEKA» [115] (Waikato Environment for Knowledge Analysis) написана на языке Java в университете Вайкато (Новая Зеландия), предоставляет пользователю возможность предобработки данных, решения задач классификации, регрессии, кластеризации и поиска ассоциативных правил, а также визуализации данных и результатов. Программа очень проста в освоении, бесплатна и может быть дополнена новыми алгоритмами, средствами предобработки и визуализации данных.

Стартовое окно «Weka GUI Chooser», в котором будет предложено выбрать один из четырёх модулей программы, приведено на рисунке 1.2. Приведем краткое описание модулей программы.

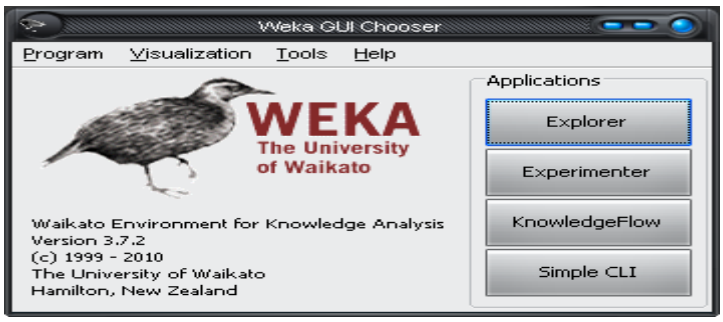

Рисунок 1.2 – Основное окно программы «WEKA»

Модуль «Explorer» – основной модуль программы (рис. 1.3), который позволяет загрузить и предобработать данные (вкладка Preprocess), решить задачу классификации (Classify), кластеризации (Claster), поиска ассоциаций (Associate), селекции признаков (Select Attributes) и визуализации (Visualize).

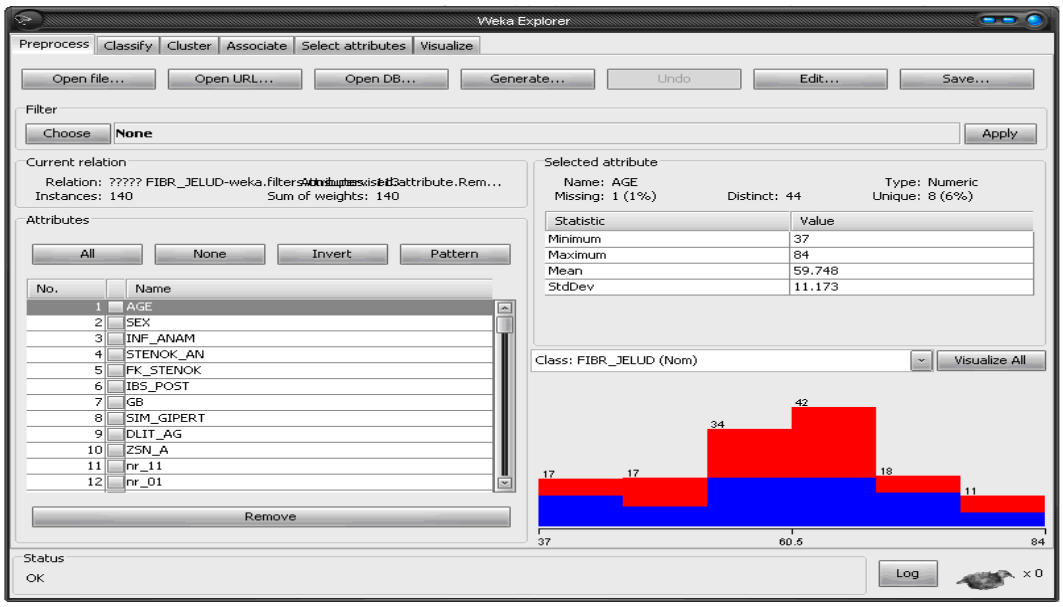

Рисунок 1.3 – Вкладка «Preprocess» модуля «Explorer»

Модуль «Experimenter» служит для проведения экспериментов и сравнивания результатов работы нескольких алгоритмов между собой на нескольких задачах.

Модуль «KnowledgeFlow» служит для демонстрации процесса решения задачи в виде графа. Для этого необходимо указать этапы решения в виде вершин, соединить их дугами, а затем запустить этот процесс на исполнение.

Модуль «Simple CLI» предоставляет интерфейс текстовой строки для ввода команд с целью увеличения возможностей пользователя при работе с программной системой.

Также в меню стартового окна «Weka GUI Chooser» есть несколько полезных функций [19]. Например, на вкладке «Tools» размещён редактор arff-файлов, простой модуль просмотра баз данных и программа для построения и обучения байесовских сетей, а на вкладке «Visualization» различные средства визуализации: визуализация данных, визуализация сохранённых ROC-кривых, визуализация деревьев решений и визуализация поверхностей решения, которая позволяет посмотреть на разделяющую поверхность классификатора.

Программа RapidMiner [106], написанная на языке Java, предназначена для решения задач машинного обучения и анализа данных. Пользователь моделирует весь желаемый процесс анализа данных в виде цепочки (графа) операторов и запускает его на выполнение. Цепочка операторов представляется в «RapidMiner» в виде интерактивного графа и в виде выражения на языке XML. Ко всем основным функциям имеется доступ через Java API и версию программы для командной строки.

Сейчас в системе реализовано более 400 операторов. Из них [19]:

– операторы обучения по прецедентам, в которых реализованы алгоритмы классификации, регрессии, кластеризации и поиска ассоциаций, а также мета-алгоритмы;

– операторы системы WEKA;

– операторы предобработки (дискретизация, фильтрация, заполнение пропусков, уменьшение размерности и т.д.);

– операторы работы с признаками (селекция и генерация признаков);

– мета-операторы (например, оператор оптимизации по нескольким параметрам);

– операторы оценки качества (скользящий контроль и т.д.);

– операторы визуализации;

– операторы загрузки и сохранения данных.

Имеется также возможность по добавлению в систему своих операторов. На рисунке 1.4 приведено основное окно программы «RapidMiner».

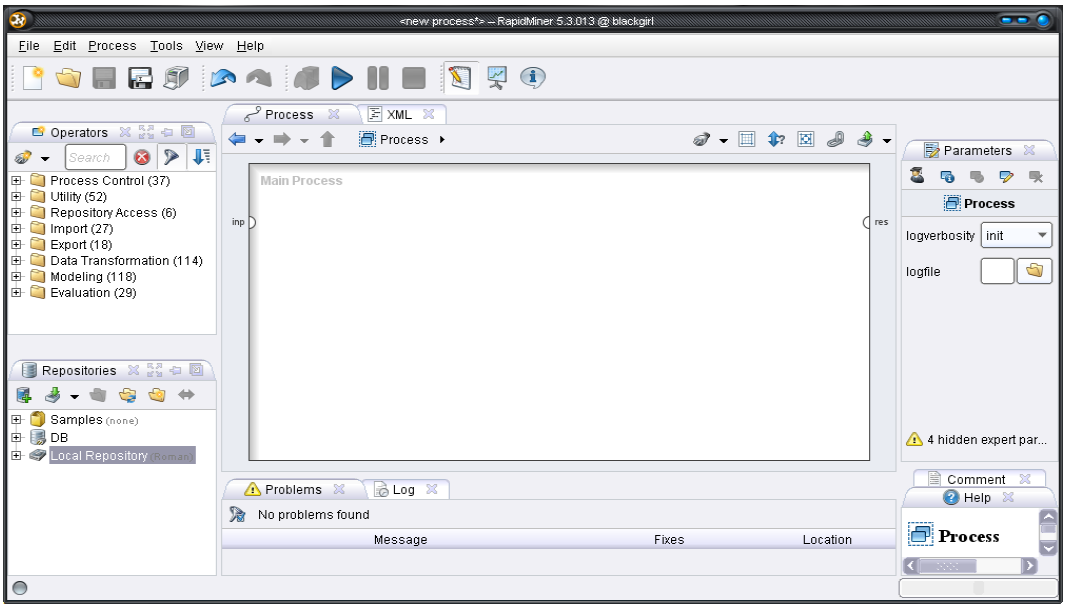

Рисунок 1.4 – Основное окно программы «RapidMiner»

Rapidminer обладает графическими средствами для визуализации данных и моделей: дендрограммы, деревья решений, каталоги кластеров. Также в программной системе реализована концепция многоуровневого представления данных, которая обеспечивает эффективную работу с данными.

Выводы

В первой главе рассмотрены основные логические алгоритмы классификации, алгоритмы поиска закономерностей в форме конъюнкций для них, проведен анализ программных систем, решающих практические задачи классификации.

Выяснено, что предикат является закономерностью, если он покрывает достаточно много наблюдений одного класса и почти не покрывает наблюдения другого класса. Чем больше наблюдений своего класса по сравнению с

наблюдениями всех других классов покрывает закономерность, тем она более информативна. Для определения какая из закономерностей является информативнее, приводятся статистический и эвристический критерии информативности.

Информативные закономерности служат исходными данными для построения логических алгоритмов классификации. Как правило, закономерности строятся в виде конъюнкций, ведь конъюнкции небольшой ранга обладают важным преимуществом – они имеют вид привычных для человека логических высказываний и легко поддаются интерпретации. Среди алгоритмов синтеза конъюнкций можно выделить: «градиентный» алгоритм, жадный алгоритм, случайный локальный поиск, генетические алгоритмы, каждый из которых имеет свои особенности, преимущества и недостатки.

Изучены основные алгоритмы логической классификации: решающие списки, решающие деревья, алгоритмы простого и взвешенного голосования правил. Приводятся их преимущества и недостатки, способы построения. Для решающих списков и решающих деревьев имеется один существенный недостаток, заключающийся в том, что каждое наблюдение классифицируется только одним правилом, что не позволяет правилам компенсировать ошибки друг друга. Следовательно, для классификации наиболее приемлем алгоритм, основанный на голосовании правил. Также важную роль играет построение классификатора, обладающего хорошей обобщающей способностью.

Проанализировав программные системы, определено два пути развития программных средств: узкоспециализированные пакеты, которые направлены на небольшой круг решаемых задач, а их алгоритмической базой является какой-нибудь один из альтернативных подходов к классификации, и программные средства, основанные на включении основных существующих подходов.

2 МЕТОД ЛОГИЧЕСКОГО АНАЛИЗА ДАННЫХ И ЕГО МОДИФИКАЦИИ

2.1 Описание подхода

В диссертационной работе рассматривается задача классификации следующего вида [13]. Имеется выборка данных, которая состоит из двух непересекающихся множеств Ω^+ и Ω^- n-мерных векторов, принадлежащих соответственно положительному или отрицательному классу. Компоненты вектора, называемые также признаками, могут быть как численными или номинальными, так и бинарными. Задача состоит в том, чтобы для некоторого нового наблюдения, являющегося также вектором n переменных, определить, к какому классу он принадлежит.

В основе предлагаемого подхода к классификации данных лежит метод, происходящий из теории комбинаторной оптимизации и называемый логическим анализом данных (*Logical Analysis of Data – LAD*) [92]. Этот метод успешно использовался для решения ряда задач из различных областей [27, 67, 68, 93, 94]. Основная идея метода заключается в совместном использовании действий по «дифференцированию» и «интегрированию», производимых на области пространства исходных признаков, содержащей заданные положительные и отрицательные наблюдения. На шаге «дифференцирования» определяется семейство малых подмножеств, обладающих характерными положительными и отрицательными чертами. На шаге «интегрирования» формируемые определенным образом объединения этих подмножеств рассматриваются как аппроксимации областей пространства признаков, содержащих положительные и, соответственно, отрицательные наблюдения [40].

Последовательные элементы метода [94, 50]:

а) Для исключения избыточных переменных в исходной выборке данных во множестве переменных определяется некоторое подмножество *S*, используя

которое можно отличать положительные наблюдения от отрицательных. Далее для работы метода используются проекции Ω_S^+ и Ω_S^- множеств Ω^+ и Ω^- на *S*.

б) Множество Ω_S^+ покрывается семейством однотипных подмножеств уменьшенного пространства, каждое из которых имеет значительное пересечение с Ω_S^+, но не пересекается с Ω_S^-, либо допускается небольшое пересечение с Ω_S^- для большего увеличения пересечения с Ω_S^+. Такие подмножества называются «положительными закономерностями». Аналогично множество Ω_S^- покрывается «отрицательными закономерностями».

в) Определяется подмножество положительных закономерностей, объединение которых покрывает все наблюдения Ω_S^+, и подмножество отрицательных закономерностей, объединение которых покрывает все наблюдения Ω_S^-.

г) Положительный или отрицательный характер некоторого наблюдения, покрываемого объединением двух подмножеств, определяется с помощью классификатора, основанного на этих подмножествах.

2.2 Бинаризация признаков

Рассматриваемый метод предназначен для работы с выборками данных, в которых признаки принимают бинарные значения. Так как исходная выборка может состоять из разнотипных признаков, необходимо воспользоваться способами бинаризации.

Суть одного из простейших способов бинаризации заключается в том, что каждой метрической переменной ставится в соответствие несколько бинарных переменных. Бинарная переменная принимает значение 1, если значение соответствующей метрической переменной принимает значение выше определенного порога, и наоборот. Такой способ в [53] называется «единичным». Его недостатком является то, что существует большое число комбинаций бинарных переменных, которым не соответствуют точки в

исходном пространстве $(2^n - n - 1)$. Этот недостаток затрудняет использование такого способа для кодирования переменных критериальной функции при решении задач оптимизации, так как большое число решений будут недопустимыми. Но в данном случае, для классификации, это не имеет значения, так как бинарные переменные получаются путем кодирования заданных метрических переменных. Основным же преимуществом такого способа является соответствие расстояний в исходном и бинарном пространствах. Это значит, что точки, близкие в исходном пространстве, являются близкими и в бинаризованном. А это, в свою очередь, позволяет еще в процессе бинаризации минимизировать число порогов, ставя в соответствие близким значениям исходной переменной одно и то же значение в бинарном пространстве (при условии, что положительные и отрицательные множества наблюдений останутся непересекающимися) [45].

Существует другой способ бинаризации, приведенный в [11, 39].

Произвольный признак f: $X \to D_f$ создает термы, проверяющие попадание значения $f(x)$ в определённые подмножества множества D_f. В [11] приводятся типичные конструкции такого вида.

– Если f – номинальный признак:

$$\beta(x) = [f(x) = d], d \in D_f;$$

$$\beta(x) = [f(x) \in D'], D' \subset D_f.$$

– Если f – порядковый или количественный признак:

$$\beta(x) = [f(x) \le d], d \in D_f;$$

$$\beta(x) = [d \le f(x) \le d'], d, d' \in D_f, d < d'.$$

В случае количественных признаков f: $X \to R$ необходимо брать только такие значения порогов d, которые по-разному разделяют выборку X^ℓ. Если исключить тривиальные разбиения, обращающие $\beta(x)$ в 0 или 1 на всей выборке, то таких значений окажется не более $\ell - 1$. Например, можно взять пороги вида:

$$d_i = \frac{f(i)+f(i+1)}{2}, \qquad f^{(i)} \neq f^{(i+1)},\ i = 1,\ldots,\ell-1, \tag{2.1}$$

где $f^{(1)} \le \ldots \le f^{(\ell)}$ – последовательность значений признака f на наблюдениях выборки $f(x_1),\ldots, f(x_\ell)$, упорядоченная по возрастанию (Рис. 2.1).

Рисунок 2.1 – Вариационный ряд значений признака *f*(x) и пороги d_i

Если полученные термы в дальнейшем будут использоваться для синтеза конъюнкций, то для сокращения перебора необходимо сразу выбрать из них самые информативные. Когда признаки порядковые и количественные такая задача решается посредством оптимального разбиения диапазона значений признака на зоны. Ниже приводится процесс такого разбиения.

Пусть f: X→R – количественный признак, $d_1,\ldots,d_r$ – возрастающая последовательность порогов. Зонами значений признака f будем называть термы вида:

$$\varepsilon_0(x) = [f(x) < d_1];$$
$$\varepsilon_s(x) = [d_s \le f(x) < d_{s+1}],\ s = 1,\ldots,r-1;$$
$$\varepsilon_r(x) = [d_r \le f(x)].$$

Например, жадный алгоритм слияния зон начинает с разбиения на «мелкие зоны». Пороги определяются по формуле (2.1) и проходят между всеми парами точек x_{i-1}, x_i, ровно одна из которых принадлежит классу «*k*».

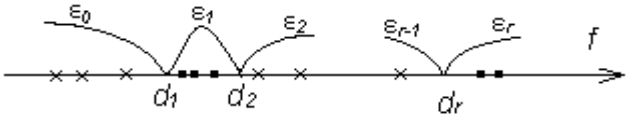

Рисунок. 2.2 – Начальное разбиение на зоны положительных (×) и отрицательных (▪) наблюдений

Начальное разбиение приведено на рисунке 2.2 и состоит из чередующихся зон «только k – только не k». Потом зоны укрупняются посредством слияния троек соседних зон. Сливаются именно тройки, поскольку слияние пар приводит к нарушению чередования «k – не k», в результате некоторые «мелкие зоны» могут так и остаться неслитыми. Критериями остановки алгоритма слияния зон являются момент получения заданного количество зон r или момент превышения информативности некоторых исходных зон ε_{i-1}, ε_i и ε_{i+1} над информативностью слитой зоны $\varepsilon_{i-1} \vee \varepsilon_i \vee \varepsilon_{i+1}$. Выбор тройки для слияния обусловлен достижением максимального выигрыша информативности при данном слиянии.

2.3 Построение опорного множества

Важность отбора признаков общеизвестна в статистике, и долгое время признавалась главным элементом исследований в машинном обучении, выявлении экспертных знаний, добычи данных, и других связанных областях. Всесторонний обзор многих существующих методов, начиная с 1970 гг. представлен обзором в 1997 M.Dash и H.Liu [84], монографиями 1998 [99], [100] H.Liu, H. Motoda. Некоторые из классических техник статистики (анализ главных компонент) являются весьма подходящими к области отбора признаков. Среди методов, специально созданных для этой цели, следует отметить и полный перебор эвристик (последовательный прямой/обратный выбор, метод ветвей и границ [84]), использование искусственных нейронных сетей [98, 112], метод опорных векторов [72, 79], генетические алгоритмы [80]. Критерии для вычисления ранга каждого признака в наборе с целью создания пула выбранных признаков рассмотрены в [67].

Представление слишком большого числа признаков в выборке может создавать огромную вычислительную сложность. Например, в геномике и протеомике, двух наиболее развивающихся областях биоинформатики, где выражение уровней яркости тысяч или десятков тысяч генов или белков

включено в набор данных, несмотря на факт, что очень маленького подмножества этих признаков достаточно для превосходного разделения положительных и отрицательных наблюдений [65, 66].

Одним из усложняющих факторов в извлечении значимого подмножества признаков является тот факт, что существует явное различие между значимостью индивидуальных признаков и значимостью набора признаков. Приведем типичный пример: высоко значимый набор из 7 пептидов, идентифицирующих рак яичников, включает только 4 пептида, уровень которых сильно коррелирует с присутствием рака яичника [65].

Необходимо разработать подходы для идентификации подмножества признаков, которые могут разделить с высокой точностью положительные и отрицательные наблюдения.

Предлагается один из таких подходов, основанный на выборе подмножества признаков путем построения оптимизационной модели в виде задачи комбинаторной оптимизации.

Множество S признаков называется опорным множеством, если проекция Ω_S^+ множества Ω^+ на S не пересекается с проекцией Ω_S^- множества Ω^- на S. Множество всех признаков является опорным, так как изначально Ω^+ и Ω^- не пересекаются. Опорное множество будем называть минимальным, если, исключив из него любую оставшуюся переменную, получается выборка данных, в которой некоторые положительное и отрицательное наблюдения совпадают.

Чтобы выявить минимальное опорное множество, в соответствие каждому признаку x_i, $i = 1,\ldots,t$ бинарной выборки ставится новая бинарная переменная y_i, которая равна 1, если x_i принадлежит опорному множеству, и равна 0, если нет. Следует обозначить $U = (u_1,u_2,\ldots,u_t)$ – бинарный вектор, ассоциированный с положительным наблюдением, и $V = (v_1, v_2,\ldots, v_t)$ – с отрицательным наблюдением. Вводится переменная:

$$w_i(U,V)=\begin{cases}1, u_i \neq v_i,\\ 0, u_i = v_i.\end{cases}$$

Условие раздельности множеств Ω_s^+ и Ω_s^- эквивалентно требованию выполнения неравенства $\sum w_i(U,V)y_i \geq 1$ для любых $U \in \Omega_S^+$ и $V \in \Omega_S^-$.

Для того чтобы выборка данных была более устойчива к ошибкам измерений, в результате которых они были получены, это условие следует усилить, заменив число 1 в правой части неравенства на некоторое целое число *d*. Это означает, что положительное и отрицательное наблюдения должны отличаться не менее чем *d* признаками.

Таким образом, задача минимизации опорного множества может быть сформулирована как задача условной псевдобулевой оптимизации:

$$\sum_{j=1}^{t} y_j \to \min,$$

$$\sum_{i=1}^{t} w_i(U,V)y_i \geq d \text{ для любых } U \in \Omega_S^+ \text{ и } V \in \Omega_S^-,$$

где $y \in \{0,1\}^t$.

Целевая функция задачи является унимодальной монотонной псевдобулевой функцией [69], т.е. имеет единственный безусловный минимум, находящийся в точке y^0 = (0,0,…,0), и возрастает при удалении от точки минимума (при смене любой компоненты с 0 на 1). Функция ограничения является также унимодальной и монотонной псевдобулевой функцией, причем заданной алгоритмически, так как для ее вычисления необходимо перебрать все возможные пары положительных и отрицательных наблюдений.

Альтернативным подходом для отбора признаков является разработанная алгоритмическая процедура получения усеченного набора, которая основана на оценке важности признаков [28, 38].

Оценка важности признака – это частота его включения в закономерности, задействованные в классификаторе [73]. Таким образом, чем чаще признак встречается в получаемых закономерностях, тем он более важен.

А те признаки, которые не встречаются или почти не участвуют в построении закономерностей, важными не являются.

Алгоритмическая процедура получения усеченного набора признаков состоит из четырех этапов.

Первым этапом процедуры получения усеченного набора признаков является проведение классификации на полном наборе признаков с целью определения важности каждого признака.

Вторым этапом является установление исследователем порога важности, относительно которого можно судить о важности конкретного признака.

Третьим этапом является ранжирование признаков по важности и выделение тех признаков, значение важности которых оказалось ниже установленного порога.

Четвертый этап состоит в исключении выделенных на третьем этапе признаков из рассмотрения. Оставшиеся признаки составляют усеченный набор. Таким образом, исследователь, варьируя порог важности, получает различные усеченные наборы признаков, которые в дальнейшем используются на этапе формирования закономерностей.

2.4 Формирование закономерностей

В основе рассматриваемого подхода лежит понятие закономерности. Положительной закономерностью называется подкуб пространства булевых переменных B_2^t, который пересекается с множеством Ω_s^+ и не имеет общих элементов с множеством Ω_s^-. Отрицательная закономерность задается аналогично. Положительная ω-закономерность для $\omega \in \{0,1\}^t$ – это закономерность, содержащая в себе точку ω. Для каждой точки $\omega \in \Omega_s^+$ найдем максимальную ω-закономерность, то есть покрывающую наибольшее число точек Ω_s^+.

Соответствующий подкуб задается с помощью переменных y_j:

$$y_j = \begin{cases} 1, \text{ если } i\text{ - ый признак зафиксирован в подкубе,} \\ 0, \text{ в противном случае.} \end{cases}$$

То есть путем фиксирования l переменных исходного куба размерностью t получаем подкуб размерностью $(t - l)$ и с числом точек 2^{t-l}.

Условие, говорящее о том, что положительная закономерность не должна содержать ни одной точки Ω_s^-, требует, чтобы для каждого наблюдения $\rho \in \Omega_s^-$ переменная y_j принимала значение 1 по меньшей мере для одного j, для которых $\rho_j \neq \omega_j$:

$$\sum_{\substack{j=1 \\ \rho_j \neq \omega_j}}^{t} y_j \geq 1 \text{ для любого } \rho \in \Omega_s^-.$$

Усиление ограничения для повышения устойчивости к ошибкам производится путем замены числа 1 в правой части неравенства на целое положительное число d.

С другой стороны, позитивное наблюдение $\sigma \in \Omega_s^+$ будет тогда входить в рассматриваемый подкуб, когда переменная y_j принимает значение 0 для всех индексов j, для которых $\sigma_j \neq \omega_j$. Таким образом, число положительных наблюдений, покрываемых ω-закономерностью, может быть вычислено как:

$$\sum_{\sigma \in \Omega_s^+} \prod_{\substack{j=1 \\ \sigma_j \neq \omega_j}}^{t} (1 - y_j).$$

Таким образом, для формирования закономерностей получается задача условной псевдобулевой оптимизации с алгоритмически заданными функциями [46]:

$$\sum_{\sigma \in \Omega_S^+} \prod_{\substack{j=1 \\ \sigma_j \neq \omega_j}}^{t} (1 - y_j) \to \max \qquad (2.2)$$

$$\sum_{\substack{j=1 \\ \rho_j \neq \omega_j}}^{t} y_j \geq d \text{ для любого } \rho \in \Omega_s^-, y \in \{0,1\}^t. \qquad (2.3)$$

Целевая функция (2.2) и функция ограничения (2.3) в этой задаче являются унимодальными монотонными псевдобулевыми функциями.

Аналогично формулируется задача нахождения максимальных отрицательных закономерностей.

Каждая найденная закономерность характеризуется покрытием – числом захватываемых наблюдений своего класса, и степенью – числом фиксированных переменных, которые определяют эту закономерность. Согласно приведенной выше оптимизационной модели (2.2)–(2.3) полученные закономерности не покрывают ни одного наблюдения другого класса (из обучающей выборки).

Наибольшую ценность представляют закономерности, которые имеют наибольшее покрытие. Чем больше покрытие, тем лучше закономерность отображает образ класса.

Специфика описанной выше задачи классификации состоит в том, что база данных имеет большое число неизмеренных значений (пропущенных данных), а сделанные измерения могут быть неточны либо ошибочны. Известно, что погрешность напрямую связана с точностью измерений, характеризующей близость результатов измерений к истинным значениям измеренных величин. Точность измерений может быть большей или меньшей, в зависимости от выделенных ресурсов (затрат на средства измерений, проведение измерений, стабилизацию внешних условий и т. д.). Понятно, что она должна быть приемлемой для выполнения поставленной задачи, но не более, так как дальнейшее повышение точности приведет к излишним финансовым затратам [74].

Числовые множества данных имеют погрешности в значениях числовых признаков из-за неточных инструментов, методов измерений или человеческих ошибок. Шумы и выбросы приводят к тому, что наблюдения различных классов «накладываются» друг на друга, попадая в «область» противоположного класса. В результате вычисляемые закономерности получаются с большей степенью и с существенно меньшим покрытием, чем, если бы выбросов и неточностей не было, а классификатор состоит из большого

числа маленьких закономерностей (с малым покрытием). Это не позволяет построить эффективный классификатор с «хорошо интерпретируемыми» правилами, в которых участвует небольшое число признаков, и с высокой точностью классификации [13].

Для повышения устойчивости метода к выбросам следует ослабить ограничение (2.3). Тогда степень вычисляемых закономерностей уменьшится, а покрытие увеличится.

Ограничение оптимизационной модели будет выглядеть следующим образом [46]:

$$\sum_{\rho\in\Omega_S^-} z_\rho \le D, \text{ где } z_\rho = \begin{cases} 0, если \sum\limits_{\substack{j=1 \\ \rho_j \neq \omega_j}}^{t} y_j \ge d \\ 1, в\ противном\ случае \end{cases} \quad (2.4)$$

D – число наблюдений другого класса, которым допускается быть покрытыми закономерностью (целое неотрицательное число).

Функции (2.2)–(2.4) построенной модели оптимизации задаются алгоритмически, т.е. вычисляются через определенную последовательность операций. Для решения задачи оптимизации используются алгоритмы оптимизации, основанные на поиске граничных точек допустимой области [47, 48, 70, 71]. Эти алгоритмы были разработаны специально для этого класса задач и основаны на поведении монотонных функций модели оптимизации в пространстве булевых переменных. Алгоритмы поиска граничных точек являются поисковыми, т.е. не требуют задания функций в явном виде, с помощью алгебраических выражений, а используют вычисления функций в точках.

Согласно модели (2.2, 2.4) наиболее предпочтительными являются закономерности с наибольшим покрытием. Следствием этого является то, что формируемые закономерности имеют маленькую степень, т.е. состоят из небольшого числа термов и используют лишь малую часть признаков. Закономерности с маленькой степенью соответствуют большим областям в

пространстве признаков. Это приводит к возможному покрытию наблюдений другого класса (отсутствующих в обучающей выборке) и повышению количества неверно классифицированных наблюдений. Данная особенность влияет на информативность закономерности, уменьшая ее. Поэтому с целью повышения информативности предлагается алгоритмическая процедура наращивания закономерностей. Она применяется к каждой построенной закономерности и заключается в максимальном увеличении степени данных закономерностей при условии сохранения покрытия:

$$\sum_{j=1}^{t} y_j \to \max$$

$$fc(Y) = fc'(Y),$$

где $fc(Y)$ – значение целевой функции (покрытие) для закономерности до процедуры наращивания, $fc'(Y)$ – значение целевой функции для закономерности после процедуры наращивания.

Таким образом, применение процедуры наращивания закономерностей позволяет повысить их информативность путем уменьшения покрытия правилами наблюдений другого класса, тем самым, способствуя повышению точности принимаемых решений классификатором.

На следующем этапе метода решается проблема построения адекватного классификатора, который смог бы классифицировать вновь поступающее наблюдение, т.е. наблюдение, не принимавшее участие при его построении.

2.5 Построение классификатора

Результатом предыдущего этапа метода является семейство максимальных закономерностей, число которых ограничено мощностью выборки данных $|\Omega^+ \cup \Omega^-|$. Классификатор состоит из полного набора положительных и отрицательных закономерностей.

Чтобы классифицировать новое наблюдение, воспользуемся следующим решающим правилом [13]:

1) Если наблюдение удовлетворяет условиям одной или нескольких положительных закономерностей и не удовлетворяет условиям ни одной из отрицательных, то оно классифицируется как положительное.

2) Если наблюдение удовлетворяет условиям одной или нескольких отрицательных закономерностей и не удовлетворяет условиям ни одной из положительных, то оно классифицируется как отрицательное.

3) Выбор алгоритма голосования:

а) Алгоритм простого голосования. Если наблюдение удовлетворяет условиям p' из p положительных закономерностей и q' из q отрицательных, то знак наблюдения определяется как $p'/p - q'/q$.

б) Алгоритм взвешенного голосования [33]. Если наблюдение удовлетворяет условиям p' из p положительных закономерностей и q' из q отрицательных, то знак наблюдения определяется как $\sum_{n=1}^{p'} a_n - \sum_{n=1}^{q'} b_n$, где a и b – веса для положительных и отрицательных закономерностей. Вес для n-й положительной закономерности находиться по формуле: $a_n = \frac{H_n}{\sum_{n=1}^{p} H_n}$, где H_n – информативность n-й положительной закономерности, которая вычисляется по критерию бустинга (2.6) [34]. Сумма весов всех положительных закономерностей равна единице: $\sum_{n=1}^{p} a_n = 1$. Аналогично вычисляется информативность и вес для n-й отрицательной закономерности.

4) В случае, если наблюдение не удовлетворяет условиям ни одной закономерности, положительной или отрицательной, то оно относится к классу, имеющему наименьшую цену ошибки.

2.6 Модификации для метода логического анализа данных

Формирование закономерностей и построение классификатора являются ключевыми этапами метода логического анализа данных. Именно от реализации этих этапов напрямую зависит качество классификации. Поэтому при разработке модификаций для метода предлагаются алгоритмические процедуры, связанные с реализацией данных этапов.

Итак, на этапе формирования закономерностей предлагается подход для задания целевой функции модели оптимизации при построении закономерностей, который основывается на модификации целевой функции (2.2) для увеличения различности правил в классификаторе. Данный подход, базируется на том, что при голосовании закономерности должны быть различны, иначе они становятся бесполезны для классификации.

Согласно целевой функции (2.2), каждая формируемая закономерность максимизирует свое покрытие, захватывая наблюдения, которые являются типичными представителями класса, а нетипичные наблюдения класса остаются непокрытыми и в классификаторе отсутствуют закономерности, учитывающие их. Поэтому мы получаем набор сходных закономерностей для класса, тем самым, снижая качество классификации. Для получения классификатора с более высокой различностью правил, которая позволяет выделять существенно различные подмножества наблюдений, предлагается модифицировать целевую функцию (2.2) для нахождения положительных закономерностей следующим образом:

$$\sum_{\sigma\in\Omega_S^+} K_\sigma \cdot \prod_{\substack{j=1 \\ \sigma_j\neq\omega_j}}^{t}(1-y_j) \to \max, \qquad (2.5)$$

где K_σ – вес позитивного наблюдения $\sigma\in\Omega_s^+$, который уменьшается при покрытии данного наблюдения, тем самым, понижая свой приоритет участия в формировании следующей закономерности в пользу непокрытых наблюдений.

Аналогично формируется целевая функция модели оптимизации для нахождения отрицательных закономерностей.

Для того чтобы использовать модель оптимизации с целевой функцией (2.5) для формирования закономерностей необходимо задать начальные веса для всех наблюдений и правило изменения весов для наблюдений, которые приняли участие при формировании текущей закономерности. Начальные веса предлагается выбрать равными 1 для каждого наблюдения в обучающей выборке. Правило изменения веса для наблюдения, который принял участие при формировании текущей закономерности:

$$K_{i+1} = \max\left[0, K_i - \frac{1}{N_{\max}}\right],$$

где K_i, K_{i+1} – веса покрываемого наблюдения при формировании текущей и следующей закономерностей, $N_{\max}$ – параметр, задаваемый исследователем, который означает максимальное количество закономерностей, которыми может покрываться наблюдение обучающей выборки в классификаторе.

Таким образом, используя оптимизационную модель с целевой функцией (2.5) для построения закономерностей, получаются логические правила, покрывающие существенно различные подмножества наблюдений. В дальнейшем из них выбирают те, у которых значения целевой функции больше нуля, и объединяют их в классификаторе.

На следующем этапе метода решается проблема построения адекватного классификатора, который смог бы верно отнести новое наблюдение, т.е. наблюдение, не принимавшее участие при его построении, к тому или иному классу.

Ввиду того, что объем выборки данных может быть значителен, встает вопрос о сокращении числа закономерностей, так как это число равно в исходном классификаторе мощности обучающей выборки данных $|\Omega^+ \cup \Omega^-|$. То есть необходимо определить классификатор, состоящий из некоторого числа закономерностей таким образом, чтобы он был способен классифицировать те

же наблюдения, которые можно классифицировать с помощью полной системы закономерностей.

В данной диссертационной работе предлагаются алгоритмические процедуры сокращения количества закономерностей в исходном классификаторе:

– выбор базовых наблюдений для формирования закономерностей [36, 37];

– построение классификатора как композиции информативных закономерностей [34, 35].

Чтобы реализовать алгоритмическую процедуру выбора базовых наблюдений для формирования закономерностей, необходимо выполнить ряд последовательных действий. Во-первых, на основе наблюдений обучающей выборки получить центроиды для каждого класса, используя алгоритм «k-средних». Согласно алгоритму «k-средних» следует определить каждое наблюдение обучающей выборки к одному из k кластеров таким образом, чтобы каждый кластер был представлен центроидом соответствующих наблюдений, а расстояние от каждого наблюдения до центроида своего кластера было меньше, чем до центроидов всех других кластеров. Данный алгоритм позволяет найти набор центроидов, который наилучшим образом представляет распределение наблюдений обучающей выборки.

Алгоритм состоит из следующих шагов, представленных в [1]:

Шаг 1. Выбираются k начальных центроидов $y_1(1)$, $y_2(2),\ldots,y_k(1)$. Начальные центроиды выбираются произвольно, например, первые k наблюдений из обучающей выборки.

Шаг *l*. На *l*-м шаге итерации множество наблюдений $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}$ распределяется по k кластерам по следующему правилу:

$$\mathbf{x} \in T_j(l), \text{если} \left\| \mathbf{x} - \mathbf{y}_j(l) \right\| < \left\| \mathbf{x} - \mathbf{y}_i(l) \right\|$$

для всех i = 1,2,…,k, $i \neq j$, где $T_j(l)$ – множество наблюдений, входящих в кластер с центроидом $y_j(l)$. В случае равенства решение принимается произвольным образом.

Шаг l+1. На основе результатов шага l определяются новые центроиды кластеров $y_j(l+1)$, j = 1,2,…,k, исходя из условия, что сумма квадратов расстояний между всеми наблюдениями, принадлежащими множеству $T_j(l)$, и новым центроидом данной кластера должна быть минимальной.

Центроид $y_j(l+1)$, обеспечивающий минимизацию $J_j = \sum_{\mathbf{x} \in T_j(l)} \left\| \mathbf{x} - \mathbf{y}_j(l+1) \right\|^2$, j = 1,2,…,k, является выборочным средним, определенным по множеству $T_j(l)$. Следовательно, новые центроиды кластеров определяются как:

$$\mathbf{y}_j(l+1) = \frac{1}{N_j} \sum_{x \in T_j(l)} x, \ j = 1,2,\ldots,k,$$

где N_j – число выборочных наблюдений, входящих во множество $T_j(l)$. Очевидно, что название алгоритма «k-средних» определяется способом, принятым для последовательной коррекции назначения центроидов кластеров.

Равенство $y_j(l+1) = y_j(l)$ при j = 1,2,…,k является условием сходимости алгоритма, и при его достижении выполнение алгоритма заканчивается. Полученные множества $T_j(l)$, j = 1,2,…,k, и образуют искомые кластеры. В противном случае последний шаг повторяется.

Данный алгоритм применяется для разделения на кластеры наблюдений обучающей выборки каждого класса. В результате его использования получается для каждого класса свой набор центроидов.

Во-вторых, добавить полученные наборы центроидов к наблюдениям обучающей выборки. В-третьих, использовать центроиды в качестве базовых наблюдений для формирования закономерностей.

Таким образом, реализуя описанную выше эвристическую процедуру, получаем новый классификатор, состоящий из меньшего числа закономерностей. Число закономерностей в классификаторе будет равно суммарному количеству центроидов, полученных для каждого класса. Ясно,

что точность классификации зависит от числа центроидов для каждого класса, поэтому необходимо провести несколько экспериментов с разными по количеству наборами центроидов, чтобы определить зависимость точности классификации от количества центроидов для каждого класса.

Процедура выбора базовых наблюдений для формирования закономерностей реализуется до момента построения классификатора, тем самым, сокращая трудоемкость его построения, так как значительно уменьшается количество формируемых закономерностей, однако точность классификации, как правило, немного снижается. Для устранения этого недостатка в работе предлагается другой подход с целью сокращения количества закономерностей в исходном классификаторе. Необходимо построить классификатор, количество закономерностей которого равно мощности обучающей выборки данных, и сократить данное количество закономерностей при сохранении высокой точности классификации. Для реализации этого подхода предлагается процедура построения классификатора как композиции информативных закономерностей, базирующая на понятии их информативности.

В литературе для измерения информативности закономерности существует несколько критериев [91]. В данной работе предлагается использовать критерий бустинга, т.к. он адекватно оценивает информативность закономерности и прост для вычисления:

$$H(p,n)=\sqrt{p}-\sqrt{n}, \tag{2.6}$$

где p – количество наблюдений своего класса, которые захватывает построенная закономерность; n – количество наблюдений другого класса, которые захватывает построенная закономерность.

Изначально, классификатор содержал все закономерности, которые строились относительно каждого наблюдения обучающей выборки. В результате, если объем обучающей выборки увеличивается, то и размер набора правил классификатора возрастает. Причем построенные закономерности

характеризуются разной информативностью. Закономерности, покрывающие мало наблюдений, статистически не надежны – среди них слишком много таких, которые допускают на независимых контрольных данных больше ошибок, чем на обучающей выборке. Поэтому предлагается формировать классификатор только из информативных закономерностей, т.е. информативность которых выше некоторого порога информативности (H_0), задаваемого исследователем. В результате это приведет к сокращению числа закономерностей в классификаторе без потери точности классификации или при незначительном ее изменении в положительную или отрицательную сторону.

При решении данной задачи возникает проблема выбора порога информативности. Для решения данной проблемы в работе разработана следующая итеративная процедура. На первом шаге порог информативности выбрать равным нулю для положительного и для отрицательного набора закономерностей, тем самым получается исходный классификатор, состоящий из максимального числа закономерностей. На следующем шаге процедуры предлагается выбрать порог информативности для отрицательных (положительных) закономерностей равный значению средней информативности ($H_{ср}$) по всем отрицательным (положительным) закономерностям:

$$H_{ср} = \frac{1}{\mathrm{q}} \cdot \sum_{i=1}^{q} H_i ,$$

где q – количество отрицательных (положительных) закономерностей в классификаторе, H_i – информативность отрицательной (положительной) i-й закономерности, рассчитанная по формуле (2.6).

Для получения нового классификатора, состоящего из более информативных закономерностей, удаляем из исходного классификатора все отрицательные (положительные) закономерности, значения информативности которых ниже найденного значения порога информативности для них. Рассчитав значения средней информативности для отрицательных и

положительных закономерностей текущего классификатора, будем их использовать для построения последующего классификатора, состоящего из закономерностей, информативность которых превышает значения средней информативности текущего классификатора. Таким образом, строим каждый последующий классификатор, используя значения средней информативности текущего. При этом количество закономерностей сокращается, а значения средней информативности возрастают для каждого последующего классификатора. Условием остановки следует считать момент увеличения количества неклассифицированных (непокрытых) наблюдений при классификации, т.е. закономерности, входящие в текущий классификатор, не покрывают некоторые наблюдения, входящие в экзаменующую выборку. Поэтому необходимо вернуться либо к предыдущему классификатору, поменяв значение двух порогов информативности на предыдущие их значения, либо поменять значение только одного порога информативности по отрицательным (положительным) закономерностям и посмотреть, каким образом это изменение отразиться на количестве неклассифицированных наблюдений и на результатах классификации в целом.

На основе разработанных алгоритмических процедур предлагаются модификации для метода логического анализа данных с целью усиления обобщающих способностей классификатора и повышения его интерпретируемости за счет сокращения числа правил, используемых в нем:

– применение целевой функции (2.5) и ограничения (2.4) для формирования закономерностей и построение классификатора только из тех правил, значение целевой функции для которых больше нуля;

– использование алгоритмической процедуры выбора базовых наблюдений для формирования закономерностей и применение к полученным правилам процедуры наращивания;

– применение алгоритмической процедуры построения классификатора как композиции информативных закономерностей на базе оптимизационной модели (2.2, 2.4) с процедурой наращивания.

Предлагаемые модификации для метода логического анализа данных позволяют повысить качество классификации новых наблюдений.

2.7 Решение задач псевдобулевой оптимизации

В процессе работы описанного метода необходимо многократное решение задач условной псевдобулевой оптимизации следующего вида:

$$\begin{cases} C(X) \to \max\limits_{X \in B_2^n} \\ A_j(X) \le H_j,\ j = \overline{1,m} \end{cases},$$

где $B_2 = \{0,1\}$, $B_2^n = B_2 \times B_2 \times \cdots \times B_2$; $C(X)$ и $A_j(X)$ – псевдобулевые функции, обладающие свойствами унимодальности и монотонности [69].

Как показано выше, функции эти в общем случае задаются алгоритмически, т.е. вычисляются через определенную последовательность операций. Поэтому наиболее приемлемыми для решения этой задачи являются так называемые поисковые алгоритмы оптимизации, которые не требуют задания функций в явном виде, с помощью алгебраических выражений, а используют вычисления функций в точках.

Для решения этой задачи разработан регулярный алгоритм псевдобулевой оптимизации [3], который находит точное оптимальное решение задачи за ограниченное время, причем показано, что этот алгоритм реализует информационную сложность данного класса задач и в этом смысле является неулучшаемым.

В то же время для многократного решения данной задачи с числом переменных 100 и выше наиболее целесообразным является использование приближенных алгоритмов [4, 47, 70], разработанных специально для этого класса задач и основанного на поведении монотонных функций $C(X)$ и $A_j(X)$ в

пространстве булевых переменных B_2^n. Эти алгоритмы с успехом применяются для решения прикладных задач в различных областях [17, 44]. Далее следует рассмотреть обоснование и описание некоторых приближенных алгоритмов условной псевдобулевой оптимизации.

Ниже в формализованном виде приведены некоторые понятия и определения, необходимые для описания работы алгоритмов [4, 5].

– Псевдобулевой функцией называют вещественную функцию на множестве булевых переменных: $f : B_2^n \to R^1$, где $B_2 = \{0,1\}$, $B_2^n = B_2 \times B_2 \times \cdots \times B_2$.

– Точки $X^1, X^2 \in B_2^n$ называются *k*-соседними, если они отличаются значением *k* координат, $k = \overline{1,n}$. 1–соседние точки называют просто соседними.

– Множество $O_k(X)$, $k = \overline{0,n}$, всех точек B_2^n, *k*-соседних к точке X, называют *k*-м уровнем точки X.

– Множество точек $W(X^0, X^l) = \{X^0, X^1, \ldots, X^l\} \subset B_2^n$ называют путем между точками X^0 и X^1, если $\forall\, i = 1, \ldots, l$ точка X^i является соседней к X^{i-1}.

– Множество $A \subset B_2^n$ называют связным множеством, если $\forall X^0, X^l \in A$ существует путь $W(X^0, X^l) \subset A$.

– Точку $X^* \in B_2^n$, для которой $f(X^*) < f(X), \forall X \in O_1(X^*)$, называют локальным минимумом псевдобулевой функции *f*.

– Псевдобулевую функцию, имеющую только один локальный минимум, называют унимодальной на B_2^n функцией.

– Унимодальную функцию f называют монотонной на B_2^n, если $\forall X^k \in O_k(X^*), k = \overline{1,n}:\ f(X^{k-1}) \le f(X^k), \forall X^{k-1} \in O_{k-1}(X^*) \cap O_1(X^k)$.

Таким образом, имеется задача следующего вида:

$$C(X) \to \max_{X \in S \subset B_2^n}, \qquad (2.7)$$

где $C(X)$ – монотонно возрастающая от X^0 псевдобулевая функция; $S \subset B_2^n$ – некоторая подобласть пространства булевых переменных, определяемая заданной системой ограничений (2.8):

$$A_j(X) \le H_j,\ j = \overline{1,m}. \qquad (2.8)$$

Далее вводится ряд понятий для подмножества точек пространства булевых переменных:

– Точка $Y \in S$ является граничной точкой множества S, если существует $X \in O_1(Y)$, для которой $X \notin S$.

– Точку $Y \in O_i(X^0) \cap S$ называют крайней точкой множества S с базовой точкой $X^0 \in S$, если $\forall X \in O_1(Y) \bigcap O_{i+1}(X^0)$ выполняется $X \notin S$.

– Ограничение, определяющее подобласть пространства булевых переменных, называют активным, если оптимальное решение задачи условной оптимизации не совпадает с оптимальным решением соответствующей задачи оптимизации без учета ограничения.

Следует рассмотреть некоторые свойства множества допустимых решений [68]:

– Если целевая функция является монотонной унимодальной функцией, а ограничение активно, то оптимальным решением задачи (2.7) будет точка, принадлежащая подмножеству крайних точек множества допустимых решений S с базовой точкой X^0, в которой целевая функция принимает наименьшее значение:

$$C(X^0) = \min_{X \in B_2^n} C(X).$$

– Рассматривается задача (2.7) с одним ограничением (2.8). Если функция ограничения (2.8) является унимодальной функцией, то множество допустимых решений S задачи (2.7) представляет собой связное множество.

С учетом указанных свойств множеств допустимых решений рассматриваются приближенные алгоритмы поиска граничных точек [46]. Для любой эвристики поиска граничных точек предлагается пара алгоритмов –

прямой и двойственный. Прямой алгоритм начинает поиск из допустимой области и движется по пути возрастания целевой функции, пока не найдет крайнюю точку допустимой области. Напротив, двойственный алгоритм ведет поиск в недопустимой области по пути убывания целевой функции, пока не найдет некоторого допустимого решения (некоторую граничную точку, которая может и не являться крайней).

Общая схема прямого алгоритма поиска:

1. Установить $X_1 = X^0$, $i = 1$.

2. В соответствии с правилом выбрать $X_{i+1} \in O_i(X^0) \cap O_1(X_i) \cap S$. Если таких точек нет, то на 3; иначе $i = i + 1$ и шаг повторяется.

3. $X_{opt} = X_{i+1}$.

Общая схема двойственного алгоритма поиска:

1. Установить $X_1 \in O_n(X^0)$, $i = 1$.

2. В соответствии с правилом выбрать $X_{i+1} \in O_{n-i}(X^0) \cap O_1(X_i)$. Если $X_{i+1} \in S$, то на 3; иначе $i = i + 1$ и повторяем шаг.

3. $X_{opt} = X_{i+1}$.

Существует несколько алгоритмов поиска граничных точек, которые отличаются друг от друга лишь правилом выбора следующей точки на шаге 2 общих схем. Далее приводятся два алгоритма, показавшие наиболее высокую эффективность при решении задач данного вида.

Правило 1. Гриди алгоритм

Точка X_{i+1} выбирается из условия:

$$\lambda(X_{i+1}) = \max_j \lambda(X^j),$$

где $X^j \in O_i(X^0) \cap O_1(X_i) \cap S$ для прямого алгоритма и $X^j \in O_{n-i}(X^0) \cap O_1(X_i)$ для двойственного.

Функция $\lambda(X)$ выбирается исходя из специфики задачи, например:

а) целевая функция $\lambda(X) = C(X)$,

б) удельная ценность $\lambda(X) = C(X)/A(X)$ (для одного ограничения) и т.д.

Правило 2. Модифицированный случайный поиск граничных точек

Точка X_{i+1} выбирается из условия

$$\lambda(X_{i+1}) = \max_r \lambda(X^r),$$

где X^r – точки, выбранные случайным образом с равной вероятностью из точек $O_i(X^0) \cap O_1(X_i) \cap S$ для прямого алгоритма (из точек $O_{n-i}(X^0) \cap O_1(X_i)$ для двойственного), $r = \overline{1, R}$; R – задаваемый параметр алгоритма.

Выводы

Данная глава посвящена описанию основных этапов метода логического анализа данных, созданию оптимизационных моделей для формирования закономерностей и разработке алгоритмических процедур, позволяющих улучшить интерпретируемость классификатора, сокращая количество правил в нем.

Формирование закономерностей и опорного множества представляет собой задачи условной псевдобулевой оптимизации, для решения которых используются алгоритмы оптимизации, основанные на поиске граничных точек допустимой области. Эти алгоритмы были разработаны специально для этого класса задач и основаны на поведении монотонных функций модели оптимизации в пространстве булевых переменных. Алгоритмы поиска граничных точек являются поисковыми, т.е. не требуют задания функций в явном виде, с помощью алгебраических выражений, а используют вычисления функций в точках.

Предложена алгоритмическая процедура получения усеченного набора признаков для формирования опорного множества, основанная на исключении неважных признаков в методе логического анализа данных, что позволяет сократить количество признаков, участвующих при формировании закономерностей, тем самым, снижая сложность задачи.

Разработана алгоритмическая процедура наращивания закономерностей, полученных на базе оптимизационной модели с максимальным покрытием наблюдений обучающей выборки, которая позволяет повысить информативность закономерностей, тем самым, способствуя увеличению точности принимаемых классификатором решений.

Создана модель оптимизации для формирования закономерностей с покрытием существенно различных подмножеств наблюдений обучающей выборки, которая позволяет повысить обобщающие способности классификатора, получаемого на базе данных правил.

Разработана алгоритмическая процедура выбора базовых наблюдений для формирования закономерностей с целью сокращения количества правил в классификаторе и снижения трудоемкости его построения при сохранении высокой точности.

Разработана алгоритмическая процедура построения классификатора как композиции информативных закономерностей с целью сокращения количества правил в классификаторе при сохранении высокой точности.

На основе разработанных алгоритмических процедур предложены модификации для метода логического анализа данных, позволяющие повысить интерпретируемость классификатора за счет сокращения числа правил в нем, сохраняя при этом высокую точность классификации при решении практических задач.

Следует отметить, что особенностью метода является то, что вместо того, чтобы просто ответить на вопрос, к какому из классов принадлежит новое наблюдение, он строит аппроксимацию областей пространства признаков,

содержащую наблюдения соответствующих классов. Наиболее важные преимущества такого подхода – это возможность дать объяснение для любого решения, полученного методом, возможность выявления новых классов наблюдений, возможность анализа роли и природы признаков.

3 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ И ЭКСПЕРИМЕНТАЛЬНЫЕ ИССЛЕДОВАНИЯ НА ПРАКТИЧЕСКИХ ЗАДАЧАХ

3.1 Программная реализация метода логического анализа данных и особенности использования программной системы

Метод логического анализа данных реализован в виде программной системы [30, 31, 49]. Данная программная система написана на языке программирования Delphi с использованием среды быстрого программирования приложений Delphi 2009 [7, 58, 59, 62]. Это позволило наиболее полно использовать возможности, предоставляемые объектно-ориентированным подходом в программировании, а также наиболее качественно разработать графический интерфейс пользователя, стандартный для приложений, работающих под управлением операционной системы Windows.

Структурная схема программной системы приведена на рисунке 3.1.

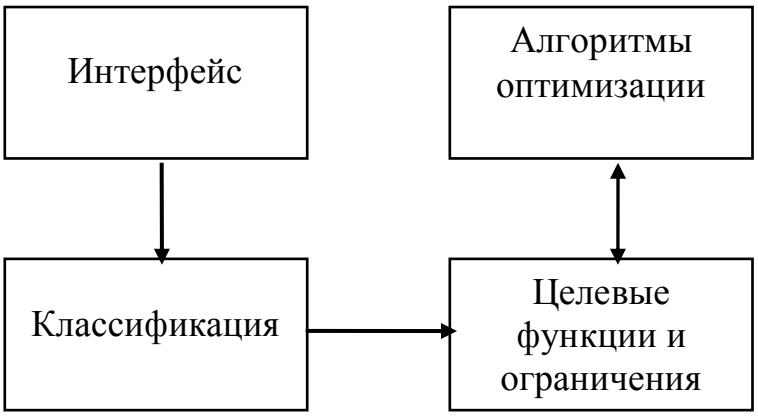

Рисунок 3.1 – Структурная схема программной системы

Разработанное программное обеспечение состоит из 3 модулей, реализующих метод логического анализа данных и модуля «Интерфейс», который позволяет осуществить взаимодействие пользователя со средой.

Краткое описание модулей:

– Модуль «Интерфейс» позволяет исследователю выбирать загружаемые данные, устанавливать самостоятельно или выбирать параметры необходимые

для работы метода, сохранять и просматривать результаты работы метода для конкретной задачи.

– Модуль «Классификация» включает в себя алгоритмы реализации всех этапов метода логического анализа данных: загрузка данных, выбор признаков, настройка теста, бинаризация признаков, нахождение опорного множества признаков, формирование закономерностей, построение классификатора, тестирование и сохранение полученных результатов.

– Модуль «Целевые функции и ограничения» содержит целевые функции и ограничения оптимизационных моделей для нахождения опорного множества и закономерностей.

– Модуль «Алгоритмы оптимизации» включает в себя алгоритмы оптимизации, служащие для нахождения оптимальных решений при поиске опорного множества и закономерностей.

Стартовое окно программной системы выглядит следующим образом (рис. 3.2):

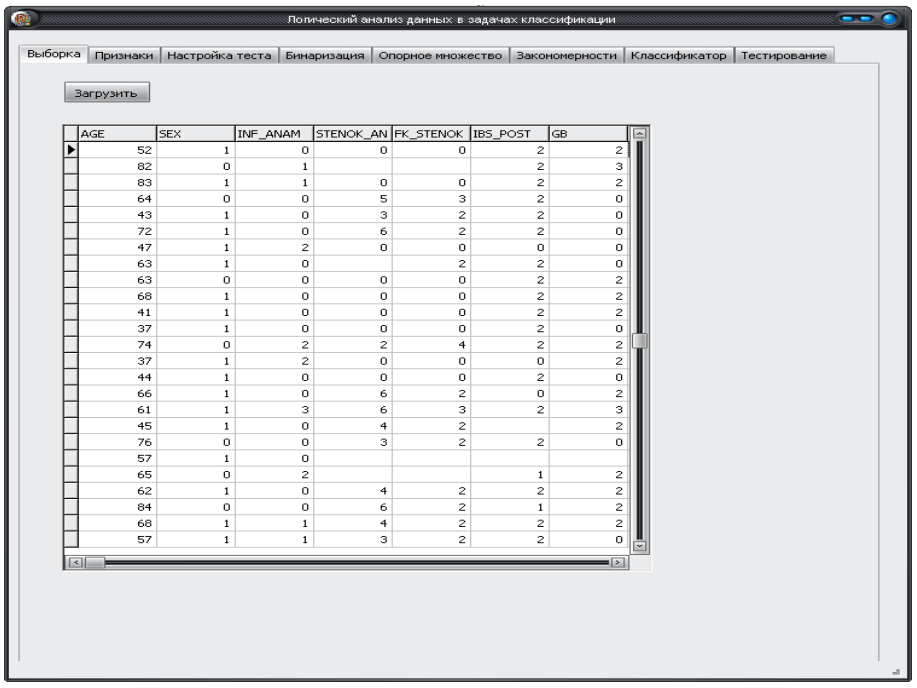

Рисунок 3.2 – Стартовое окно программной системы

В данной программной системе восемь вкладок, которые последовательно реализуют метод логического анализа данных. Название вкладок сообщает исследователю о той операции, которая выполняется на данной вкладке.

Для того чтобы реализовать процедуру классификации необходимо выполнить ряд шагов в программной системе, последовательно переходя от одной вкладки к другой [57]:

– загрузить данные, программа работает с таблицами формата *.dbf (вкладка «Выборка»);

– выбрать признаки, которые будут участвовать в процедуре классификации, и классификационный признак (вкладка «Признаки»);

– выбрать способ тестирования (вкладка «Настройка теста»);

– бинаризовать признаки – преобразовать количественные и номинальные переменные в бинарные переменные, так как метод логического анализа данных работает только с бинарными переменными (вкладка «Бинаризация»);

– построить опорное множество признаков, по которому возможно различить наблюдения двух классов (вкладка «Опорное множество»);

– построить логические закономерности для каждого класса (вкладка «Закономерности»);

– построить классификатор, на основе которого будет приниматься решение о принадлежности к конкретному классу (вкладка «Классификатор»);

– проверить полученный классификатор на тестовой выборке, т.е. получить и проанализировать результаты классификации (вкладка «Тестирование»).

Следует отметить, что в данной программной системе возможно три способа тестирования (вкладка «Настройка теста»):

– Первый способ. Использование обучающей выборки, т.е. все исходные данные используются как для обучения, так и для тестирования;

– Второй способ. Процентное разделение, т.е. исходная выборка разделяется на две части: обучающую и тестовую. Процент обучающей выборки задается в интервале от 1 до 100. Процент тестовой выборки вычисляется программой как разность между 100 и процентом обучающей выборки;

– Третий способ. Кросс-проверка, если множество исходных данных представляет относительно небольшую выборку наблюдений. Наиболее часто использующийся метод кросс-проверки – к-областной метод статистики. Этот метод заключается в случайном делении выборки на k приблизительно одинаковых подмножества, одно из этих подмножеств помечается как тестовое множество, модель строится на k-1 подмножествах, а затем тестируется на k-том. Этот процесс повторяется k раз, каждый раз выбирается новое тестовое множество, затем средняя точность выводится как мера качества используемого метода. Случай k-областей называется методом перочинного ножа или поочередного пропуска, если число k берется равным количеству наблюдений в выборке, т.е. тестовое множество состоит всегда только из 1 наблюдения. Точность классификации определяем простым отношением верно классифицированных наблюдений при тесте [32, 57].

В результате выполнения процедуры бинаризации исследователь видит общее число исходных переменных и полученное общее число бинарных переменных. Также при выделении конкретного признака в листе признаков, определяется его тип переменной, количество бинарных переменных для данного признака и конкретные значения порогов. Перед началом процедуры бинаризации исследователь может задавать самостоятельно число порогов для количественных переменных, регулируя общее число бинарных переменных.

Осуществляя поиск опорного множества, исследователь устанавливает минимальное число различий между наблюдениями двух классов, т.е. количество признаков по которым они должны отличаться, и ищет опорное множество для заданного числа различий. В результате в листе признаков

можно посмотреть все признаки, которые будут участвовать в классификации. Также указывается общая мощность множества признаков, т.е. количество переменных участвующих в классификации. Варьируя число различий между наблюдениями двух классов, получаются разные наборы признаков, участвующие в классификации.

При поиске закономерностей исследователем: выбирается максимальное число положительных и отрицательных закономерностей, которые необходимо построить; устанавливается количество термов в закономерности, по которым должны отличаться положительные от отрицательных наблюдений; устанавливается допуск, показывающий число наблюдений другого класса, которые захватывает закономерность; указывается максимальное количество правил, которыми может покрываться любое наблюдение обучающей выборки. В результате работы метода для каждого найденной закономерности указывается количество наблюдений своего и другого классов, которые она захватывает, степень данной закономерности.

При построении классификатора выбирается необходимое количество закономерностей, указываются пороги информативности для каждого класса. По желанию исследователя приводятся все закономерности, имеется возможность сохранить полученную модель в текстовый редактор (рис. 3.3).

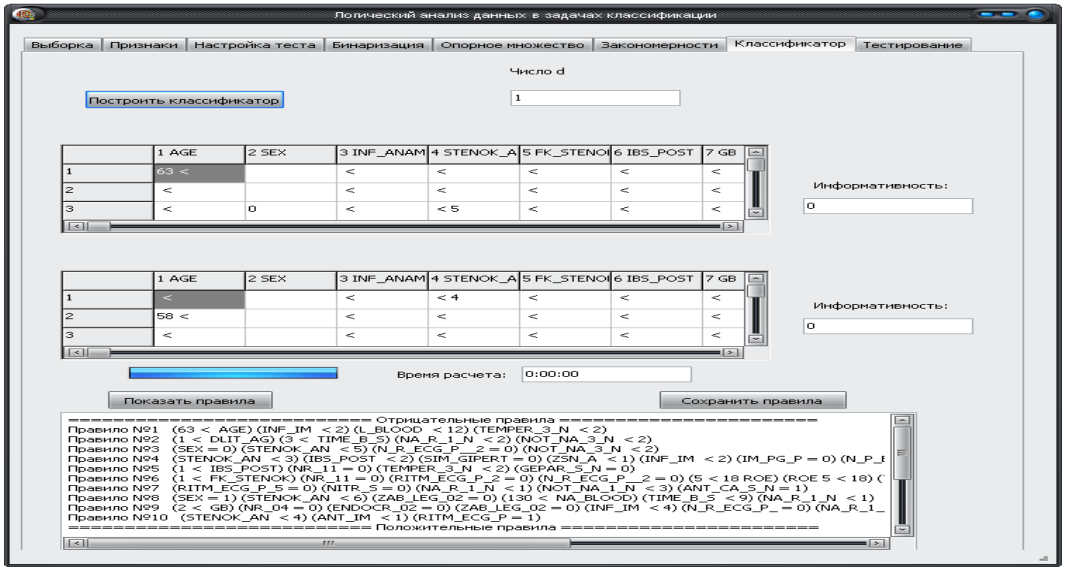

Рисунок 3.3 – Вкладка «Классификатор» программной системы

Помимо самой процедуры классификации, определяется важность каждого признака. Эта информация может быть использована исследователем для уменьшения количества переменных в задачи путем удаления признаков, не участвующих при построении закономерностей.

При выполнении тестирования определяются результаты классификации с учетом построенного классификатора. Сначала выбираются веса для классов положительных и отрицательных наблюдений. Сумма весов равна 1, поэтому достаточно задать вес для класса отрицательных наблюдений из интервала от 0 до 1, программная система автоматически вычислит вес для другого класса. Результаты классификации для каждого из классов указываются в полях «Точность». Также результаты классификации приводятся в итоговой таблице: первый столбец – номер наблюдения в экзаменующей выборке; второй столбец – количество закономерностей отрицательного класса, покрывающих наблюдение; третий столбец – число закономерностей положительного класса, покрывающих наблюдение; четвертый столбец – класс, который определил метод логического анализа данных; пятый столбец – реальный класс (рис. 3.4). Итоговую таблицу можно сохранить в целях анализа полученных результатов.

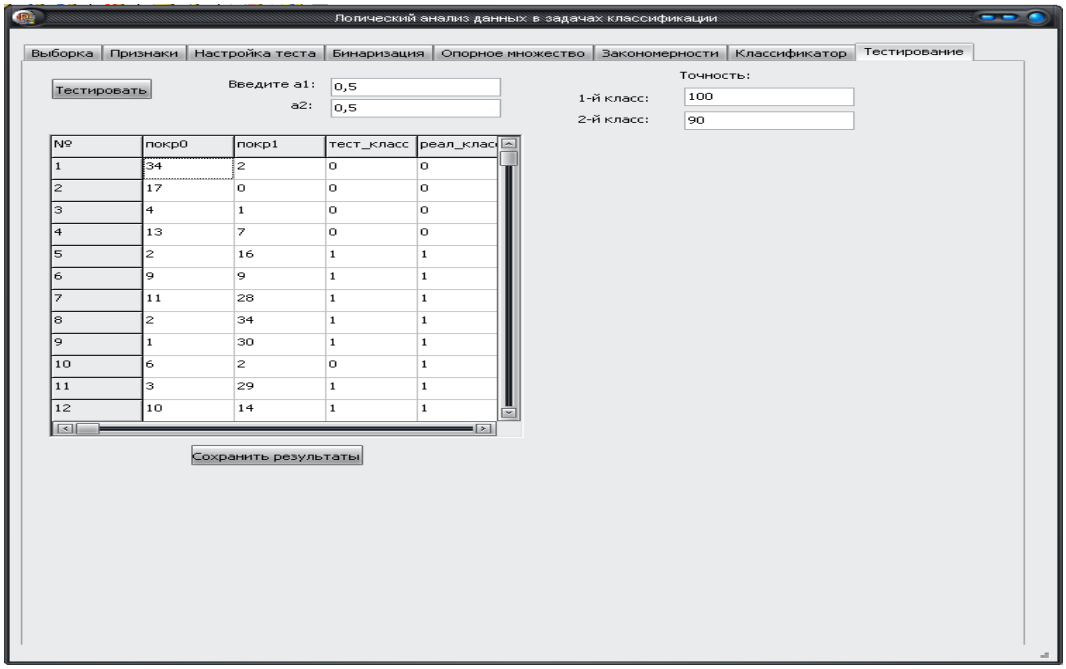

Рисунок 3.4 – Результаты классификации

Данная программная система решает задачи классификации с высокой точностью. Задача может иметь произвольную входную размерность, но количество выходов должно быть равно единице. Если в задаче больше выходов необходимо декомпозировать данную задачу на несколько задач.

3.2 Результаты экспериментальных исследований метода логического анализа данных и разработанных для него модификаций на практических задачах классификации

В диссертационном исследовании рассматриваются две задачи классификации, взятые из репозитория машинного обучения UCI, и задача прогнозирования осложнений инфаркта миокарда (фибрилляция желудочков, фибрилляция предсердий, отек легких, разрыв сердца и летальный исход).

Приводится описание задачи классификации результатов радарного сканирования ионосферы [113]. Система радаров, с помощью которой собраны данные, состоит из фазированной антенной решетки, которая включает 16 высокочастотных антенн с общей передаваемой мощностью порядка 6,4 кВт. Цели системы были свободные электроны в ионосфере. «Хороший» радарный возврат является доказательством некоторого типа структуры в ионосфере. «Плохой» – ее отсутствия, то есть сигналы проходят через ионосферу.

При проведении классификации результатов радарного сканирования ионосферы с целью определения существования некоторого типа структуры в ионосфере использовалась выборка данных, состоящая из 225 положительных наблюдений (структура существует) и 126 отрицательных наблюдений (структура отсутствует). Каждое наблюдение характеризовался вектором из 34 численных признаков. Выборка не имеет пропусков в данных. Двадцать процентов выборки используется для тестирования и в построении классификатора не участвует. В результате бинаризации из 34 признаков получено 136 бинарных признаков.

В таблице 3.1 представлены результаты испытаний с использованием четырех оптимизационных моделей. Проведено 20 экспериментов, результаты экспериментов усреднены.

Таблица 3.1 – Точность для задачи классификации результатов радарного сканирования ионосферы

Задача оптимизации	Мн–во правил	Кол–во правил	Покрытие отрицательных наблюдений	Покрытие положительных наблюдений	Степень правила	Точность классификации, %
Целевая функция (2.2), ограничение (2.3)	отр.	95	18	0	3	77
	пол.	186	0	68	5	95
Целевая функция (2.2), ограничение (2.4)	отр.	95	25	5	3	81
	пол.	186	5	102	4	95
Целевая функция (2.2), ограничение (2.4) с процедурой наращивания	отр.	95	25	4	4	84
	пол.	186	4	102	5	95
Целевая функция (2.5), ограничение (2.4)	отр.	54	20	5	5	81
	пол.	29	5	62	3	95

Следующей задачей классификации является выявление спама по электронной почте [113].

Понятие «спам» разнообразно: реклама продукции или веб-сайтов, схемы быстрого заработка денег, «письма счастья» и т.д. Необходимо отличать спам от важной корреспонденции, иметь признаки, которые будут полезны при построении спам-фильтра общего назначения для электронной почты.

При проведении классификации для данной задачи с целью определения спама и не спама использовалась выборка данных, состоящая из 181 положительных наблюдений (спам) и 279 отрицательных наблюдений (не спам). Каждое наблюдение характеризовался вектором из 57 численных признаков. Выборка не имеет пропусков в данных. Двадцать процентов выборки используется для тестирования и в построении классификатора не участвует. В результате бинаризации из 57 признаков получено 228 бинарных признаков.

В таблице 3.2 представлены результаты испытаний с использованием четырех оптимизационных моделей. Проведено 20 экспериментов, результаты экспериментов усреднены.

Таблица 3.2 – Точность классификации для задачи выявления спама

Задача оптимизации	Мн–во правил	Кол–во правил	Покрытие отрицательных наблюдений	Покрытие положительных наблюдений	Степень правила	Точность классификации, %
Целевая функция (2.2), ограничение (2.3)	отр.	234	49	0	4	98
	пол.	134	0	29	4	68
Целевая функция (2.2), ограничение (2.4)	отр.	234	96	5	5	98
	пол.	134	5	50	4	81
Целевая функция (2.2), ограничение (2.4) с процедурой наращивания	отр.	234	96	4	7	98
	пол.	134	4	50	5	87
Целевая функция (2.5), ограничение (2.4)	отр.	49	69	5	4	96
	пол.	59	5	31	4	72

Проведенные эксперименты показали высокую точность классификации для данных задач, взятых из репозитория машинного обучения UCI. Степень правил в данных задачах небольшая, поэтому построенные закономерности являются наглядными и легко интерпретируемыми для классификации наблюдений.

Третьей практической задачей, решаемой в рамках диссертационной работы, является задача прогнозирования осложнений инфаркта миокарда (ИМ). В настоящее время инфаркт миокарда является очень распространенным заболеванием. Стремительное распространение этого заболевания сделало его одной из наиболее острых проблем современной медицины. Заболеваемость ИМ отмечена во всех странах мира. Особенно подвержено ИМ городское население высокоразвитых стран, испытывающее быстрый ритм современной жизни и подвергающееся хроническому воздействию стрессовых факторов, имеющее не всегда сбалансированное питание [25].

Несмотря на то, что внедрение современных лечебно-профилактических мероприятий несколько снизило смертность от инфарктов, она продолжает

оставаться довольно высокой. Около 15-20% больных острым ИМ погибают на догоспитальном этапе, еще 15% в больнице [25], т.е. общая летальность при остром ИМ 30-35%. В США каждый день около 140 человек погибают от острого ИМ [25]. Настораживает высокая смертность в госпитальный период (т.е. во время нахождения больного в клинике), которая по данным различных российских авторов составляет от 10 до 20%. Госпитальная летальность больных острым ИМ в г. Красноярске держится на уровне 12-15% [26].

Течение заболевания у пациентов с ИМ протекает по-разному. ИМ может протекать без осложнений или с осложнениями, не ухудшающими долгосрочный прогноз. В то же время, около половины пациентов в острый и подострый периоды имеют осложнения, приводящие к ухудшению течения заболевания и даже летальному исходу. Предвидеть развитие этих осложнений не всегда может даже опытный специалист. В связи с этим, прогнозирование осложнений ИМ с целью своевременного проведения необходимых профилактических мероприятий представляется актуальной задачей.

Для решения этой задачи сотрудниками кафедры внутренних болезней № 1 Красноярской государственной медицинской академии была собрана информация о течении заболевания у 1700 больных ИМ, проходивших лечение в 1989-1995 годах в Кардиологическом центре городской больницы № 20 г. Красноярска. Информация получена из историй болезни пациентов и сконцентрирована в 124 полях электронной таблицы формата Paradox (Приложение А) [14, 51].

База данных содержит сведения, относящиеся к 11 группам медицинских параметров, которые определяют [14]:

1. состояние сердечно-сосудистой системы;
2. параметры эндокринной системы;
3. параметры системы органов дыхания;
4. артериальное давление по данным кардиологической бригады и приемного отделения;

5. осложнения, возникшие в момент транспортировки больного в клинику;

6. глубину и локализацию некроза сердечной мышцы;

7. основной водитель ритма, наличие (отсутствие) аритмий и нарушений проводимости на ЭКГ в момент поступления больного в реанимационное отделение;

8. вид лекарственного препарата примененного (непримененного) у пациента при проведении фибринолитической терапии;

9. электролитные сдвиги крови;

10. информацию о концентрации в крови некоторых ферментов, лейкоцитов, времени госпитализации от момента возникновения инфаркта миокарда;

11. течение заболевания в первые дни ИМ.

Среди признаков встречаются как бинарные (большая часть) и номинальные, так и численные признаки. В выборке присутствует значительное число пропущенных данных.

Среди выбранных осложнений два серьезных нарушения сердечного ритма (фибрилляция предсердий, фибрилляция желудочков), отек легких, разрыв сердца, а так же летальный исход. Фибрилляция предсердий встречается у 10-15% больных ИМ [56, 63] нередко вызывая расстройство кровообращения и способствуя возникновению внутрипредсердных тромбов с последующими эмболиями [60]. Поэтому очень важно предвидеть ее возникновение с целью своевременного проведения необходимых профилактических мероприятий.

Ранее задача прогнозирования осложнений ИМ была решена с помощью нейронных сетей [14]. При ее решении отмечено, что классификатор дает невысокие результаты в случае существенного различия в количестве наблюдений каждого класса в исходной выборке, поэтому был предложен следующий подход к решению данной задачи. Число пациентов с некоторым осложнением (положительные наблюдения) примерно в десять раз меньше

числа пациентов, у которых это осложнение не наблюдалось (отрицательные наблюдения). Исходная выборка (1700 наблюдений) разделяется на тестовую выборку и 10 обучающих выборок для каждой осложнения, причем положительные наблюдения в обучающих выборках остаются одни и те же, а отрицательные наблюдения изменяются. Метод обучается на каждой обучающей выборке отдельно, но тестируется на общей экзаменующей выборке. Окончательно решение по каждому наблюдению экзаменующей выборки принимается путем большинства голосов отданных всеми классификаторами, полученными на базе 10 обучающих выборок. При использовании данного подхода для решения задачи, помимо повышения качества классификации, появляется возможность сравнения результатов классификации методов логического анализа данных и нейронных сетей.

Для нахождения правил использовались четыре оптимизационные модели: «жесткая» модель, не допускающая, чтобы построенные правила покрывали наблюдения другого класса; модифицированная модель, позволяющая, чтобы правила покрывали некоторое ограниченное число наблюдений другого класса; модифицированная модель с процедурой наращивания закономерностей; модель для формирования закономерностей с покрытием существенно различных подмножеств наблюдений обучающейся выборки.

Следует сделать важное замечание по точности полученных результатов. Точность классификации определяется двумя значениями: чувствительностью – точностью определения пациентов с осложнением, и специфичностью – точностью определения пациентов без осложнений. На практике к чувствительности предъявляются большие требования, чем к специфичности.

Проводится апробация метода логического анализа данных для прогноза пяти осложнений ИМ – фибрилляции желудочков (ФЖ), фибрилляции предсердий (ФП), отека легких (ОЛ), разрыва сердца (РС) и летального исхода (ЛИ). Количество пациентов с осложнениями и без осложнений каждой из 10

выборок, а также объем тестовой выборки для всех рассматриваемых осложнений ИМ представлены в таблице 3.3.

Таблица 3.3 – Состав каждой выборки для всех осложнений ИМ

	ФЖ	ФП	ОЛ	РС	ЛИ
Кол-во пол. наблюдений	70	170	159	54	160
Кол-во отр. наблюдений	181	180	173	179	172
Кол-во наблюдений экзаменующей выборки	30	50	39	28	50

Задача 1. Фибрилляция желудочков

В задаче для тестирования используются 30 наблюдений. Бинарных признаков получено 206 из 112 исходных. В таблице 3.4 представлены результаты испытаний с использованием четырех оптимизационных моделей. Приведены средние значения покрытий и степени для закономерностей.

Таблица 3.4 – Результаты классификации для задачи прогнозирования ФЖ

Задача оптимизации	Мн–во правил	Кол–во правил	Покрытие отрицательных наблюдений	Покрытие положительных наблюдений	Степень правила	Точность классификации, %
Целевая функция (2.2), ограничение (2.3)	отр.	160	14	0	5	90
	пол.	60	0	14	3	78
Целевая функция (2.2), ограничение (2.4)	отр.	160	26	5	5	90
	пол.	60	5	22	3	78
Целевая функция (2.2), ограничение (2.4) с процедурой наращивания	отр.	160	26	3	6	90
	пол.	60	3	22	4	83
Целевая функция (2.5), ограничение (2.4)	отр.	54	16	5	6	80
	пол.	51	5	18	5	83

Примеры полученных правил приведены на рисунке 3.5.

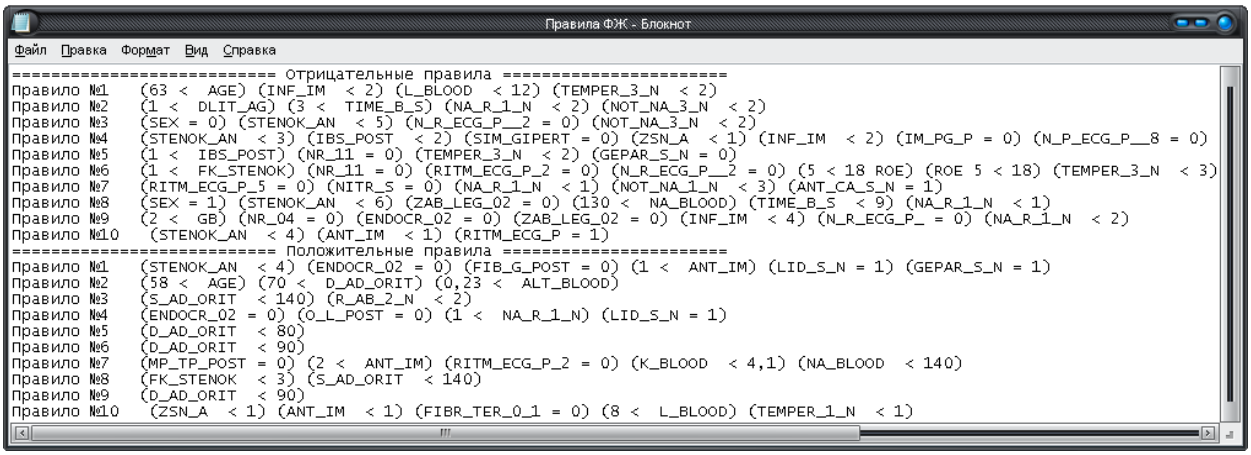

Рисунок 3.5 – Примеры правил для задачи прогнозирования ФЖ

Задача 2. Фибрилляция предсердий

Для проведения испытаний используются 50 наблюдений и в построении классификаторов не участвуют. В результате бинаризации из 112 признаков получено 214 бинарных. В таблице 3.5 представлены результаты, аналогичные результатам таблицы 3.4.

Таблица 3.5 – Результаты классификации для задачи прогнозирования ФП

Задача оптимизации	Мн–во правил	Кол–во правил	Покрытие отрицательных наблюдений	Покрытие положительных наблюдений	Степень правила	Точность классификации, %
Целевая функция (2.2), ограничение (2.3)	отр.	150	13	0	6	73
	пол.	150	0	13	5	58
Целевая функция (2.2), ограничение (2.4)	отр.	150	27	5	7	80
	пол.	150	5	27	6	68
Целевая функция (2.2), ограничение (2.4) с процедурой наращивания	отр.	150	27	4	9	80
	пол.	150	4	27	6	68
Целевая функция (2.5), ограничение (2.4)	отр.	82	16	5	6	67
	пол.	87	5	16	5	58

Задача 3. Отек легких

Для тестирования используется 39 наблюдений. Бинарных признаков получено 212 из 112 исходных. В таблице 3.6 – результаты для четырех моделей.

Таблица 3.6 – Результаты классификации для задачи прогнозирования ОЛ

Задача оптимизации	Мн–во правил	Кол–во правил	Покрытие отрицательных наблюдений	Покрытие положительных наблюдений	Степень правила	Точность классификации, %
Целевая функция (2.2), ограничение (2.3)	отр.	152	18	0	7	62
	пол.	141	0	14	4	82
Целевая функция (2.2), ограничение (2.4)	отр.	152	36	5	7	68
	пол.	141	5	29	4	89
Целевая функция (2.2), ограничение (2.4) с процедурой наращивания	отр.	152	36	4	9	68
	пол.	141	4	29	5	89
Целевая функция (2.5), ограничение (2.4)	отр.	67	16	5	6	62
	пол.	73	5	16	4	93

Задача 4. Разрыв сердца

Для проведения испытаний в данном случае используется 28 наблюдений. Бинарных признаков получено 202 из 112 исходных. В таблице 3.7 – результаты для четырех моделей.

Таблица 3.7 – Результаты классификации для задачи прогнозирования РС

Задача оптимизации	Мн–во правил	Кол–во правил	Покрытие отрицательных наблюдений	Покрытие положительных наблюдений	Степень правила	Точность классификации, %
Целевая функция (2.2), ограничение (2.3)	отр.	163	15	0	4	100
	пол.	42	0	11	3	79
Целевая функция (2.2), ограничение (2.4)	отр.	163	29	5	5	100
	пол.	42	5	21	3	79
Целевая функция (2.2), ограничение (2.4) с процедурой наращивания	отр.	163	29	4	6	100
	пол.	42	4	21	4	93
Целевая функция (2.5), ограничение (2.4)	отр.	25	20	5	3	100
	пол.	41	5	15	2	86

Задача 5. Летальный исход

В данной задаче для тестирования используются 50 наблюдений. Бинарных признаков получено 214 из 112 исходных. В таблице 3.8 – результаты для четырех моделей.

Таблица 3.8 – Результаты классификации для задачи прогнозирования ЛИ

Задача оптимизации	Мн–во правил	Кол–во правил	Покрытие отрицательных наблюдений	Покрытие положительных наблюдений	Степень правила	Точность классификации, %
Целевая функция (2.2), ограничение (2.3)	отр.	152	28	0	7	80
	пол.	130	0	11	4	85
Целевая функция (2.2), ограничение (2.4)	отр.	152	53	5	7	87
	пол.	130	5	22	4	85
Целевая функция (2.2), ограничение (2.4) с процедурой наращивания	отр.	152	53	4	8	87
	пол.	130	4	22	4	85
Целевая функция (2.5), ограничение (2.4)	отр.	44	37	5	6	87
	пол.	54	5	17	3	88

Как видно из таблиц 3.4–3.8, использование в оптимизационной модели ограничения (2.4) при поиске правил позволяет находить закономерности с более высоким покрытием. Применение такого подхода необходимо при решении задач с наличием выбросов, шумов и с пропусками в выборке данных.

В результате применения процедуры наращивания закономерностей, построенных с помощью модифицированной оптимизационной модели, получаются закономерности с максимальным покрытием и с более высокой степенью, повышая надежность принимаемых решений классификатором, построенном на базе данных правил. Повышение надежности решений основано на увеличении информативности закономерностей, связанным с тем, что количество захватываемых наблюдений своего класса (значение целевой функции для закономерности) остается неизменным, а количество захватываемых наблюдений другого класса уменьшается.

Модификация для метода логического анализа данных, связанная с применением целевой функции (2.5) в оптимизационной модели, позволяет упростить классификатор, сокращая количество закономерностей в нем относительно их полного набора для конкретной задачи, так как в классификатор попадают только правила со значением целевой функции больше нуля.

Сравнение методов логического анализа данных (LAD) и нейронных сетей [14] по точности классификации приведено в таблице 3.9.

Таблица 3.9 – Сравнение результатов классификации

Прогнозируемое осложнение	Класс	Точность классификации, %	
		Нейронные сети	LAD
ФП	пол.	85	68
	отр.	67	80
ФЖ	пол.	76	83
	отр.	70	90
РС	пол.	70	93
	отр.	60	100
ОЛ	пол.	75	89
	отр.	70	68
ЛИ	пол.	86	85
	отр.	80	87

Согласно таблице 3.9 точность решения, полученная методом логического анализа данных, сравнима с точностью решения с помощью метода нейронных сетей. Преимуществом метода логического анализа данных является формирование в явном виде правил, по которым принимается решение о принадлежности к какому-либо классу.

Необходимо проведение дальнейших исследований, которые помогут выявить влияние на точность прогноза исключения признаков, не встречающихся при формировании закономерностей [29].

Набор признаков, оставшихся после удаления, называется усеченным набором. Процедура получения усеченного набора изложена в пункте 2.3 диссертационной работы.

Исследуется изменение точности классификации метода логического анализа данных при прогнозе осложнений ИМ на различных усеченных наборах признаков.

Проведены исследования на трех усеченных наборах и сделано сравнение с результатами, полученными на полном наборе. Усеченный набор №1 получен путем исключения всех признаков, важность которых равна 0%, усеченный набор №2 – ниже 1%, усеченный набор №3 – ниже 2%.

Для решения задачи классификации проведены 20 экспериментов для каждого набора признаков. В каждом эксперименте объем выборки используемый для тестирования определялся согласно таблице 3.3.

В таблицах 3.10 – 3.14 представлена точность классификации для каждой задачи, полученная до исключения (полный набор) и после исключения (усеченный набор) минимально важных признаков.

Таблица 3.10 – Точность прогнозирования осложнения ФЖ

Модель	Количество признаков в наборе	Точность прогноза, %	
		Положительные наблюдения	Отрицательные наблюдения
Полный набор	112	82,3	83,6
Усеченный набор №1	96	78,6	88
Усеченный набор №2	80	92,7	83
Усеченный набор №3	55	73	86

В результате приведенных исследований самая высокая точность для задачи ФЖ получена на усеченном наборе №2.

Таблица 3.11 – Точность прогнозирования осложнения ФП

Модель	Количество признаков в наборе	Точность прогноза, %	
		Положительные наблюдения	Отрицательные наблюдения
Полный набор	112	77	84
Усеченный набор №1	105	72,6	76,8
Усеченный набор №2	85	87,2	69,8
Усеченный набор №3	70	65	83,8

Для задачи ФП усеченный набор №2 обеспечил наибольшую точность.

Таблица 3.12 – Точность прогнозирования осложнения ОЛ

Модель	Количество признаков в наборе	Точность прогноза, %	
		Положительные наблюдения	Отрицательные наблюдения
Полный набор	112	79,8	83
Усеченный набор №1	108	86,6	75,6
Усеченный набор №2	78	87,6	83
Усеченный набор №3	62	76,2	87,4

Для задачи ОЛ усеченный набор №2 обеспечил наибольшую точность.

Таблица 3.13 – Точность прогнозирования осложнения РС

Модель	Количество признаков в наборе	Точность прогноза, %	
		Положительные наблюдения	Отрицательные наблюдения
Полный набор	112	64,8	88,8
Усеченный набор №1	84	90	96,6
Усеченный набор №2	61	96,6	90
Усеченный набор №3	42	80	96,6

Для задачи РС усеченный набор №2 обеспечил наибольшую точность.

Таблица 3.14 – Точность прогнозирования осложнения ЛИ

Модель	Количество признаков в наборе	Точность прогноза, %	
		Положительные наблюдения	Отрицательные наблюдения
Полный набор	112	89,6	63
Усеченный набор №1	106	90,2	83,2
Усеченный набор №2	76	84	77,2
Усеченный набор №3	59	91,2	63

Для данной задачи усеченный набор №1 обеспечил наибольшую точность с учетом замечания, так как точность классификации положительных наблюдений меньше на 1%, чем при усеченном наборе №3, но правильно классифицированных отрицательных наблюдений на 20,2% больше.

Приведены для каждой задачи в Приложении Б признаки с нулевой важностью относительно каждой группы медицинских параметров.

В задаче ФЖ (таблица Б.1) признаки с нулевой важностью были выявлены в 5 группах.

В задаче ФП (таблица Б.2) 100% признаков с нулевой важностью относятся к группе, определяющей состояние сердечнососудистой системы.

В задаче ОЛ (таблица Б.3) большая часть признаков с нулевой важностью относится к группе, определяющей состояние сердечнососудистой системы.

В задаче РС (таблица Б.4) значительная часть признаков с нулевой важностью относится к группе, определяющей состояние сердечнососудистой системы, а так же группе, определяющей основной водитель ритма, наличие (отсутствие) аритмий и нарушений проводимости на ЭКГ в момент поступления больного в реанимационное отделение.

В задаче ЛИ (таблица Б.5) большая часть признаков с нулевой важностью относится к группе, определяющей состояние сердечнососудистой системы.

В приложении Б также приведены признаки с максимальной важностью в порядке убывания для каждого из пяти осложнений без указания группы медицинских параметров (таблицы Б.6 – Б.10).

Необходимо выполнить проверку эффективности модификаций метода логического анализа данных, связанных с алгоритмическими процедурами сокращения количества закономерностей в классификаторе. Апробация данных процедур проводится на двух задачах: классификация результатов радарного сканирования ионосферы и выявления спама, взятых из репозитория машинного обучения UCI.

Проверяется процедура выбора базовых наблюдений для формирования закономерностей [36, 37]. Для задачи классификации результатов радарного сканирования ионосферы генерируется по 15 центроидов для каждого класса, используя алгоритм «k-средних» в программе WEKA. Добавляются в исходную обучающую выборку сгенерированные центроиды, строятся на их базе закономерности. В данной задаче для тестирования используется 20% выборки, состоящей из 240 положительных и 141 отрицательных наблюдений. Аналогично при решении задачи выявления спама генерируются 20 центроидов для положительного класса и 25 центроидов для отрицательного класса, используя алгоритм «k-средних» в программе WEKA [115]. В данной задаче для тестирования используется 20% выборки, состоящей из 201 положительных и 304 отрицательных наблюдений. В процессе проведения экспериментов подобрано количество наблюдений другого класса, которое может захватывать закономерность. Результаты классификации приведены в таблице 3.15-3.16.

Таблица 3.15 – Точность для задачи классификации результатов радарного сканирования ионосферы

Множество правил	Покрытие отр. наблюдений в новом / исходном классификаторах	Покрытие пол. наблюдений в новом / исходном классификаторах	Степень правила в новом / исходном классификаторах	Кол-во правил в новом / исходном классификаторах	Точность нового классификатора, %	Точность исходного классификатора, %
отр.	45 / 36	15 / 15	2 / 2	15 / 95	74	68
пол.	15 / 15	139 / 130	3 / 3	15 / 186	96	98

Таблица 3.16 – Точность классификации для задачи выявления спама

Множество правил	Покрытие отр. наблюдений в новом / исходном классификаторах	Покрытие пол. наблюдений в новом / исходном классификаторах	Степень правила в новом / исходном классификаторах	Кол-во правил в новом / исходном классификаторах	Точность нового классификатора, %	Точность исходного классификатора, %
отр.	149 / 119	10 / 10	4 / 4	25 / 234	94	96
пол.	10 / 10	74 / 60	3 / 3	20 / 134	86	89

Согласно таблицам 3.15–3.16, получили небольшое изменение точности классификации для решаемых задач и сокращение количества правил классификатора для задачи классификации результатов радарного сканирования ионосферы в 9 раз, для задачи выявления спама в 8 раз. Таким образом, модификация метода, связанная с алгоритмической процедурой выбора базовых наблюдений для формирования закономерностей является эффективной с точки зрения применимости для построения правил, образующих новый классификатор.

Проверим алгоритмическую процедуру построения классификатора как композиции информативных закономерностей [34]. В задаче классификации результатов радарного сканирования ионосферы выборка состоит из 225 положительных и 126 отрицательных наблюдений, а в задаче выявления спама – из 181 положительных и 279 отрицательных наблюдений. Для теста используется 20% выборки. Результаты классификации приведены в таблицах 3.17–3.18.

Таблица 3.17 – Точность для задачи классификации результатов радарного сканирования ионосферы при изменении порога информативности, H_0

Номер опыта	Множество правил	Количество правил	Средняя информативность, Hcp	Порог информативности, H_0	Покрытие отр. наблюдений	Покрытие пол. наблюдений	Количество непокрытых наблюдений	Точность классификации, %
1	отр.	95	2,95	0	25	5	0	71
	пол.	186	7,87	0	5	103		97
2	отр.	55	3,76	2,95	33	5	0	81
	пол.	121	8,45	7,87	5	114		97
3	отр.	24	4,22	3,76	36	5	0	74
	пол.	53	8,79	8,45	5	121		97
4	отр.	13	4,52	4,22	37	5	3	65
	пол.	23	9,02	8,79	5	126		95
5	отр.	24	4,22	3,76	36	5	0	71
	пол.	23	9,02	8,79	5	126		97
6	отр.	24	4,22	3,76	36	5	0	87
	пол.	10	9,22	9,02	5	131		95

Таблица 3.18 – Результаты классификации для задачи выявления спама при изменении порога информативности, H_0

Номер опыта	Множество правил	Количество правил	Средняя информативность, *Hcp*	Порог информативности, H_0	Покрытие отр. наблюдений	Покрытие пол. наблюдений	Количество непокрытых наблюдений	Точность классификации, %
1	отр.	234	7,84	0	120	10	0	96
	пол.	134	4,49	0	10	57		89
2	отр.	132	8,51	7,84	134	10	0	93
	пол.	79	5,49	4,49	10	70		85
3	отр.	68	8,85	8,51	141	10	1	87
	пол.	39	6,05	5,49	10	77		79
4	отр.	68	8,85	8,51	141	10	0	98
	пол.	79	5,49	4,49	10	70		87
5	отр.	34	9,03	8,85	146	10	0	96
	пол.	79	5,49	4,49	10	70		89
6	отр.	18	9,12	9,03	148	10	0	96
	пол.	79	5,49	4,49	10	70		89

Согласно полученным результатам (таблицы 3.17–3.18) можно отметить, что модификация метода, связанная с данной алгоритмической процедурой, позволяет упростить классификатор, поскольку количество правил, которые его составляют, сокращается в 2-8 раз относительно полного набора правил для конкретной задачи. При этом точность классификации либо не уменьшается, либо уменьшается незначительно. В некоторых случаях при удалении менее информативных закономерностей из классификатора точность классификации возрастает. Это объясняется тем, что эти правила, являясь статистически ненадежными, покрывали наблюдения экзаменующей выборки, т.е. участвовали в решающем правиле наряду с информативными закономерностями, тем самым, повышая ошибку классификации.

3.3 Настройка параметров метода логического анализа данных с учетом специфики решаемых задач

Метод логического анализа данных является достаточно гибким инструментом анализа данных, позволяющим учитывать специфику конкретной задачи классификации и требования заказчика (исследователя) при ее решении. На этапах построения опорного множества, формирования закономерностей, построения классификатора имеются параметры метода, которые путем целенаправленной их настройки позволяют соблюдать баланс между различными критериями сравнения алгоритмов классификации. Ниже приводится описание параметров метода для каждого из перечисленных этапов, особенности их настройки.

На этапе построения опорного множества исследователь устанавливает минимальное число различий между наблюдениями двух классов, т.е. количество признаков по которым они должны отличаться. Варьируя данный параметр, получаются разные наборы признаков, которые используются в дальнейшем при построении закономерностей. Изменения данного параметра влияют на точность классификации и трудоемкость построения правил. Чем меньше набор признаков, используемых для разделения, тем ниже трудоемкость построения правил, так как сокращается пространство поиска. Но при значительном сокращении пространства поиска не удается построить правила и композицию из этих правил, корректно классифицирующую наблюдения экзаменующей выборки. Для получения корректного набора признаков, применяющихся при построении закономерностей, в диссертационной работе предложена алгоритмическая процедура получения усеченного набора признаков для формирования опорного множества, основанная на исключении неважных признаков. Эффективность этой процедуры эмпирически доказана на практических задачах.

На этапе формирования закономерностей применение оптимизационной модели, разрешающей покрытие правилом малого числа наблюдений другого класса, позволяет находить закономерности с более высоким покрытием, из которых строится более точный классификатор. Такой подход эффективен при решении задач с наличием выбросов и шумов и с большим количеством пропусков в выборке данных.

При использовании данной оптимизационной модели исследователем устанавливается количество наблюдений другого класса, которые может захватить каждое правило. С помощью регулирования данного параметра устанавливается компромисс между распознающей и обобщающей способностью классификатора. При низком значении параметра происходит эффект переобучения, поскольку процент правильно классифицируемых наблюдений из обучающей выборки превосходит процент правильно классифицируемых наблюдений экзаменующей выборки. Увеличивая значение параметра, достигаем баланса между распознающей и обобщающей способностью классификатора.

Также на этапе поиска закономерностей при использовании оптимизационной модели для формирования правил, выделяющих существенно различные подмножества наблюдений выборки, возникает параметр $N_{\max}$, который означает максимальное количество закономерностей, покрывающих наблюдение обучающей выборки в классификаторе. Параметр $N_{\max}$ для каждого класса задается в диапазоне от 1 до максимального количества построенных закономерностей для данного класса. Если $N_{\max}$ принимает значение близкое или равное максимальному количеству закономерностей для данного класса, то новый классификатор работает аналогично построенному на базе оптимизационной модели с максимальным покрытием. Если $N_{\max}$ стремиться к 1, сокращается количество правил со значением целевой функции (2.5) больше 0, которые составляют классификатор, т.к. изначально захватываются все наблюдения и их веса обнуляются. В новом классификаторе присутствует

недостаточное количество закономерностей, которые, в итоге, не способны классифицировать вновь поступающие наблюдения, т.е. качество классификации снижается. При этом возникает высокий процент отказов от классификации. Эмпирическим путем проверено, что значение параметра необходимо выбирать в диапазоне от 5 и до значения среднего покрытия закономерностей, построенных с использованием оптимизационной модели с максимальным покрытием, причем чем ниже значение $N_{\max}$, тем меньше количество правил в классификаторе, то есть возрастает его интерпретируемость.

Ещё одним параметром на этапе поиска правил является количество центроидов для каждого класса в алгоритмической процедуре выбора базовых наблюдений для формирования закономерностей. Параметр позволяет определить количество правил в классификаторе, трудоемкость его построения. При недостаточном количестве правил в классификаторе точность классификации снижается из-за роста отказов от классификации. Поэтому, варьируя количество центроидов для каждого класса необходимо следить за изменением числа неклассифицированных наблюдений экзаменующей выборки. Использование алгоритмической процедуры выбора базовых наблюдений, может быть, оправдано требованием получения интерпретируемого классификатора и необходимостью увеличения скорости обучения.

На этапе построения классификатора при реализации алгоритмической процедуры построения классификатора как композиции информативных закономерностей порог информативности выступает параметром, регулирующим число закономерностей в классификаторе. При его постепенном увеличении интерпретируемость классификатора возрастает, так как уменьшается число правил в нем, но, начиная с определенного значения параметра, происходит рост отказов от классификации, следовательно, снижение точности классификации в целом. Рост отказов происходит из-за

удаления всех правил, которые ранее покрывали определенные наблюдения экзаменующей выборки, то есть появления непокрытых наблюдений при тесте. Поэтому необходимо устанавливать корректное значение порога информативности с целью соблюдения баланса между интерпретируемостью классификатора и точностью классификации.

Таким образом, в зависимости от требований заказчика и специфики задачи, выполняется настройка параметров метода. Имеется набор параметров метода, корректная настройка которых позволяет найти компромисс между различными критериями сравнения алгоритмов классификации.

3.4 Сравнительный анализ метода логического анализа данных с другими алгоритмами классификации на практических задачах

Сравнительный анализ метода логического анализа данных проводится со следующими алгоритмами классификации: алгоритм построения 1-правил, RIPPER, CART, C4.5, Random Forest, Adaboost. Приводится краткое описание данных алгоритмов классификации, указываются их преимущества и недостатки, особенности применения.

Алгоритм построения 1-правил [9]

Имеются независимые переменные $A^1 \ldots A^j \ldots A^k$, принимающие значения $< x_1^1 \ldots x_n^1 >, \ldots, < x_1^j \ldots x_n^j >, \ldots, < x_1^k \ldots x_n^k >$ соответственно, и зависимая переменная C, принимающая значения $c_1 \ldots c_r$. Для любого возможного значения каждой независимой переменной формируется правило, которое классифицирует наблюдение из обучающей выборки. В если-части правила указывают значение независимой переменной (Если $A^j = x_i^j$). В то-части правила указывается наиболее часто встречающееся значение зависимой переменной у данного значения независимой переменной (то $C = c_r$). Ошибкой правила является количество наблюдений, имеющих данное значение рассматриваемой независимой переменной ($A^j = x_i^j$), но не имеющих наиболее часто

встречающееся значение зависимой переменной у данного значения независимой переменной ($C \neq c_r$). Оценив ошибки, выбирается переменная, для которой ошибка набора минимальна. Недостатком алгоритма является сверхчувствительность, заключающаяся в том, что алгоритм выбирает переменную, стремящуюся к ключу. Ключ – переменная с максимальным количеством значений, у ключа ошибка вообще 0, но он не несет информации. Отличительной особенностью алгоритма является то, что классификация строится по одному атрибуту, который в явном виде известен.

Алгоритм RIPPER

Предложенный Уильямом Коэном алгоритм состоит из двух этапов: построения правил и их оптимизации [81].

Этап построения включает две процедуры: наращивание и редукция. Процедура наращивания правил, начиная с пустого правила, жадно добавляет термы, пытаясь сделать правило совершенным, чтобы оно покрывало наблюдения только своего класса. Процедура наращивания пытается перебрать все возможные значения каждого атрибута и выбрать терм с самым высоким информационным выигрышем. При редукции удаляются из дальнейшего рассмотрения все положительные и отрицательные наблюдения, покрываемые правилом. Построенное правило добавляется в набор, вычисляется длина описания набора правил. Алгоритм переходит к построению следующего правила. Правила продолжают добавлять в набор до тех пор, пока длина описания получаемого набора правил меньше или равна наименьшей длине описания набора правил, полученной до сих пор, или до момента, когда доля ошибок для генерируемых правил становится больше либо равна 50%.

На этапе оптимизации, после создания начального набора правил, для каждого правила в наборе рассматривают два его альтернативных варианта. Первый вариант генерируется из пустого правила, второй генерируется жадно добавлением термов в исходное построенное правило. Сравнивается длина описания трех наборов правил, содержащих исходное правило и два его

альтернативных варианта. Набор правил с минимальной длиной описания выбирается в качестве итогового.

Если в задаче классификации имеются два класса, то правила строятся только для одного из них. При классификации нового наблюдения, если оно не покрывается построенными правилами, то оно относится к другому классу.

Алгоритм CART

Алгоритм CART (Classification and Regression Tree) создан для решения задач классификации и регрессии построением дерева решений. Он разработан в 1974–1984 годах четырьмя профессорами статистики: Лео Брейманом (Беркли), Джеромом Фридманом (Jerome H. Friedman, Стэнфорд), Чарлзом Стоуном (Charles Stone, Беркли) и Ричардом Олшеном (Richard A. Olshen, Стэнфорд) [75].

Каждый узел бинарного дерева при разбиении имеет двух потомков. На каждом шаге построения дерева правило, находящееся в узле, разделяет обучающую выборку на две части: часть, в которой оно выполняется, и часть, в которой оно не выполняется. Смысл при построении дерева состоит в выборе среди всех возможных разбиений такого, для которого результирующие вершины-потомки являлись наиболее однородными.

Функция оценки качества разбиения, которая применяется для выбора оптимального правила, - индекс *Gini*. Она базируется на идее уменьшения неопределенности в узле. Если обучающая выборка T содержит n классов, тогда индекс *Gini* определяется как:

$$Gini(T) = 1 - \sum_{i=1}^{n} p_i^2$$

где p_i – вероятность (частота) класса i в T.

Если выборка T разбивается на две части T_1 и T_2 с числом наблюдений в каждом N_1 и N_2 соответственно, тогда показатель качества разбиения равен:

$$Gini_{split}(T) = \frac{N_1}{N} \cdot Gini(T_1) + \frac{N_2}{N} \cdot Gini(T_2)$$

Наилучшим является разбиение с минимальным $Gini_{split}(T)$.

Существенным особенностью, выделяющей CART среди других алгоритмов конструирования деревьев решений, является механизм отсечения дерева. Отсечение рассматривается как компромисс между получением дерева нужной глубины и получением точной оценки классификации. Отсечение важно не только для упрощения деревьев, но и для отсутствия переобучения. Идея отсечения заключается в получении последовательности уменьшающихся деревьев, при этом деревья рассматриваются не все, а только «лучшие представители» [75].

Перекрёстная проверка в алгоритме CART представляет собой способ выбора окончательного дерева, при условии, что выборка имеет небольшой объем или же наблюдения выборки настолько специфические, что разделить ее на обучающую и экзаменующую выборку не представляется возможным.

К критериям остановки алгоритма CART относятся:

– все наблюдения в вершине принадлежат одному классу;

– все наблюдения в вершине имеют одинаковые значения оставшихся атрибутов;

– глубина дерева достигала заранее заданного значения;

– число наблюдений в вершине не превосходит заранее заданного минимального значения.

Среди достоинств алгоритма CART можно выделить:

– CART легко борется с выбросами, так как механизм «разбиения», заложенный в алгоритме помещает выбросы в отдельный узел, что позволяет очистить имеющиеся данные от шумов.

– Алгоритм не требует принимать в расчет никаких предположений или допущений перед проведением анализа.

Среди недостатков алгоритма CART можно выделить:

– Деревья решений, строящиеся алгоритмом, не являются стабильными, так как результат, полученный на одной выборке, бывает не повторим на другой.

– CART может не идентифицировать правильную структуру данных при построении дерева с более сложной структурой.

Алгоритм C4.5

Алгоритм C4.5 предложен Куинланом (Quinlan) в 1984 [106]. Алгоритм относится к методике «Разделяй и властвуй», основанной на рекурсивном разбиении множества наблюдений из обучающей выборки на подмножества, содержащие наблюдения, относящиеся к одинаковым классам [9].

Имеется множество T, каждое наблюдение которого характеризуется списком атрибутов $A=(A_1,\ldots,A_n,C)$, где $C=(c_1,\ldots,c_k)$ – атрибут класса.

У алгоритма имеется три основных этапа:

1) Критерии остановки:

а) После очередного разбиения в вершине оказываются наблюдения, принадлежащие одному классу. Вершина становится листом, а класс, которому принадлежат её наблюдения, будет решением листа.

б) Вершина оказалась ассоциированной с пустым множеством. Она становится листом, а в качестве решения выбирается наиболее часто встречающийся класс у непосредственного предка этой вершины.

2) Выбор атрибута для разбиения. Выбирается атрибут A_i, который используется для разбиения выборки.

3) Построение дерева. Алгоритм выполняет следующим образом:

а) создает вершину дерева r с атрибутом A_i;

б) разбивает выборку на m подмножеств T_1, T_2, ... , T_m в соответствии со значениями атрибута A_i, рекурсивно вызывает процедуру построения дерева для каждого подмножества T_j с уменьшенным множеством атрибутов;

в) создает m ребер из r в корни деревьев T_1, T_2, ... , T_m, построенных в результате рекурсивных вызовов.

г) возвращает построенное дерево.

На втором этапе алгоритма при выборе атрибута A_i для разбиения единственной доступной информацией является распределение классов во

множестве T и его подмножествах, получаемых при разбиении. Данная информация используется при выборе атрибута A_i.

Пусть через $\Pr(C=c_i)$ обозначается вероятность того, что случайно выбранное наблюдение имеет метку класса c_i:

$$\Pr(C = c_i) = \frac{|T_i|}{|\mathrm{T}|},$$

где $|T_i|$ - мощность множества наблюдений обучающей выборки, относящихся к классу c_i.

В соответствии с определением энтропия выборки T относительно C равна: $H(T) = -\sum_{i=1}^{k} \Pr(C = c_i)\log_2(\Pr(C = c_i))$.

Определяется энтропия выборки T после разбиения, использующего атрибут A_i: $H_{A_i}(T) = \sum_{j=1}^{m} \frac{|T_j|}{|\mathrm{T}|} \cdot H(T_j)$.

Прирост информации (информационный выигрыш) – разница между энтропией до и после разбиения: $Gain(T, A_i) = H(T) - H_{A_i}(T)$.

Критерий *Gain* рассчитывается для всех независимых атрибутов, после чего выбирается атрибут с максимальным значением *Gain* [9].

При классификации нового наблюдения обход построенного дерева начинается с корня. На каждом внутреннем узле проверяется значение наблюдения по атрибуту, который соответствует проверке в данном узле. В зависимости от полученного ответа, находится соответствующее ветвление, продвигаются к узлу, находящему на уровень ниже и т.д. Обход дерева заканчивается при встрече узла решения, который и определяет класс для нового наблюдения.

Алгоритм C4.5 по сравнению со своим предшественником алгоритмом ID3 обладает следующими дополнительными возможностями:

– отсечение ветвей;

– работа с количественными атрибутами;

– построение дерева из обучающей выборки, в которой отсутствуют значения некоторых атрибутов.

Алгоритм Random Forest

Random forest (Случайный лес) – алгоритм, предложенный Лео Брейманом и Адель Катлер, в основе которого лежит использование комитета (ансамбля) деревьев принятия решений [76].

Суть алгоритма заключается в том, что на каждой итерации выбирают случайный набор переменных, после чего, на этом новом наборе запускают построение дерева принятия решений. При этом используется «бэггинг» – выборка случайных двух третей наблюдений выборки данных для обучения, а оставшаяся треть используется для оценки результата. Как правило, подобную операцию проделывают сотни или тысячи раз. Оптимальное число деревьев подбирается таким образом, чтобы минимизировать ошибку классификатора на тестовой выборке. В случае отсутствия тестовой выборки, минимизируется оценка ошибки out-of-bag: доля наблюдений обучающей выборки, неправильно классифицируемых комитетом, если не учитывать голоса деревьев на наблюдениях, входящих в их собственную обучающую подвыборку [76].

Классификация новых наблюдений проводится путем голосования: каждое дерево комитета относит классифицируемое наблюдение к одному из классов, и выигрывает класс, за который проголосовало наибольшее число полученных при моделировании деревьев.

Преимущества алгоритма Random Forest:

– высокое качество результата, особенно для данных с большим количеством переменных и малым количеством наблюдений;

– способность эффективно обрабатывать данные с большим числом признаков и классов.

Недостатки алгоритма Random Forest:

– каждое из деревьев огромное, в результате модель получается огромная;

– долгое построение модели, для достижения хороших результатов;

– сложная интерпретация модели, поскольку сотни или тысячи больших деревьев сложны для интерпретации.

Алгоритм Adaboost

AdaBoost (сокращение от Adaptive Boosting) — алгоритм усиления классификаторов, путем объединения их в комитет, предложенный Йоавом Фройндом (Yoav Freund) и Робертом Шапиром (Robert Schapire) [89, 90]. Этот алгоритм может использоваться в сочетании с несколькими алгоритмами классификации для улучшения их эффективности. AdaBoost является адаптивным в том смысле, что каждый следующий комитет классификаторов строится по наблюдениям, некорректно классифицированным предыдущими комитетами.

Алгоритм вызывает слабые классификаторы в цикле. После каждого вызова обновляется распределение весов, которые определяют важность каждого из наблюдений обучающего множества для классификации. На каждой итерации веса каждого некорректно классифицированного наблюдения возрастают, таким образом, новый комитет классификаторов строится, именно, по этим наблюдениям [19].

Достоинства алгоритма AdaBoost:

– хорошая обобщающая способность;

– простота реализации;

– возможность определить наблюдения, являющиеся выбросами.

Недостатки алгоритма AdaBoost:

– склонен к переобучению при наличии значительного уровня шума в данных;

– требует достаточно длинных обучающих выборок;

– может приводить к построению громоздких композиций, состоящих из сотен алгоритмов.

В таблице 3.19 проводится сравнительный анализ работы описанных выше алгоритмов при решении практических задач классификации.

Результаты работы алгоритмов построения 1-правил (1-R), RIPPER, CART, C4.5, Random Forest, Adaboost получены в системе анализа данных WEKA [115], а метода логического анализа данных (LAD) – с помощью программной системы, разработанной авторами [49]. Для тестирования использовались все практические задачи, рассмотренные в диссертационной работе. Выборки были разделены случайным образом на обучающую (80%) и тестовую (20%) для задач выявления спама и классификации результатов радарного сканирования ионосферы. Проведено по 20 экспериментов, результаты усреднены. Для задачи прогнозирования осложнений ИМ объем выборки используемый для тестирования по каждому осложнению определялся согласно таблице 3.3.

Таблица 3.19 – Результаты классификации

Задача	Алгоритм / Показатель	1-R	RIPPER	CART	C4.5	Random Forest	Adaboost	LAD
Выявление спама	Количество верно классифицированных наблюдений, %	82,6	91,3	90,2	90,2	89,1	91,3	92,4
Радарное сканирование ионосферы	Количество верно классифицированных наблюдений, %	78,6	82,8	82,8	81,4	84,2	88,5	90
ФП	Количество верно классифицированных наблюдений, %	58	66	62	70	70	74	76
ФЖ	Количество верно классифицированных наблюдений, %	87,3	86,7	63,3	83,3	68,3	90	90
РС	Количество верно классифицированных наблюдений, %	85,7	78,6	85,7	85,7	71,4	89,3	96,4
ОЛ	Количество верно классифицированных наблюдений, %	69,2	69,2	71,8	76,9	66,7	69,7	79,5
ЛИ	Количество верно классифицированных наблюдений, %	64	74	74	66	76	74	86

Каждый из исследуемых алгоритмов показал достаточно высокие результаты по точности в процессе исследования, однако наиболее приемлемыми для решения данных задач, согласно таблице 3.19, являются: LAD, Adaboost, RIPPER. Метод LAD в целом показал наилучший результат классификации, кроме того, он позволяет лучше учитывать специфику конкретной задачи классификации и требования заказчика (исследователя) при ее решении, обладает возможностью соблюдения баланса между различными критериями сравнения алгоритмов классификации.

Выводы

Создана программная система, реализующая разработанные алгоритмы с использованием подходов объектно-ориентированного программирования. Данная программная система используется в качестве системы поддержки принятия решений для задач классификации.

Решены с использованием метода логического анализа данных следующие задачи классификации:

– выявление спама;

– классификация результатов радарного сканирования ионосферы;

– прогнозирование осложнений инфаркта миокарда (фибрилляция предсердий, фибрилляция желудочков, разрыв сердца, отек легких, летальный исход).

Разработанные модификации для метода логического анализа данных доказали свою эффективность при решении практических задач. Во-первых, при использовании данных модификаций повышается интерпретируемость классификатора, так как сокращается количество закономерностей в нем. Во-вторых, повышается качество классификации новых наблюдений, т. е. улучшаются обобщающие способности классификатора.

Использование оптимизационной модели, позволяющей, чтобы правила покрывали некоторое ограниченное число наблюдений другого класса, приводит к поиску закономерностей с наиболее высоким покрытием, из

которых строится более точный классификатор. Применение такого подхода дает наилучший результат при решении задач с наличием выбросов и шумов и с большим количеством пропусков в выборке данных.

Установлено, что в результате применения алгоритмической процедуры наращивания закономерностей получаются закономерности с максимальным покрытием и с более высокой степенью, повышая надежность принимаемых решений классификатором, построенном на базе данных правил.

Метод логического анализа данных является достаточно гибким инструментом анализа данных, позволяющим учитывать специфику конкретной задачи классификации и требования заказчика (исследователя) путем целенаправленной настройки параметров метода.

Проведено сравнение по точности различных алгоритмов классификации с методом логического анализа данных. В результате данный метод показал наилучшие результаты по точности классификации, кроме того, он обладает возможностью соблюдения баланса между различными критериями сравнения алгоритмов классификации.

ЗАКЛЮЧЕНИЕ

В ходе выполнения диссертационной работы получены следующие результаты:

1. Проведен анализ существующих логических алгоритмов классификации, алгоритмов поиска информативных закономерностей для них, и основных программных систем, решающих практические задачи классификации. Отмечено, что для классификации наиболее приемлем алгоритм, основанный на голосовании правил.

2. Разработана алгоритмическая процедура выбора базовых наблюдений для формирования закономерностей, отличающаяся от известных целенаправленным выбором базовых наблюдений, получаемых путем применения алгоритма «k-средних» к множеству наблюдений обучающей выборки.

3. Разработана алгоритмическая процедура наращивания закономерностей, полученных на базе оптимизационной модели с максимальным покрытием наблюдений обучающей выборки.

4. Создана модель оптимизации для формирования закономерностей, отличающаяся от известных наличием в целевой функции весового коэффициента покрываемого наблюдения, а также возможностью захвата наблюдений другого класса.

5. Разработана алгоритмическая процедура построения классификатора как композиции информативных закономерностей, отличающаяся от известных совместным использованием критерия бустинга для оценки информативности закономерностей и новой итеративной процедуры выбора порога информативности.

6. Модифицирован метод логического анализа данных на основе разработанных алгоритмических процедур, при использовании которых повышается интерпретируемость классификатора и качество классификации

новых наблюдений, т. е. улучшаются обобщающие способности классификатора.

7. В результате решения практических задач эмпирически проверена пригодность оптимизационных моделей для формирования информативных закономерностей и эффективность разработанных алгоритмических процедур для метода логического анализа данных.

8. Проведено сравнение по точности метода логического анализа данных с другими алгоритмами классификации на практических задачах. В результате метод показал лучшие результаты по точности решения предложенных задач.

Таким образом, в диссертационной работе разработаны, исследованы и проверены на практических задачах модификации для метода логического анализа данных, основанные на создании оптимизационных моделей для формирования информативных закономерностей и алгоритмических процедур сокращения количества правил в классификаторе при сохранении высокой точности, что является вкладом в теорию и практику интеллектуального анализа данных.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Айвазян, С. А. Прикладная статистика: классификация и снижение размерности / С.А. Айвазян, В.М. Бухштабер, И.С. Енюков, Л.Д. Мешалкин – М.: Финансы и статистика, 1989 – 607 с., ил

2. Айзерман, М.А., Браверман Э.М., Розоноэр Л.И. Метод потенциальных функций в теории обучения машин / М.А. Айзерман, Э.М. Браверман, Л.И. Розоноэр - М.: Наука, 1970. – 383 с.

3. Антамошкин, А.Н. Не улучшаемый алгоритм условной оптимизации монотонных псевдобулевых функций / А. Н. Антамошкин, И.С. Масич // Электронный журнал «Исследовано в России» – 2004 – № 64 – С. 703-708. http://zhurnal.ape.relarn.ru/articles/2004/064.pdf.

4. Антамошкин, А.Н. Гриди алгоритмы и локальный поиск для условной псевдобулевой оптимизации / А. Н. Антамошкин, И.С. Масич // Электронный журнал «Исследовано в России» – 2003 – № 177 – С. 2143-2149. http://zhurnal.ape.relarn.ru/articles/2003/177.pdf

5. Антамошкин А.А. Идентификация свойств псевдобулевых функций / А. Н. Антамошкин, И.С. Масич // Электронный журнал «Исследовано в России» - 2004. № 130, С. 1391-1396. http://zhurnal.ape.relarn.ru/articles/2004/130.pdf.

6. Антамошкин, А.Н. Исследование свойств задачи оптимизации при поиске логических закономерностей в данных / А. Н. Антамошкин, И.С. Масич // Научно-технический журнал: «Системы управления и информационные технологии». - N4.1(46), 2011г. – С. 111-115.

7. Архангельский, А.Я. Программирование в Delphi 7. / А.Я. Архангельский. – М: ООО «Бином-пресс», 2003. – 1152 с.

8. Барабаш, Ю. Л. Автоматическое распознавание образов / Ю. Л. Барабаш, Б. В. Варский, В. Т. Зиновьев и др. Киев: изд. КВАИУ, 1963. – 168 с.: ил.

9. Барсегян, А.А. Метод и модели анализа данных: OLAP и Data Mining / А.А. Барсегян, М.С. Куприянов, В.В. Степаненко, И.И. Холод. – СПб.: БХВ-Петербург, 2004. – 336 с.: ил.

10. Вайнцвайг, М. Н. Алгоритм обучения распознаванию образов «кора» // Алгоритмы обучения распознаванию образов / Под ред. В. Н. Вапник. - М.: Советское радио, 1973.- С. 110-116.

11. Воронцов, К.В. Лекции по логическим алгоритмам классификации [Электронный ресурс] / К. В. Воронцов – 2007. http://www.ccas.ru/voron/download/LogicAlgs.pdf.

12. Гладков, Л. А. Генетические алгоритмы./ Л.А. Гладков, В.В. Курейчик, В.М. Курейчик. - М.: Физматлит, 2006. - 320 с.

13. Головенкин, С.Е. Модель логического анализа для решения задачи прогнозирования инфаркта миокарда / С.Е. Головенкин, Т.К. Гулакова, Р.И. Кузьмич, И.С. Масич, В.А. Шульман // Вестник СибГАУ. - Вып. 4 (30). – 2010. – С. 68-73.

14.Головенкин, С.Е. Осложнения инфаркта миокарда: база данных для апробации систем распознавания и прогноза / С.Е. Головенкин, А.Н. Горбань, В.А. Шульман и др. – Красноярск, Вычислительный центр СО РАН: Препринт – №6 – 1997.

15. Горбань, А.Н. Нейронные сети на персональном компьютере / А.Н. Горбань, Д.А. Россиев. - Новосибирск.: «Наука», 1996. - 276 с.

16. Гулакова, Т.К. Поиск закономерностей в задаче классификации / Т.К. Гулакова, Р.И. Кузьмич // Материалы VI всероссийской научно-практической конференции студентов, аспирантов и молодых специалистов «Актуальные проблемы авиации и космонавтики»: в 2 т. Т. 1. Технические науки / Сиб. гос. аэрокосмич. ун-т. – Красноярск, 2010. - С. 317-318.

17. Дегтерев, Д.А. Оптимизация загрузки технологического оборудования предприятия / Д.А. Дегтерев, И.С. Масич, Г.А. Нейман // Вестник ассоциации выпускников КГТУ, Красноярск: ИПЦ КГТУ, 2002, Вып. 8, С. 166-170.

18. Донской, В.И. Дискретные модели принятия решений при неполной информации / В.И. Донской, А.И. Башта // – Симферополь: Таврия, 1992. – 166 с.

19. Дьяконов, А. Г. Анализ данных, обучение по прецедентам, логические игры, системы WEKA, RapidMiner и MatLab (Практикум на ЭВМ кафедры математических методов прогнозирования): Учебное пособие. – М.: Издательский отдел факультета ВМК МГУ имени М.В. Ломоносова, 2010. – 278 с.

20. Дюличева, Ю.Ю. Стратегии редукции решающих деревьев (обзор) / Ю.Ю. Дюличева // Таврический вестник информатики и математики. - 2002.- № 1. - С. 10–17.

21. Емельянов, В. В. Теория и практика эволюционного моделирования / В.В. Емельянов, В.В. Курейчик, В.М. Курейчик - М.: Физматлит, 2003. - 432 с.

22. Журавлев, Ю.И. Распознавание. Математические методы. Программная система. Практические применения / Ю.И. Журавлев, В.В. Рязанов, О.В. Сенько – М.: ФАЗИС, 2006. – с. 159

23. Загоруйко, Н. Г. Прикладные методы анализа данных и знаний / Н.Г. Загоруйко– Новосибирск: Из-во ин-та математики, 1999 – 270 с.

24.Загоруйко, Н. Г. Методы распознавания, основанные на алгоритме AdDel / Н. Г. Загоруйко, О.А. Кутненко // Сиб. журн. индустр. матем., **7**:1 (2004), С. 39-47

25. Кардиология в таблицах и схемах. / под ред. М. Фрида, С. Грайнс; пер. с англ. – М.: Практика, 1996. – 736 с.

26. Костюк, Ф.Ф. Инфаркт миокарда. / Ф.Ф. Костюк. – Красноярск, 1993 – 224 с.

27. Кузьмич, Р.И. Применение логических алгоритмов классификации для решения задач диагностики медицинских заболеваний / Р.И. Кузьмич // Материалы VII всероссийской научно-практической конференции студентов, аспирантов и молодых специалистов «Актуальные проблемы авиации и

космонавтики»: в 2 т. Т. 1. Технические науки / Сиб. гос. аэрокосмич. ун-т. – Красноярск, 2011. - С.324-325.

28. Кузьмич, Р.И. The determination of important attributes in the prognosis task of myocardial infarction complications / Р.И. Кузьмич, Т.К. Гулакова // Материалы X Всероссийской студенческой научной конференции на иностранных языках «Молодежь. Общество. Современная наука, техника и инновации». - Сиб. гос. аэрокосмич. ун-т: Красноярск, 2011. – С. 37-39.

29. Кузьмич, Р.И. Classification accuracy of logical data analysis comparison on full and truncated set of attributes / Р.И. Кузьмич // Материалы XI Международной научной конференции аспирантов, магистрантов, бакалавров и школьников «Молодежь. Общество. Современная наука, техника и инновации» / Сиб. гос. аэрокосмич. ун-т. – Красноярск, 2012. – С. 63-65.

30. Кузьмич, Р. И. Перспективность создания программной системы на основе метода логического анализа данных / Р.И. Кузьмич // Материалы 50-й Международной научной студенческой конференции «Студент и научно-технический прогресс»: Информационные технологии / Новосиб. гос. ун-т. Новосибирск, 2012г. – С. 125.

31. Кузьмич, Р.И. Создание программной системы для решения задачи прогнозирования осложнений инфаркта миокарда / Р.И. Кузьмич // Материалы XLIX Международной научной студенческой конференции «Студент и научно-технический прогресс»: Информационные технологии / Новосиб. гос. ун-т. Новосибирск, 2011г. – С. 115.

32. Кузьмич, Р.И. Поиск закономерностей при решении задачи управления приземлением космического корабля / Р.И. Кузьмич // Материалы VIII Всероссийской научной-практической конференции творческой молодежи «Актуальные проблемы авиации и космонавтики», посвященной 55-летию запуска первого искусственного спутника Земли: в 2 т. Т. 1. Технические науки. Информационные технологии. Сообщения школьников. - Сиб. гос. аэрокосмич. ун-т. – Красноярск, 2012. – С. 306-307.

33. Кузьмич, Р.И. Взвешенное голосование правил в задаче классификации данных / Р.И. Кузьмич // Тезисы IX Всероссийской научно-практической конференции творческой молодежи «Актуальные проблемы авиации и космонавтики»: в 2 т. Т. 1. Технические науки. Информационные технологии. Сообщения школьников. - Сиб. гос. аэрокосмич. ун-т. – Красноярск, 2013. – С. 335-336.

34. Кузьмич, Р.И. Построение модели классификации как композиции информативных паттернов / Р.И. Кузьмич, И.С. Масич // Научно-технический журнал: «Системы управления и информационные технологии». - N2 (48), 2012г. – С. 18-22.

35. Кузьмич, Р.И. Применение информативных паттернов для построения модели классификации при решении задач медицинской диагностики / Р.И. Кузьмич, А.А. Ступина, И.С. Масич // Материалы Всероссийской научно-технической конференции студентов, аспирантов и молодых ученых «Научная сессия ТУСУР–2013». – Томск: В-Спектр, 2013: В 5 частях. – Ч. 3. – С. 183-186.

36. Кузьмич, Р.И. Генерирование объектов для построения паттернов с целью сокращения модели классификации / Р.И. Кузьмич, И.С. Масич // Материалы XV международной научной конференции «Решетневские чтения», Сиб. гос. аэрокосмич. ун-т. - Красноярск, 2011г. – Ч. 2. - С. 462-463.

37. Кузьмич, Р.И. Применение процедуры кластеризации для генерирования объектов с целью сокращения числа паттернов в модели классификации / Р.И. Кузьмич, А.И. Виноградова // Вестник КрасГАУ. – 2013. № 9(84). – Красноярск, 2013. – С. 51-55.

38. Кузьмич, Р.И. Определение важности признаков при формировании паттернов в задаче классификации / Р.И. Кузьмич // Материалы XIV международной научной конференции «Решетневские чтения», Сиб. гос. аэрокосмич. ун-т. - Красноярск, 2010г. – Ч. 2. - С. 394-395.

39. Кузьмич, Р.И. Способы бинаризации разнотипных признаков в задачах классификации / Р.И. Кузьмич, Т.К. Гулакова // Материалы VI всероссийской научно-практической конференции студентов, аспирантов и

молодых специалистов «Актуальные проблемы авиации и космонавтики»: в 2 т. Т. 1. Технические науки / Сиб. гос. аэрокосмич. ун-т. – Красноярск, 2010. - С.323-325.

40. Кузьмич, Р.И. Обоснование создания программной системы на основе метода логического анализа данных / Р.И. Кузьмич // Материалы 51-й международной научной студенческой конференции «Студент и научно-технический прогресс»: Информационные технологии / Новосиб. гос. ун-т. Новосибирск, 2013г – С. 229.

41. Лидовский, В.В. Теория информации: Учебное пособие / В.В. Лидовский. - М.: Компания Спутник+, 2004. – 111 с.

42. Лбов, Г.С. Методы обработки разнотипных экспериментальных данных / Г.С. Лбов. – Новосибирск: Наука, 1981. – 160 с.

43. Лбов, Г.С. Метод обнаружения логических закономерностей на эмпирических таблицах / Г.С. Лбов, В.И. Котюков, Ю.П. Машаров // - Вычислительные системы, 1976, вып. 67, С. 29-42.

44. Масич, И.С. Оптимизация загрузки производственных мощностей литейного производства / И.С. Масич, К.В. Шарыпова // Научно-технический журнал: «Системы управления и информационные технологии». – №3(29), 2007г. – С. 76-80.

45. Масич, И.С. Модель логического анализа для прогнозирования осложнений инфаркта миокарда / И.С. Масич // Информатика и системы управления – №3(25), 2010 – С. 48-56.

46. Масич, И.С. Комбинаторная оптимизация в задаче классификации / И.С. Масич // Научно-технический журнал: «Системы управления и информационные технологии». – №1.2(35), 2009г. – С. 283-288.

47. Масич, И.С. Приближенные алгоритмы поиска граничных точек для задачи условной псевдобулевой оптимизации / И.С. Масич // Вестник СибГАУ. – Вып. 1(8) – 2010 – С. 39-43.

48. Масич, И.С. Поисковые алгоритмы псевдобулевой оптимизации в задаче классификации данных / И.С. Масич, Р.И. Кузьмич // Материалы XV

международной научной конференции «Решетневские чтения», Сиб. гос. аэрокосмич. ун-т. - Красноярск, 2011г. – Ч. 2. - С. 472-473.

49. Масич, И.С. Логический анализ данных в задачах классификации / И.С. Масич, Р.И. Кузьмич, Е.М. Краева // – М: Роспатент, 2011. № гос. рег. 2011612265.

50. Масич, И.С. Сравнительный анализ методов классификации данных на практических задачах прогнозирования и диагностики / И.С. Масич, Е.М. Краева, Р.И. Кузьмич, Т.К. Гулакова // Научно-технический журнал: «Системы управления и информационные технологии». - N1(43), 2011г. – С. 20-25.

51. Масич, И.С. Сравнение методов классификации данных на практических задачах прогнозирования и диагностики / И.С. Масич, Р.И. Кузьмич // Материалы III Международной молодежной научно-технической конференции «Молодежь, техника, космос» - Санкт-Петербург: БГТУ,2011. – С. 215-217.

52. Перегудов, Ф.И. Основы системного анализа: Учеб. 2-е изд., доп. / Ф.И. Перегудов, Ф.П. Тарасенко – Томск: Изд-во НТЛ, 1997. – 396 с.: ил.

53. Растригин, Л.А. Решение задач разношкальной оптимизации методами случайного поиска / Л.А. Растригин, Э.Э. Фрейманис// Проблемы случайного поиска – 1988 – №11 – С. 9-25.

54. Редько, В. Г. Эволюционная кибернетика / В.Г. Редько. - М.: Наука, 2003. - 155 с.

55. Россиев, Д.А. Прогнозирование осложнений инфаркта миокарда нейронными сетями / Д.А. Россиев, С.Е. Головенкин, В.А. Шульман, Г.В. Матюшин // Нейроинформатика и ее приложения. Материалы III Всероссийского рабочего семинара. 6-8 октября 1995 г. Красноярск.- 1995.- С. 128-166.

56. Руда, М.Я. Инфаркт миокарда / М.Я. Руда, А.П. Зыско – М.: «Медицина», 1981. – 288 с.

57. Ступина, А.А. Программная реализация логических алгоритмов классификации для прогнозирования осложнений инфаркта миокарда / А.А.

Ступина, Р.И. Кузьмич// Материалы Всероссийской молодежной научной конференции с международным участием «Современные проблемы фундаментальных и прикладных наук» / ФГБОУ ВПО «Кемеровский технологический институт пищевой промышленности» Кемерово, Кузбассвузиздат; 2011. – С. 84-87.

58. Ступина, А.А. Разработка программной системы на основе логических алгоритмов классификации для решения задач медицинской диагностики и прогнозирования / А.А. Ступина, И.С. Масич, Р.И. Кузьмич, О.Г. Ступин // Сборник научных трудов по материалам XIV Международной научно-технической конференции «Фундаментальные и прикладные проблемы приборостроения и информатики» – М.: МГУПИ, 2011. – С. 112-117.

59. Ступина, А.А. Особенности формирования программного обеспечения для систем управления техническими объектами / А.А. Ступина, А.И. Пережилин, Л.Н. Корпачева, Р.И. Кузьмич // Вестник КрасГАУ. – 2011. № 1(40). – Красноярск, 2011. – С. 12-16.

60. Сыркин, А.Л. Инфаркт миокарда. / А. Л. Сыркин. – М.: «Медицина», 1991. – 303 с.

61. Уоссермен, Ф. Нейрокомпьютерная техника. Теория и практика / Ф. Уоссермен; пер. с англ – Ю.А. Зуев, В.А. Точенов, 1992. – 184 с.

62. Фаронов, В.В. Delphi 6. Учебный курс / В.В. Фаронов. – М.: издатель Молгачева С.В., 2001. – 672 с., ил.

63. Чазов, Е.И. Болезни сердца и сосудов / Е.И. Чазов. - В 2-х т. Т.2. / М.: «Медицина», 1992. – 512 с.

64. Шеннон, К. Э. Математическая теория связи // Работы по теории информации и кибернетике / Пер. С. Карпова. — М.: ИИЛ, 1963. — 830 с.

65. Alexe, G. Logical Analysis of the Proteomic Ovarian Cancer Dataset / G. Alexe, S. Alexe, P.L. Hammer, L. Liotta, E. Petricoin, M. Reiss // RUTCOR Technical Report, RTR 2-2002, 2002. [Electronic resource]. URL: http://rutcor.rutgers.edu/ ~rrr/rtr/2-2002.pdf.

66. Alexe, G. Combinatorial analysis of breast cancer data from image cytometry and gene expression microarrays / G. Alexe, S. Alexe, D. Axelrod, E. Boros, P.L. Hammer, M. Reiss // RUTCOR Technical Report, RTR 3-2002, 2002. [Electronic resource]. URL: http://gatekeeper.dec.com/pub/toomany/paper151.pdf.

67. Alexe, G. Pattern-based feature selection in genomics and proteomics / G. Alexe, S. Alexe, P.L. Hummer, B. Vizvari // RUTCOR Research Report 7-2003, March 2003. [Electronic resource]. URL: http://rutcor.rutgers.edu/pub/rrr/reports2003/7_2003.pdf.

68. Alexe, S. Coronary Risk Prediction by Logical Analysis of Data // S. Alexe, E. Blackstone, P.L. Hammer and others / Annals of Operations Research – 2003 – 119 – Pp. 15-42.

69. Antamoshkin, A.N. Identification of pseudo-Boolean function properties / A.N. Antamoshkin, I.S. Masich // Engineering & automation problems (Проблемы машиностроения и автоматизации). – 2007 – №2 – Pp. 66-69.

70. Antamoshkin, A.N. Heuristic search algorithms for monotone pseudo-boolean function conditional optimization / A.N. Antamoshkin, I.S. Masich // Engineering & automation problems (Проблемы машиностроения и автоматизации). – 2006 – V. 5, N. 1. – Pp. 55-61.

71. Antamoshkin A.N. Pseudo-Boolean optimization in case of unconnected feasible sets / A.N. Antamoshkin, I.S. Masich // Models and Algorithms for Global Optimization. Series: Springer Optimization and Its Applications, Vol. 4, edited by A. Törn, J. Žilinskas. – Springer, 2007 – XVI. – Pp. 111-122.

72. Bradley, P.S. Feature selection via concave minimization and support vector machines / P.S. Bradley, O.L. Mangasarian // In J. Shavlik, editor, Proceedings of the Fifteenth International Conference on Machine Learning, Morgan Kaufmann, San Francisco, CA, (1998). - Pp. 82-90.

73. Brauner, M.W. Logical analysis of computer tomography data to differentiate entities of idiopathic interstitial pneumonias / M.W. Brauner, D. Brauner, P.L. Hammer, I. Lozina, D. Valeyre // RUTCOR Research Report 30-2004,

2004. [Electronic resource]. URL: http://rutcor.rutgers.edu/pub/rrr/reports2004/30_2004.pdf

74. Boros, E. An implementation of logical analysis of data / E. Boros, P.L. Hammer, T. Ibaraki, A. Kogan // RUTCOR Research Report 22-96, 1996.

75. Breiman, L. Classification and Regression Tree / L. Breiman, J.H. Friedman, R. Olshen, C.J. Stone // Wadsworth & Brooks/Cole Advanced Books & Software, Pacific California, 1984.

76. Breiman, L. Random Forests / L. Breiman // Machine Learning 45 (1): 5–32, 2001.

77. Breslow, L.A. Simplifying Decision Trees: A Survey / L.A. Breslow, D.W. Aha // Knowledge Engineering Review 12. – 1997. – Pp. 1-40.

78. Breslow, L.A. Comparing Tree-Simplification Procedures / L.A. Breslow, D.W. Aha // Nave Center for Applied Research in Artificial Intelligence, Technical Report No. AIC-96-015. – 1996. – Pp. 1-10.

79. Burges, Christopher J.C. A Tutorial on Support Vector Machines for Pattern Recognition – 1998 г. – [Electronic resource]. URL: http://research.microsoft.com/en-us/um/people/cburges/papers/SVMTutorial.pdf.

80. Chtioui, Y., Bertrand, D., Barba, D. Feature selection by a genetic algorithm / Y. Chtioui, D. Bertrand, D. Barba // Application to seed discrimination by artificial vision, Journal of the Science of Food and Agriculture, 76 (1), (1998). - Pp. 77-86.

81. Cohen, W. W. Fast Effective Rule Induction / W. W. Cohen // Machine Learning: Proceedings of the 12th International Conference, Lake Tahoe, California: Morgan Kaufman, 1995. [Electronic resource]. URL: http://cs.utsa.edu/~bylander/cs6243/cohen95ripper.pdf

82. Cohen, W. W. A simple, fast and effective rule learner / W.W. Cohen, Y. Singer // Proc. of the 16 National Conference on Artificial Intelligence. - 1999. - Pp. 335–342.

83. Cremilleux, B. Use of Attribute Selection Criteria in Decision Trees in Uncertain Domains / B. Cremilleux, C. Robert // Uncertainty in Intelligent and

Information Systems, Advances in Fuzzy Systems, Application and Theory. – World Scientific. – 2000. – Vol.20. – Pp. 150-161.

84. Dash, M. Feature selection for classification / M. Dash, H. Lui // Intelligent data analysis – 1997 – №1, (3) – Pp. 131-156

85. De Jong, K. A. Genetic Algorithms: A 10 Year Perspective / K.A. De Jong //In: Procs of the First Int. Conf. on Genetic Algorithms, 1985. – Pp. 167 – 177.

86. Dubner, P.N. Statistical tests for feature selection in KORA recognition algorithms / P.N. Dubner // Pattern Recognition and Image Analysis. - 1994. - Vol. 4, no. 4. - Pp. 396.

87. Elomaa, T. An Analyses of Reduced Error Pruning / T. Elomaa, M. Kaariainen // Journal of Artificial Intelligence Research – 2001. – Vol.15. – Pp. 163-187.

88. Esposito, F. Comparative Analysis of Methods for Pruning Decision Trees / F. Esposito, D. Malerba, G.A. Semeraro // IEEE Transactions on Pattern Analyses and Machine Intelligence. - 1997. - Vol. 19(5). - Pp. 476-491.

89. Freund, Y. A decision-theoretic generalization of on-line learning and an application to boosting / Y. Freund, R.E. Schapire // European Conference on Computational Learning Theory. - 1995. - Pp. 23-37.

90. Freund, Y. A Short Introduction to Boosting / Y. Freund, R.E. Schapire // Journal of Japanese Society for Artificial Intelligence, 14(5):771-780, September – 1999 г. – [Electronic resource]. URL: http://www.yorku.ca/gisweb/eats4400/boost.pdf.

91. Furnkranz, J. Roc ‘n’ rule learning-towards a better understanding of covering algorithms / J. Furnkranz, P.A. Flach // Machine Learning. - 2005. - Vol. 58, no. 1. - Pp. 39–77.

92. Hammer, P.L. The Logic of Cause-effect Relationships / P.L. Hammer // Lecture at the International Conference on Multi-Attribute Decision via Operations Research-based Expert Systems. – Passau, Germany, 1986.

93. Hammer, P.L., Kogan, A., Lejeune, M. Modeling Country Risk Ratings Using Partial Orders / P.L. Hammer, A. Kogan, M. Lejeune // RUTCOR Research

Report 24-2004, 2004. [Electronic resource]. URL: http://rutcor.rutgers.edu/pub/rrr/reports2004/24_2004.pdf.

94. Hammer, P.L. Logical Analysis of Data: From Combinatorial Optimization to Medical Applications / P.L. Hammer, T. Bonates // RUTCOR Research Report 10-2005, 2005. [Electronic resource]. URL: http://rutcor.rutgers.edu/pub/rrr/reports2005/10_2005.pdf.

95. Ho, T.K. C4.5 Decision Forests / T.K. Ho // Proceedings of the 14th International Conference of Pattern Recognition, Brisbane, Australia. – 1998. – Pp. 17-20.

96. Hutter F. Efficient Stochastic Local Search for MPE Solving / F. Hutter, H. H. Hoos and T. Stützle // - Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI-05). - 2005. - Pp. 169-174.

97. Kothari, R. Decision Trees for Classification: A Review and Some New Results / R. Kothari, M. Dong // Pattern Recognition: From Classical to Modern Approaches, S.R. Pal (Eds.). – Chapter 6, World Scientific. – 2001. – Pp. 169-184.

98. Leray, P. Feature selection with neural networks / P. Leray, P. Galliary // [Electronic resource]. URL: http://www.idrbt.ac.in/education/sample2%201/Seminar/leray98feature.pdf.

99. Lui, H. Feature selection for knowledge discovery and data mining / H. Lui, H. Motoda // Kluwer Academic publishers – 1998. – P. 214

100. Lui, H. Feature extraction, construction and selection: A data missing perspective / H. Lui, H. Motoda // Kluwer Academic publishers – 1998. – P. 410

101. Loh, W.-Y. Split Selection Methods for Classification Trees / W.-Y. Loh // Statistica Sinica. – 1997. – Vol.7. – Pp. 815-840.

102. Mahmoud, H.M. On Tree-Growing Search Strategies / H.M. Mahmoud // The Annals of Applied Probability. – 1996. – Vol.6. – P.1284-1302.

103. Marchand M. Learning with the set covering machine / M. Marchand, J. Shawe-Taylor // Proc. 18th International Conf. on Machine Learning. - Morgan Kaufmann, San Francisco, CA, 2001. - Pp. 345–352.

104. Merill T. On the effectiveness of receptors in recognition systems / T. Merill, O.M. Green // IEEE Trans. Inform. Theory. 1963. V. IT-9. Pp. 11-17

105. Quinlan, J.R. Induction of decision trees / J.R. Quinlan // Machine Learning. - 1986. - Vol. 1, no. 1. - Pp. 81-106.

106. Quinlan, J.R. Bagging, Boosting, and C4.5 / J.R. Quinlan // Proceedings of 13th National Conference on Artificial Intelligence. – 1996. – Pp. 725-730.

107. Quinlan, J.R. Improved Use of Continuous Attributes in C4.5 / J.R. Quinlan // Journal of Artificial Intelligence Research. – 1996. – Vol.4. – Pp. 77-90.

108. RapidMiner – PredictiveAnalytics, Data Mining, Self-service, open source. [Electronic resource]. URL: http://www.rapidminer.com.

109. Rivest, R.L. Learning decision lists / R.L. Rivest // Machine Learning. - 1987. - 2 (3) – Pp. 229–246

110. Rossiev, D.A. Forecasting of myocardial infarction complications with the help of neural networks / D.A. Rossiev, S.E. Golovenkin, V.A. Shulman, G.V. Matyushin // Proc. WCNN'95. (World Congress on Neural Networks' 95). - Washington, DC, July 1995. - Pp. 185-188.

111. Rossiev, D.A. The employment of neural network to model implantation of pasemaker in patients with arrhythmias and heart blocks / D.A. Rossiev, S.E. Golovenkin, V.A. Shulman, G.V. Matyushin // Modelling, Measurement & Control. - 1995.-V.48. - N.2.- Pp.39-46.

112. Setiono, R. Neural network Feature selection / R. Setiono, H. Lui // IEEE Transaction on neural networks, 1997 – №8 (3) – Pp. 654-662.

113. UCI Machine Learning Repository [Electronic resource]. URL: http://archive.ics.uci.edu/ml/index.html.

114. Wegener, I. On the Optimization of Monotone Polynomials by Simple Randomized Search Heuristics / I. Wegener, C. Witt // Combinatorics, Probability and Computing / Volume 14 / Issue 1-2 / January 2005, Pp. 225-247

115. Weka 3 - Data Mining with Open Source Machine Learning Software in Java [Electronic resource]. URL: http://www.cs.waikato.ac.nz/~ml/weka/index.html.

ПРИЛОЖЕНИЕ А

(Справочное)

Названия полей базы данных и расшифровка их значений

(В скобках даны сокращения названий полей, используемые в структуре базы данных)

1.Возраст. (AGE).

2.Пол: (SEX).

0 - женский

1 - мужской

3.Количество инфарктов миокарда в анамнезе: (INF_ANAM).

0 - нет

1 - один

2 - два

3 - три и.т.д.

4.Стенокардия напряжения в анамнезе: (STENOK_AN).

0 - нет

1 - менее 1 года

2 - один год

3 - два года

4 - три года

5 - 4-5 лет

6 - более 5 лет

5.Функциональный класс стенокардии в последний год: (FK_STENOK).

0 - нет стенокардии

1 - I ф.к.

2 - II ф.к.

3 - III ф.к.

4 - IV ф.к.

6.Характер ИБС в последние недели, дни перед пост. в больницу:

(IBS_POST).

0 - нет ИБС

1 - стенокардия напряжения

2 - нестабильная стенокардия

7.Наличие гипертонической болезни: (GB).

0 - нет Г.Б.

1 - I стадия Г.Б.

2 - II стадия Г.Б.

3 - III стадия Г.Б.

8.Симптоматическая гипертония: (SIM_GIPERT).

0 - нет

1 - да

9.Длительность течения арт. гипертензии: (DLIT_AG).

0 - нет А.Г.

1 - год

2 - два года

3 - три года

4 - четыре года

5 - пять лет

6 - 5-10 лет

7 - более 10 лет

10.Наличие хронической сердечной недостаточности (СН) в анамнезе: (ZSN_A).

0 - нет

1 - I стадии

2 - IIA стадия (застой по большому кругу)

3 - IIA стадия (застой по малому кругу)

4 - IIБ стадия

5 - III стадия

11.Нарушения ритма в анамнезе, не уточнено какие именно (nr11).

0 - нет

1 - да

12.Предсердная экстрасистолия в анамнезе (nr01).
0 - нет
1 - да
13. Желудочковая экстрасистолия в анамнезе (nr02).
0 - нет
1 - да
14.Пароксизмы фибрилляции/трепетании предсердий в анамнезе (nr03).
0 - нет
1 - да
15.Постоянная форма фибрилляции предсердий в анамнезе (nr04).
0 - нет
1 - да
16.Желудочковая пароксизмальная тахикардия в анамнезе (nr05).
0 - нет
1 - да
17.Фибрилляция желудочков в анамнезе (nr07).
0 - нет
1 - да
18.А-в блокада I степени в анамнезе (np01).
0 - нет
1 - да
19.А-в блокада III степени в анамнезе (np04).
0 - нет
1 - да
20.Блокада передней ветви левой ножки пучка Гиса в анамнезе (np05).
0 - нет
1 - да
21.Неполная блокада левой ножки пучка Гиса в анамнезе (np07).
0 - нет
1 - да
22.Полная блокада левой ножки пучка Гиса в анамнезе (np08).
0 - нет
1 - да
23.Неполная блокада правой ножки пучка Гиса в анамнезе (np09).
0 - нет
1 - да
24.Полная блокада правой ножки пучка Гиса в анамнезе (np10).
0 - нет
1 - да
25.Сахарный диабет в анамнезе (endocr_01).
0 - нет
1 - да
26.Ожирение в анамнезе (endocr_02).
0 - нет
1 - да
27.Тиреотоксикоз в анамнезе (endocr_03).
0 - нет
1 - да
28.Хронический бронхит в анамнезе (zab_leg_01).
0 - нет
1 - да
29.Обструктивный хронический бронхит в анамнезе (zab_leg_02).
0 - нет
1 - да
30.Бронхиальная астма в анамнезе (zab_leg_03).
0 - нет
1 - да
31.Хроническая пневмония в анамнезе (zab_leg_04).
0 - нет

1 - да

32.Туберкулез легкого (легких) в анамнезе (zab_leg_06).

0 - нет

1 - да

33.Систолическое АД по данным кардиобригады (S_AD_KBRIG).

34.Диастолическое АД по данным кардиобригады (D_AD_KBRIG).

35.Систолическое.АД по данным ОРиИТ (S_AD_ORIT).

36.Диастолическое.АД по данным ОРиИТ (D_AD_ORIT).

37.Отек легких в момент поступления в ОРиИТ: (O_L_POST).

0 - нет

1 - да

38.Кардиогенный шок в момент поступления в ОРиИТ: (K_SH_POST).

0 - нет

1 - да

39.Пароксизм фибрилляции предсердий (ТП) в момент поступления в ОРиИТ, (или на догоспитальном этапе): (MP_TP_POST).

0 - нет

1 - да

40.Пароксизм суправентрикулярной тахикардии в момент поступления в ОРиИТ, (или на догоспитальном этапе): (SVT_POST).

0 - нет

1 - да

41.Пароксизм желудочковой тахикардии в момент поступления в ОРиИТ, (или на догоспитальном этапе): (GT_POST).

0 - нет

1 - да

42.Фибрилляция желудочков в момент поступления в ОРиИТ, (или на догоспитальном этапе): (FIB_G_POST).

0 - нет

1 - да

43.Наличие инфаркта передней стенки левого желудочка (изменения на ЭКГ в отведениях V2 - V4) (ant_im).

0 - нет

1 - форма комплекса QRS не изменена

2 - форма QRS комплекса QR

3 - форма QRS комплекса Qr

4 - форма QRS комплекса QS

44.Наличие инфаркта боковой стенки левого желудочка (изменения на ЭКГ в отведениях V5 - V6, I, AVL. (lat_im).

0 - нет

1 - форма комплекса QRS не изменена

2 - форма QRS комплекса QR

3 - форма QRS комплекса Qr

4 - форма QRS комплекса QS

45.Наличие инфаркта нижней стенки левого желудочка (изменения на ЭКГ в отведениях III, AVF, II). (inf_im).

0 - нет

1 - форма комплекса QRS не изменена

2 - форма QRS комплекса QR

3 - форма QRS комплекса Qr

4 - форма QRS комплекса QS

46.Наличие инфаркта задней стенки левого желудочка (изменения на ЭКГ в отведениях V7 - V9, реципрокные изменения в отведениях V1 - V3). (post_im).

0 - нет

1 - форма комплекса QRS не изменена

2 - форма QRS комплекса QR

3 - форма QRS комплекса Qr

4 - форма QRS комплекса QS

47.Наличие ИМ правого желудочка IM_PG

0 - нет

1 - да

48. Ритм по ЭКГ при поступлении - синусовый (с чсс 60-90 в мин.) (ritm_ecg_p_01).

0 - нет

1 - да

49. Ритм по ЭКГ при поступлении - фибрилляция предсердий (ritm_ecg_p_02).

0 - нет

1 - да

50. Ритм по ЭКГ при поступлении - предсердный (ritm_ecg_p_04).

0 - нет

1 - да

51. Ритм по ЭКГ при поступлении - идиовентрикулярный (ritm_ecg_p_06).

0 - нет

1 - да

52. Ритм по ЭКГ при поступлении - синусовый с ЧСС более 90 в мин. (синусовая тахикардия) (ritm_ecg_p_07).

0 - нет

1 - да

53. Ритм по ЭКГ при поступлении - синусовый с ЧСС менее 60 в мин. (синусовая брадикардия) (ritm_ecg_p_08).

0 - нет

1 - да

54.Предсердная экстросистолия на ЭКГ при поступлении (n_r_ecg_p_01).

0 - нет

1 - да

55. Частая предсердная экстросистолия на ЭКГ при поступлении (n_r_ecg_p_02).

0 - нет

1 - да

56. Желудочковая экстросистолия на ЭКГ при поступлении (n_r_ecg_p_03).

0 - нет

1 - да

57. Частая желудочковая экстросистолия на ЭКГ при поступлении (n_r_ecg_p_04).

0 - нет

1 - да

58.Пароксизмы фибрилляции предсердий на ЭКГ при поступлении (n_r_ecg_p_05).

0 - нет

1 - да

59. Постоянная форма фибрилляции предсердий на ЭКГ при поступлении (n_r_ecg_p_06).

0 - нет

1 - да

60. Суправентрикулярная пароксизмальная тахикардия на ЭКГ при поступлении (n_r_ecg_p_08).
0 - нет
1 - да
61. Желудочковая пароксизмальная тахикардия на ЭКГ при поступлении (n_r_ecg_p_09).
0 - нет
1 - да
62. Фибрилляция желудочков на ЭКГ при поступлении (n_r_ecg_p_10).
0 - нет
1 - да
63. Синоатриальная блокада на ЭКГ при поступлении (n_p_ecg_p_01).
0 - нет
1 - да
64. А-В блокада I степени на ЭКГ при поступлении (n_p_ecg_p_03).
0 - нет
1 - да
65. А-В блокада II степени I типа на ЭКГ при поступлении (n_p_ecg_p_04).
0 - нет
1 - да
66. А-В блокада II степени II типа на ЭКГ при поступлении (n_p_ecg_p_05).
0 - нет
1 - да
67. А-В блокада III степени на ЭКГ при поступлении (n_p_ecg_p_06).
0 - нет
1 - да
68. Блокада передней ветви левой ножки пучка Гиса на ЭКГ при поступлении (n_p_ecg_p_07).
0 - нет
1 - да
69. Блокада задней ветви левой ножки пучка на ЭКГ при поступлении (n_p_ecg_p_08).
0 - нет
1 - да
70. Неполная блокада левой ножки пучка Гиса на ЭКГ при поступлении (n_p_ecg_p_09).
0 - нет
1 - да
71.Полная блокада левой ножки пучка Гиса на ЭКГ при поступлении (n_p_ecg_p_10).
0 - нет
1 - да
72. Неполная блокада правой ножки пучка Гиса на ЭКГ при поступлении (n_p_ecg_p_11).
0 - нет
1 - да
73. Полная блокада правой ножки пучка Гиса на ЭКГ при поступлении (n_p_ecg_p_12).
0 - нет
1 - да
74. Проведение фибринолитической терапии целиазой 750 тыс. ЕД (fibr_ter_01).
0 - нет
1 - да
75. Проведение фибринолитической терапии целиазой 1 млн. ЕД (fibr_ter_02).
0 - нет
1 - да
76. Проведение фибринолитической терапии

стрептодеказой 3 млн. ЕД (fibr_ter_03).

0 - нет

1 - да

77. Проведение фибринолитической терапии стрептазой (fibr_ter_05).

0 - нет

1 - да

78. Проведение фибринолитической терапии целиазой 500 тыс. ЕД (fibr_ter_06).

0 - нет

1 - да

79. Проведение фибринолитической терапии целиазой 250 тыс. ЕД (fibr_ter_07).

0 - нет

1 - да

80. Проведение фибринолитической терапии стрептодеказой 1,5 млн. ЕД (fibr_ter_08).

0 - нет

1 - да

81. Гипокалиемия (< 4 ммоль/л) (GIPO_K).

0 - нет

1 - да

82. Содержание K^+ в сыворотке крови (K_BLOOD).

83. Увеличение Na в сыворотке крови (более 150 ммоль/л) (GIPER_Na).

0 - нет

1 - да

84. Содержание Na в сыворотке крови (Na_BLOOD).

85. Содержание АлАТ в крови (ALT_BLOOD).

86. Содержание АсАТ в крови (AST_BLOOD).

87. Содержание лейкоцитов в крови ($\times 10^9$ /л) (L_BLOOD).

88. СОЭ (скорость оседания эритроцитов) (ROE).

89. Время, прошедшее от начала ангинозного приступа до поступления

в стационар: (TIME_B_S).

1 - менее 2 часов

2 - 2-4 часа

3 - 4-6 часов

4 - 6-8 часов

5 - 8-12 часов

6 - 12-24 часов

7 - более 1 суток

8 - более 2 суток

9 - более 3 суток

90. Повышение температуры в первые сутки: (TEMPER_1).

0 - нет

1 - 37-37,5°

2 - 37,5-38°

3 - 38-38,5°

4 - 38,5-39°

5 - 39-39,5°

6 - >39,5°

91. Повышение температуры во вторые сутки: (TEMPER_2).

0 - нет

1 - 37-37,5°

2 - 37,5-38°

3 - 38-38,5°

4 - 38,5-39°

5 - 39-39,5°

6 - >39,5°

92. Повышение температуры в третьи сутки: (TEMPER_3).

0 - нет

1 - 37-37,5°

2 - 37,5-38°

3 - 38-38,5°

4 - 38,5-39°

5 - 39-39,5°

6 - >39,5°

93. Рецидивирование ангинозных болей в 1-е сутки стац. лечения:
(R_AB_1).
0 - не рецидивировали
1 - однократно
2 - 2 раза
3 - 3 раза и.т.д.

94. Рецидивирование ангинозных болей во 2-е сутки стац. лечения:
(R_AB_2).
0 - не рецидивировали
1 - однократно
2 - более 1-го раза

95. Рецидивирование ангинозных болей в 3-и сутки стац. лечения: (R_AB_3).
0 - не рецидивировали
1 - однократно
2 - более 1-го раза

96. Применение наркотических анальгетиков кардиобригадой: (NA_KB).
0 - не вводились
1 - вводились однократно

97. Применение ненаркотических анальгетиков кардиобригадой: (NOT_NA_KB).
0 - нет
1 - да

98. Введение лидокаина кардиобригадой: (LID_KB).
0 - нет
1 - да

99. Применение жидких нитратов в ОРиИТ: (NITR_S).
0 - нет
1 - да

100. Применение наркотических анальгетиков в ОРиИТ в 1 сутки: (NA_R_1).
0 - не вводились
1 - вводились однократно
2 - вводились двукратно
3 - вводились 3-х кратно
4 - вводились 4-х кратно

101. Применение наркотических анальгетиков в ОРиИТ во 2 сутки: (NA_R_2).
0 - не вводились
1 - вводились однократно
2 - вводились более 1 раза

102. Применение наркотических анальгетиков в ОРиИТ в 3 сутки: (NA_R_3).
0 - не вводились
1 - вводились однократно
2 - вводились более 1 раза

103. Применение ненаркотических анальгетиков в ОРиИТ в 1 сутки: (NOT_NA_1).
0 - не вводились
1 - вводились однократно
2 - вводились двукратно
3 - вводились 3-х кратно
4 - вводились 4-х кратно
и.т.д.

104. Применение ненаркотических анальгетиков в ОРиИТ во 2 сутки: (NOT_NA_2).
0 - не вводились
1 - вводились однократно
2 - вводились более 1 раза

105. Применение ненаркотических анальгетиков в ОРиИТ в 3 сутки: (NOT_NA_3).
0 - не вводились
1 - вводились однократно
2 - вводились более 1 раза

106. Введение лидокаина в ОРиИТ: (LID_S).
0 - нет
1 - да

107. Прием блокаторов в ОРиИТ: (B_BLOK_S).
0 - нет

1 - да

108. Прием антагонистов Са в ОРиИТ: (ANT_Ca_S).

0 - нет

1 - да

109. Введение антикоагулянтов (гепарин): (GEPAR_S).

0 - нет

1 - да

110.Прием аспирина: (ASP_S).

0 - нет

1 - да

111. Прием тиклида: (TIKL_S).

0 - нет

1 - да

112. Прием трентала: (TRENT_S).

0 - нет

1 - да

113.Фибрилляция/трепетание предсердий (FIBR_PREDS).

0 - нет

1 - да

114.Суправентрикулярная тахикардия (PREDS_TAH).

0 - нет

1 - да

115.Желудочковая тахикардия (JELUD_TAH).

0 - нет

1 - да

116.Фибрилляция желудочков (FIBR_JELUD).

0 - нет

1 - да

117.Полная а-в блокада (A_V_BLOK).

0 - нет

1 - да

118.Отек легких (OTEK_LANC).

0 - нет

1 - да

119.Разрыв сердца (RAZRIV).

0 - нет

1 - да

120.Синдром Дресслера (DRESSLER).

0 - нет

1 - да

121.Хроническая сердечная недостаточность (ZSN).

0 - нет

1 - да

122.Рецидив инфаркта миокарда (REC_IM).

0 - нет

1 - да

123.Постинфарктная стенокардия (P_IM_STEN).

0 - нет

1 - да

124.Летальный исход (LET_IS).

0 - нет

1 - кардиогенный шок

2 - отек легких

3 - разрыв миокарда

4 - прогрессирование застойной сердечной недостаточности

5 - тромбоэмболия

6 - асистолия

7 - фибрилляция желудочков

ПРИЛОЖЕНИЕ Б

(Справочное)

Признаки с нулевой и максимальной важностью для задачи прогнозирования осложнений инфаркта миокарда

Таблица Б.1 – Признаки с нулевой важностью для задачи ФЖ

Группа медицинских параметров	**Номер признака в базе данных**	**Название признака**
1	2	3
Состояние сердечнососудистой системы	12	1. Предсердная экстрасистология в анамнезе
	16	2. Желудочковая пароксизмальная тахикардия в анамнезе
	17	3. Фибрилляция желудочков в анамнезе
	18	4. А-В блокада I степени в анамнезе
	19	5. А-В блокада III степени в анамнезе
	21	6. Неполная блокада левой ножки пучка Гиса в анамнезе
	22	7. Полная блокада левой ножки пучка Гиса в анамнезе
	24	8. Полная блокада правой ножки пучка Гиса в анамнезе
Система органов дыхания	31	1. Хроническая пневмония в анамнезе
Течение заболевания в первые дни ИМ	96	1. Применение наркотических анальгетиков кардиобригадой
Осложнения, возникшие в момент транспортировки больного в клинику	38	1. Кардиогенный шок в момент поступления в отделение реанимации и интенсивной терапии
	40	2. Пароксизм суправентрикулярной тахикардии в момент поступления в отделение реанимации и интенсивной терапии (или на догоспитальном этапе)

1	2	3
Вид лекарственного препарата примененного (не примененного) у пациента при проведении фибринолитической терапии	74 77 79 80	1. Проведение фибринолитической терапии целиазой 750 тыс. ЕД 2. Проведение фибринолитической терапии стрептазой 3. Проведение фибринолитической терапии целиазой 250 тыс. ЕД 4. Проведение фибринолитической терапии стрептодеказой 1,5 млн. ЕД

Таблица Б.2 – Признаки с нулевой важностью для задачи ФП

Группа медицинских параметров	**Номер признака в базе данных**	**Название признака**
Состояние сердечнососудистой системы	17 18 19 21 22 23 24	1. Фибрилляция желудочков в анамнезе 2. A-B блокада I степени в анамнезе 3. A-B блокада III степени в анамнезе 4. Неполная блокада левой ножки пучка Гиса в анамнезе 5. Полная блокада левой ножки пучка Гиса в анамнезе 6. Неполная блокада правой ножки пучка Гиса в анамнезе 7. Полная блокада правой ножки пучка Гиса в анамнезе

Таблица Б.3 – Признаки с нулевой важностью для задачи ОЛ

Группа медицинских параметров	**Номер признака в базе данных**	**Название признака**
Состояние сердечнососудистой системы	12 16 17	1. Предсердная экстрасистология в анамнезе 2. Желудочковая пароксизмальная тахикардия в анамнезе 3. Фибрилляция желудочков в анамнезе
Система органов дыхания	31	1. Хроническая пневмония в анамнезе

Таблица Б.4 – Признаки с нулевой важностью для задачи РС

Группа медицинских параметров	**Номер признака в базе данных**	**Название признака**
1	2	3
Состояние сердечнососудистой системы	12 14 16 19 20 21 22 23 24	1. Предсердная экстрасистология в анамнезе 2. Пароксизмы фибрилляции предсердий в анамнезе 3. Желудочковая пароксизмальная тахикардия в анамнезе 4. А-В блокада III степени в анамнезе 5. Блокада передней ветви левой ножки пучка Гиса в анамнезе 6. Неполная блокада левой ножки пучка Гиса в анамнезе 7. Полная блокада левой ножки пучка Гиса в анамнезе 8. Неполная блокада правой ножки пучка Гиса в анамнезе 9. Полная блокада правой ножки пучка Гиса в анамнезе
Осложнения, возникшие в момент транспортировки больного в клинику	38 40	1. Кардиогенный шок в момент поступления в отделение реанимации и интенсивной терапии 2. Пароксизм суправентрикулярной тахикардии в момент поступления вотделение реанимации и интенсивной терапии (ОРИТ),
Осложнения, возникшие в момент транспортировки больного в клинику	41 42	3. Пароксизм желудочковой тахикардии в момент поступления в ОРИТ, (или на догоспитальном этапе) 4. Фибрилляция желудочков в момент поступления в ОРИТ, (или на догоспитальном этапе)

1	2	3
Основной водитель ритма, наличие (отсутствие) аритмий и нарушений проводимости на ЭКГ в момент поступления больного в реанимационное отделение	54 55 57 60 61 62 70	1. Предсердная экстросистолия на ЭКГ при поступлении 2. Частая предсердная экстросистолия на ЭКГ при поступлении 3. Частая желудочковая экстросистолия на ЭКГ при поступлении 4. Суправентрикулярная пароксизмальная тахикардия на ЭКГ при поступлении 5. Желудочковая пароксизмальная тахикардия на ЭКГ при поступлении 6. Фибрилляция желудочков на ЭКГ при поступлении 7. Блокада задней ветви левой ножки пучка на ЭКГ при поступлении
Вид лекарственного препарата примененного (не примененного) у пациента при проведении фибринолитической терапии	74 77 79 80	1. Проведение фибринолитической терапии целиазой 750 тыс. ЕД 2. Проведение фибринолитической терапии стрептазой 3. Проведение фибринолитической терапии целиазой 250 тыс. ЕД 4. Проведение фибринолитической терапии стрептодеказой 1,5 млн. ЕД
Система органов дыхания	31 32	1. Хроническая пневмония в анамнезе 2. Туберкулез легких в анамнезе
Течение заболевания в первые дни ИМ	97 98	1. Применение ненаркотических анальгетиков кардиобригадой 2. Введение лидокаина кардиобригадой

Таблица Б.5 – Признаки с нулевой важностью для задачи ЛИ

Группа медицинских параметров	**Номер признака в базе данных**	**Название признака**
1	2	3
Состояние сердечнососудистой системы	12 16 17	1. Предсердная экстрасистология в анамнезе 2. Желудочковая пароксизмальная тахикардия в анамнезе 3. Фибрилляция желудочков в анамнезе

1	2	3
Осложнения, возникшие в момент транспортировки больного в клинику	41	1. Пароксизм желудочковой тахикардии в момент поступления в отделение реанимации и интенсивной терапии, (или на догоспитальном этапе)
Основной водитель ритма, наличие (отсутствие) аритмий и нарушений проводимости на ЭКГ в момент поступления в ОР	65	1. А-В блокада II степени I типа на ЭКГ при поступлении

Таблица Б.6 – Признаки с максимальной важностью для задачи ФЖ

Важность, %	**Номер признака в базе данных**	**Название признака**
18,82	35	Систолическое АД по данным ОРиИТ
17,74	36	Диастолическое АД по данным ОРиИТ
17,06	9	Длительность течения арт. гипертензии
16,85	89	Время, прошедшее от начала ангинозного приступа до поступления в стационар
16,45	43	Наличие инфаркта передней стенки левого желудочка (изменения на ЭКГ в отведениях V2 - V4)
12,70	4	Стенокардия напряжения в анамнезе
12,34	45	Наличие инфаркта нижней стенки левого желудочка (изменения на ЭКГ в отведениях III, AVF, II)
11,82	100	Применение наркотических анальгетиков в ОРиИТ в 1 сутки

Таблица Б.7 – Признаки с максимальной важностью для задачи ФП

Важность, %	**Номер признака в базе данных**	**Название признака**
1	2	3
26,64	89	Время, прошедшее от начала ангинозного приступа до поступления в стационар
26,52	1	Возраст
24,91	4	Стенокардия напряжения в анамнезе
24,53	9	Длительность течения арт. гипертензии

1	2	3
20,90	87	Содержание лейкоцитов в крови ($\times 10^9$ /л)
18,49	36	Диастолическое АД по данным ОРиИТ
17,15	43	Наличие инфаркта передней стенки левого желудочка (изменения на ЭКГ в отведениях V2 - V4)
15,841	35	Систолическое АД по данным ОРиИТ
15,69	82	Содержание K^+ в сыворотке крови
15,67	5	Функциональный класс стенокардии в последний год
15,26	45	Наличие инфаркта нижней стенки левого желудочка (изменения на ЭКГ в отведениях III, AVF, II)

Таблица Б.8 – Признаки с максимальной важностью для задачи ОЛ

Важность, %	**Номер признака в базе данных**	**Название признака**
32,96	9	Длительность течения арт. гипертензии
25,37	89	Время, прошедшее от начала ангинозного приступа до поступления в стационар
24,55	4	Стенокардия напряжения в анамнезе
23,82	1	Возраст
15,83	87	Содержание лейкоцитов в крови ($\times 10^9$ /л)
15,61	43	Наличие инфаркта передней стенки левого желудочка (изменения на ЭКГ в отведениях V2 - V4)
15,41	100	Применение наркотических анальгетиков в ОРиИТ в 1 сутки
14,59	35	Систолическое АД по данным ОРиИТ
14,11	99	Применение жидких нитратов в ОРиИТ

Таблица Б.9 – Признаки с максимальной важностью для задачи РС

Важность, %	**Номер признака в базе данных**	**Название признака**
1	2	3
29,03	9	Длительность течения арт. гипертензии
21,91	36	Диастолическое АД по данным ОРиИТ
20,79	1	Возраст
16,90	89	Время, прошедшее от начала ангинозного приступа до поступления в стационар

1	2	3
15,93	44	Наличие инфаркта боковой стенки левого желудочка (изменения на ЭКГ в отведениях V5 - V6, I, AVL
14,41	43	Наличие инфаркта передней стенки левого желудочка (изменения на ЭКГ в отведениях V2 - V4)
14,04	35	Систолическое АД по данным ОРиИТ
12,89	90	Повышение температуры в первые сутки

Таблица Б.10 – Признаки с максимальной важностью для задачи ЛИ

Важность, %	**Номер признака в базе данных**	**Название признака**
27,75	4	Стенокардия напряжения в анамнезе
21,57	10	Наличие хронической сердечной недостаточности (СН) в анамнезе
21,43	89	Время, прошедшее от начала ангинозного приступа до поступления в стационар
20,90	87	Содержание лейкоцитов в крови ($\times 10^9$ /л)
17,85	1	Возраст
17,19	95	Рецидивирование ангинозных болей в 3-и сутки стац. лечения
17,06	9	Длительность течения арт. гипертензии
16,30	44	Наличие инфаркта боковой стенки левого желудочка (изменения на ЭКГ в отведениях V5 - V6, I, AVL
15,77	43	Наличие инфаркта передней стенки левого желудочка (изменения на ЭКГ в отведениях V2 - V4)
15,00	103	Применение ненаркотических анальгетиков в ОРиИТ в 1 сутки
14,79	45	Наличие инфаркта нижней стенки левого желудочка (изменения на ЭКГ в отведениях III, AVF, II)
14,78	46	Наличие инфаркта задней стенки левого желудочка (изменения на ЭКГ в отведениях V7 - V9, реципрокные изменения в отведениях V1 - V3).
13,79	99	Применение жидких нитратов в ОРиИТ